Matchability-Aware Conformal Prediction for Open-Ended Language Model Generation

Ilia Azizi
University of Lausanne, Switzerland
BegooAI, Switzerland
ilia@begoo.ai

Abstract

Large language models (LLMs) often do not produce calibrated uncertainty estimates for their answers. We propose CAELUM, a framework that builds semantic prediction sets while separating aleatoric from epistemic uncertainty. CAELUM samples answers using an ensemble of clarification prompts, clusters them by semantic equivalence, and estimates the aleatoric and epistemic components via kernel-based von Neumann entropy before calibrating nonconformity scores with split conformal prediction. We use a weight $w_\lambda(x) = 1 + U_{\text{ale}}(x) + \lambda\, U_{\text{epi}}(x)$ to control for the contribution of the epistemic uncertainty of each instance and fit a coefficient $\lambda^\star$ to minimize prediction set size. When the objective is nonflat and selection is stable across splits, $\lambda^\star$ could be a diagnostic, where large values may indicate epistemic dominance (retrieve, abstain) and $\lambda^\star \to 0$ aleatoric dominance (clarify). For individual queries, a log ratio $R(x)$ compares the weighted epistemic component with the aleatoric component. Theoretically, we demonstrate that the coverage is capped by *matchability*, the fraction of items whose reference answer appears in any sampled cluster. Empirically, this matchability can be predicted from the original question features using a lightweight classifier. We use those predictions to define a stratum and derive a conformal coverage bound within it. Across 30 LLM and dataset combinations (4–32B), mean coverage is 0.74 in the learned stratum versus 0.65 overall, averaged over the 78 evaluations using the learned classifier. Project page: https://unco3892.github.io/caelum/

1 Introduction

Large language models (LLMs) [1, 2] are increasingly deployed in settings where model hallucinations can have practical consequences [3]. Assessing reliability also requires understanding underspecified requests, interactions, and how uncertainty is communicated to users [4]. Conformal methods [5, 6] offer finite-sample marginal coverage guarantees for regression and classification tasks under exchangeability. However, considerable obstacles exist to applying conformal methods to open-ended text generation tasks, including a large response space, discrepancies between semantic and exact-match response accuracy, and limitations of a single conformal parameter in separating input ambiguity from knowledge gaps.

We introduce Calibrated Aleatoric-Epistemic Language Uncertainty Modeling (CAELUM), which constructs semantic prediction sets from clarification ensembles. The idea is to distinguish the two sources of uncertainty [7, 8], namely aleatoric and epistemic uncertainty. *Aleatoric uncertainty* arises from input ambiguity and is irreducible given the query. For example, "What is the capital of South Africa?" has three valid answers depending on which branch of government the user means. *Epistemic uncertainty* reflects the model's lack of knowledge, for example where training data are sparse. Distinguishing these sources can inform different downstream responses.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

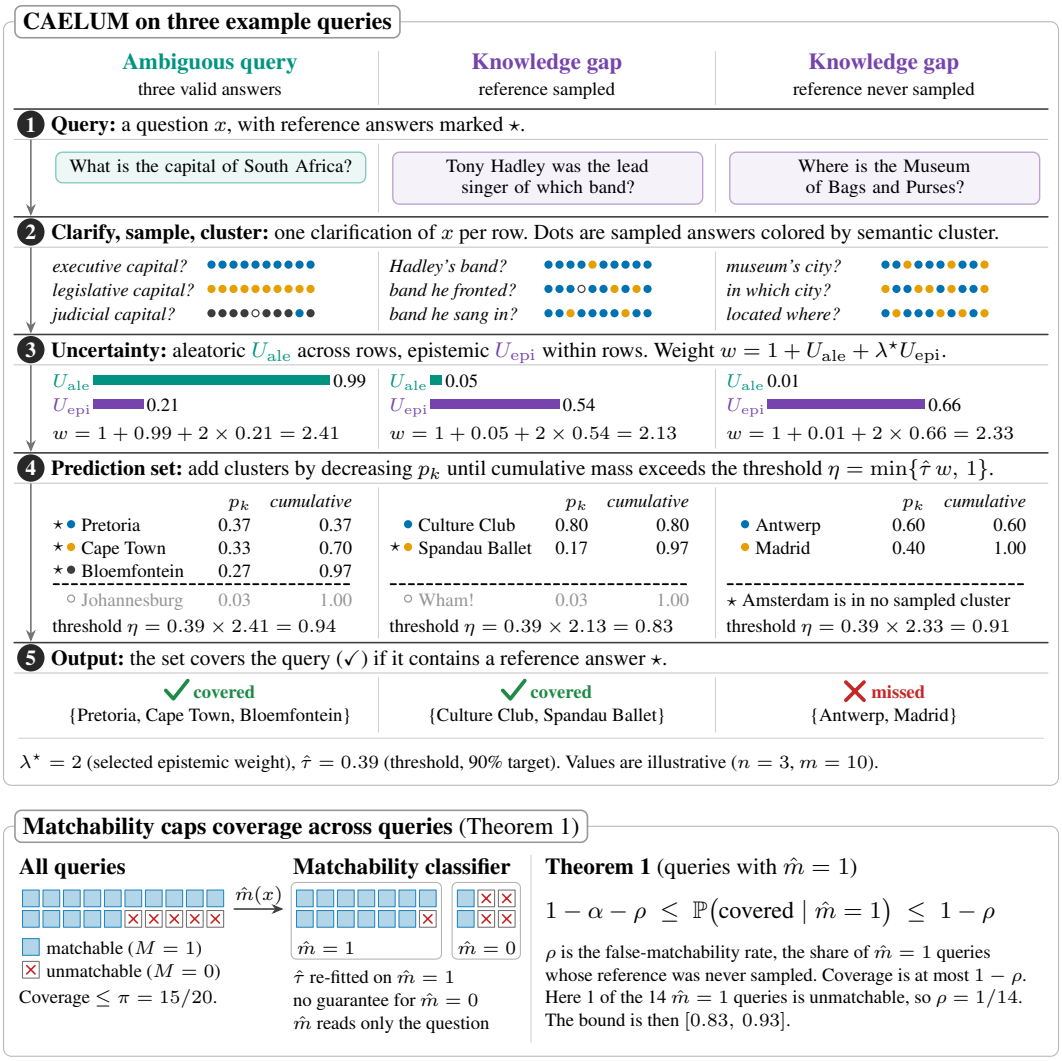


Figure 1: **Method overview.** Each top-panel column shows the same five steps. Disagreement between clarifications raises U_{ale}, and disagreement within a clarification raises U_{epi}. The weight w sets the effective threshold $\eta = \min\{\hat{\tau} w, 1\}$ for selecting clusters in Equation (7). The third column's reference answer is absent from sampled support, so expanding the set cannot include it. The lower panel shows the matchability bound. Main experiments use $n = 5$ and kernel entropy.

Existing methods address different aspects of uncertainty estimation and calibration in open-ended language generation. Semantic entropy [3, 9] clusters semantically equivalent LLM generations and computes entropy over the resulting meaning distribution. Kernel language entropy [10] extends this approach using semantic similarity kernels and von Neumann entropy. These methods use uncertainty estimates to detect hallucinations, without decomposing uncertainty into aleatoric and epistemic components or constructing prediction sets with conformal coverage guarantees. The spectral framework of Walha et al. [11] decomposes uncertainty into aleatoric and epistemic components but evaluates uncertainty *ranking* without constructing prediction sets. ConU [12] calibrates prediction sets for free-form generation using a self-consistency measure without explicitly separating the two uncertainty components. CAELUM combines the spectral decomposition with an adaptation of the two-parameter calibration approach of Azizi et al. [13], makes the matchability ceiling explicit, and derives a Mondrian coverage bound for a learned matchability stratum (see Appendix A for further discussion). CAELUM and its aleatoric-only baseline have similar average set sizes (Table 6).

What CAELUM does. For a given query x, we generate n clarifications and m LLM responses for each clarification. We perform semantic clustering over the nm responses and calculate three uncertainty scores using a kernel applied to response embeddings: the pooled uncertainty $U_{\text{total}}(x)$, the mean per-clarification uncertainty $U_{\text{epi}}(x)$, and the empirical residual $U_{\text{ale}}(x) = \max(U_{\text{total}}(x) - U_{\text{epi}}(x), 0)$. Figure 1 shows the pipeline using cluster Shannon entropy. We construct a conformal prediction set $\mathcal{C}_\lambda(x)$ using an adaptive prediction set (APS) score divided by the weight

$$w_\lambda(x) \;=\; 1 \;+\; U_{\text{ale}}(x) \;+\; \lambda \cdot U_{\text{epi}}(x), \qquad \lambda \geq 0, \tag{1}$$

tuning $\lambda^\star$ on the selection split $\mathcal{V}_1$ to minimize average prediction-set size (APSS) over a finite grid and calibrating the conformal threshold $\hat{\tau}$ on a disjoint calibration split $\mathcal{V}_2$. The per-instance score $R(x) = \log(\lambda^\star U_{\text{epi}}(x) + \epsilon) - \log(U_{\text{ale}}(x) + \epsilon)$, with $\epsilon = 10^{-6}$, summarizes the relative contributions of weighted epistemic and aleatoric uncertainty for individual queries.

The contributions are as follows:

1. We identify the *matchability ceiling*, an upper bound on the coverage of prediction sets restricted to sampled clusters. A coverage bound for a learned matchability stratum makes the false matchability penalty explicit (Theorem 1).
2. We construct semantic prediction sets using clarification ensembles, kernel uncertainty decomposition, and uncertainty-weighted APS scores. Separate splits are used for both selecting the epistemic uncertainty weight, $\lambda^\star$, to minimize average validation set size over a finite grid and setting the conformal threshold.
3. In our main experiments, we evaluate support prediction, set efficiency, and uncertainty ranking for three generators and six datasets. A finite-grid diagnostic tests whether the set-size objective can distinguish between different weights. Experiments with larger generators test whether the support limitation also holds for larger generators.

2 Method

2.1 Problem setup

Given a query x in the query space $\mathcal{X}$, a black-box LLM $\mathcal{M}$ from which we can sample responses, and a calibration dataset of queries and reference answers $\mathcal{D}_{\text{cal}} = \{(x_i, y_i^*)\}_{i=1}^N$, we seek to construct a **semantic prediction set** $\mathcal{C}(x)$ over semantic clusters of answers. The ideal target for conformal inference is marginal coverage

$$\mathbb{P}\big(y_{N+1}^* \in \mathcal{C}(x_{N+1})\big) \geq 1 - \alpha, \tag{2}$$

where $y^* \in \mathcal{C}(x)$ means that the reference cluster selected by a fixed semantic matching rule is included in $\mathcal{C}(x)$. This rule is applied to the full collection of sampled clusters before selecting the prediction set. The parameter $\alpha \in (0, 1)$ specifies the desired error rate. The probability averages over the calibration data, the test example, and the sampling randomness. In our setting, $\mathcal{C}(x)$ is restricted to clusters of sampled answers from the model. Let $M(x, y^*) = 1$ denote *matchability*, meaning that the same rule selects a sampled cluster for the reference y^*. If $M = 0$, no threshold over the sampled clusters can include the reference y^*. We distinguish support-conditional coverage,

$$\mathbb{P}\big(y_{N+1}^* \in \mathcal{C}(x_{N+1}) \mid M(x_{N+1}, y_{N+1}^*) = 1\big) \geq 1 - \alpha, \tag{3}$$

which holds for the support-restricted calibration procedure under the assumptions of Proposition 1, including $\mathbb{P}(M = 1) > 0$, from population coverage, which is bounded above by $\pi := \mathbb{P}(M = 1)$.

2.2 Sampling and semantic clustering

We use the two-stage sampling approach described by ICE [14]. We sample n clarifications $W_1, \ldots, W_n$ from the LLM for x. For factual questions, clarifications are alternative ways of stating x. For ambiguous questions, clarifications are intended to disambiguate x. We sample m responses $y_{ij} \sim \mathcal{M}(\cdot \mid W_i)$ for each clarification. We use a greedy embedding algorithm to group the nm sampled responses into approximate semantic clusters $\{G_1(x), \ldots, G_K(x)\}$. The first unassigned response in generation order becomes a cluster seed. All remaining unassigned responses whose cosine similarity to that seed is at least δ join its cluster. We repeat this process until every response has been assigned. The clusters are ordered by decreasing proportion $p_k(x) = |G_k(x)|/(nm)$, and the most frequent response in each cluster is selected as its representative. Other methods, including NLI-based bidirectional entailment [9], can be used in place of our clustering approach.

2.3 Uncertainty decomposition

We adapt the ICE decomposition [14] to semantic clusters and report results using the kernel-based von Neumann entropy (VNE) formulation of Walha et al. [11] (Appendix F). We present the cluster-Shannon formulation below for clarity. In this formulation, *total uncertainty* is the entropy of the pooled cluster proportions $p_k^{\text{pool}}(x) = p_k(x)$, $U_{\text{total}}(x) = -\sum_k p_k^{\text{pool}}(x) \log p_k^{\text{pool}}(x)$. *Epistemic uncertainty* is the mean entropy within clarifications, using the same pooled cluster assignments. We restrict these assignments to the m answers of clarification W_i to obtain $p_k^{(i)}(x)$, and then compute

$$U_{\text{epi}}(x) = \tfrac{1}{n} \sum_{i=1}^{n} H_i(x), \qquad H_i(x) = -\sum_k p_k^{(i)}(x) \log p_k^{(i)}(x). \tag{4}$$

This quantity measures response variability within a clarification. A generated clarification may not fully resolve the ambiguity in the original question. In this Shannon formulation, $U_{\text{epi}}(x) = 0$ when all responses within each clarification belong to a single cluster, even if the cluster differs across clarifications. *Aleatoric uncertainty* is the residual $U_{\text{ale}}(x) = U_{\text{total}}(x) - U_{\text{epi}}(x)$. With the same number of responses per clarification, the aleatoric residual is nonnegative in the shared-cluster Shannon formulation due to the concavity of entropy.

In the VNE formulation, the same encoder and RBF kernel are used for the pooled responses and the responses within each clarification. The entropies are not divided by the logarithm of the sample count. With equal numbers of responses per clarification, entropy concavity also guarantees a nonnegative residual in the shared kernel feature space. We use $U_{\text{ale}}(x) = \max(U_{\text{total}}(x) - U_{\text{epi}}(x), 0)$ to guard against numerical error (Appendix F). The Shannon formulation follows the identity $H(Y \mid X) = H(Y \mid X, C) + \mathcal{I}(Y; C \mid X)$, with clarifications C as the conditioning variable. Both formulations require only sampling access to the LLM.

2.4 Conformal semantic prediction sets

Nonconformity score. For a query x with clusters ordered by mass $p_1(x) \geq \cdots \geq p_K(x)$, let k^* be the index of the cluster selected by the fixed reference-matching rule for y^*, when a match exists. We use a deterministic APS-style score [15], given by the cumulative mass preceding that cluster,

$$s(x, y^*) = \sum_{j=1}^{k^*-1} p_j(x). \tag{5}$$

The APS score is 0 for questions whose selected reference cluster is first in the ordering and can approach 1 when that cluster occurs late in the ordering. If none of the clusters matches the reference, we set $M = 0$, assign $s(x, y^*) = 1$, and mark the question as uncoverable from any subset of sampled clusters. We use both deterministic and randomized APS scores. The randomized variant is $s_{\text{rand}}(x, y^*) = \sum_{j<k^*} p_j(x) + U\, p_{k^*}(x)$, where an independent $U \sim \text{Unif}(0, 1)$ is drawn for each item and held fixed across candidate values of λ. Unmatched items have score 1 under both schemes. The prediction sets in Equation (7) use no additional random draw. For matchable items, $s_{\text{rand}} \geq s$ means that $s_{\text{rand}}(x, y^*)/w_\lambda(x) \leq \hat{\tau}$ is sufficient for reference inclusion under the cumulative-threshold rule. Deterministic APS scoring is used for the main table and selective-answering results, while seeded randomized APS scoring is used for the CAELUM multi-seed results and the stratified analysis.

Uncertainty-weighted threshold. The per-instance weight

$$w_\lambda(x) = 1 + U_{\text{ale}}(x) + \lambda \cdot U_{\text{epi}}(x), \qquad \lambda \geq 0, \tag{6}$$

scales the conformal threshold. Since cluster proportions sum to one, we cap the effective threshold at one,

$$\eta_\lambda(x; \tau) = \min\{\tau \cdot w_\lambda(x),\, 1\}.$$

We include cluster $G_j(x)$ if the cumulative mass of all preceding clusters is at most this threshold,

$$\mathcal{C}_\lambda(x) = \big\{G_j(x) : \sum_{\ell<j} p_\ell(x) \leq \eta_\lambda(x; \hat{\tau})\big\}. \tag{7}$$

For a fixed threshold τ, increasing λ can only expand a query's prediction set. Smaller λ reduces the contribution of U_{epi} relative to U_{ale}. This monotonicity need not hold when the threshold is recalibrated for each λ.

Split calibration. We use three disjoint splits to select λ on $\mathcal{V}_1$, calibrate the threshold on $\mathcal{V}_2$, and evaluate on test. The encoder, δ, and matching rule may also be tuned on $\mathcal{V}_1$, then fixed for calibration and testing (Appendix S.3). We apply *score normalization*,

$$\tilde{s}_\lambda(x, y^*) = \frac{s(x, y^*)}{w_\lambda(x)}, \tag{8}$$

so the conformal quantile $\hat{\tau}_\lambda$ depends on λ. For randomized calibration, we replace s by s_{rand} in this normalization and use the resulting scores in both calibration steps. Without normalization, the scores and calibrated threshold would be independent of λ. Increasing λ would then only expand the sets, making $\lambda = 0$ an APSS minimizer, possibly tied with other grid values (Remark 3).

The quantile Q_q is the upward empirical quantile of all calibration scores, including zeros. For a calibration subset of size L, the theoretical construction uses the score at rank $r = \lceil (1-\alpha)(L+1) \rceil$. If there are fewer than r calibration items, including an empty calibration set, we set the threshold to $+\infty$ and include all sampled clusters. The quantile convention and this fallback are given in Appendix S.1.

Step 1: λ selection on $\mathcal{V}_1$. For each λ in the grid Λ, compute $\tilde{s}_{\lambda,i} = s_i / w_\lambda(x_i)$ on $\mathcal{V}_1$, set $\hat{\tau}_\lambda = Q_{\lceil (1-\alpha)(|\mathcal{V}_1|+1) \rceil / |\mathcal{V}_1|}(\{\tilde{s}_{\lambda,i}\}_{i \in \mathcal{V}_1})$, and choose

$$\lambda^\star = \arg\min_{\lambda \in \Lambda} \frac{1}{|\mathcal{V}_1|} \sum_{i \in \mathcal{V}_1} |\mathcal{C}_\lambda(x_i)|. \tag{9}$$

For this selection step, $\mathcal{C}_\lambda$ uses the provisional threshold $\hat{\tau}_\lambda$. Ties are resolved by choosing the smallest minimizing weight.

Step 2: Conformal threshold on $\mathcal{V}_2$. With $\lambda^\star$ fixed, compute

$$\hat{\tau} = Q_{\lceil (1-\alpha)(|\mathcal{V}_2|+1) \rceil / |\mathcal{V}_2|}\big(\{\tilde{s}_{\lambda^\star,i}\}_{i \in \mathcal{V}_2}\big). \tag{10}$$

The final prediction set for a new query x is $\mathcal{C}_{\lambda^\star}(x)$, as defined in Equation (7), with $\hat{\tau}$ and $\lambda^\star$. For deterministic scoring, $\tilde{s} \le \hat{\tau}$ is equivalent to $s \le \hat{\tau}\, w_\lambda(x)$ since $w_\lambda(x) > 0$. If $\hat{\tau}\, w_\lambda(x) \ge 1$, the prediction set includes all sampled clusters. For smaller values, we include clusters whose preceding cumulative mass is at most $\hat{\tau}\, w_\lambda(x)$. In Table 1, we calibrate the threshold using all of $\mathcal{V}_2$. A reference missing from the sampled clusters cannot be covered, so population coverage cannot exceed the matchability rate. The result below instead calibrates the threshold using only matched examples, $\mathcal{V}_2^M = \{i \in \mathcal{V}_2 : M_i = 1\}$.

When can the weight be identified? Write $b_{ij} = \sum_{\ell < j} p_\ell(x_i)$. On a finite grid Λ, the prediction-set size for x_i can change only if a cluster boundary satisfies

$$\min_{\lambda \in \Lambda} \eta_\lambda(x_i; \hat{\tau}_\lambda) < b_{ij} \le \max_{\lambda \in \Lambda} \eta_\lambda(x_i; \hat{\tau}_\lambda).$$

Hence,

$$f_{\mathrm{move}}(\mathcal{V}_1) = |\mathcal{V}_1|^{-1} \sum_{i \in \mathcal{V}_1} \mathbf{1}\Big\{ \min_{\lambda \in \Lambda} |\mathcal{C}_\lambda(x_i)| < \max_{\lambda \in \Lambda} |\mathcal{C}_\lambda(x_i)| \Big\}$$

measures the fraction of validation items whose set sizes differ across grid points. If every validation item's set size stays the same across grid weights, APSS is constant. Changes in individual set sizes can also cancel in the average, so several weights may minimize APSS. We report the objective range and the fraction of grid weights attaining the minimum in Table 14. The separate held-out movement audit and the validation exceptions are in Appendix S.4.

Proposition 1 (Split coverage on generated support)**.** *Assume $\mathbb{P}(M = 1) > 0$. Conditional on $\mathcal{V}_1$, assume exchangeability of the calibration examples and test example within the event $M = 1$, including their sampled generations and any score randomization. Select $\lambda^\star$ using only $\mathcal{V}_1$ and compute the conformal threshold from $\mathcal{V}_2^M = \{i \in \mathcal{V}_2 : M_i = 1\}$ at rank $r_M = \lceil (1-\alpha)(|\mathcal{V}_2^M|+1) \rceil$, using $+\infty$ if $r_M > |\mathcal{V}_2^M|$. The normalized scoring rule is therefore fixed when computing this threshold. By the standard split conformal argument [5, 6],*

$$\mathbb{P}\big(y^*_{N+1} \in \mathcal{C}_{\lambda^\star}(x_{N+1}) \mid M(x_{N+1}, y^*_{N+1}) = 1\big) \ge 1 - \alpha.$$

The calibration procedure adapts the two-split construction of CLEAR introduced by Azizi et al. [13, Definition B.1 and Lemma B.2], using semantic clusters as the output space. As the sampled support can omit the reference answer, Proposition 1 guarantees coverage conditional on matchability rather than unconditional coverage at the nominal level. See Appendix B for details.

2.5 Stratified coverage via learned matchability

Proposition 1 is conditional on generated support. On hard cells, matchability caps the population coverage at $\pi = \mathbb{P}(M = 1)$ regardless of α, so a cell with $\pi = 0.4$ makes a 0.9 nominal target unattainable. We make this dependence explicit via Mondrian conformal prediction [16, 17], which restricts calibration to a stratum where the ceiling can be estimated and audited. Train a binary classifier $\hat{m} : \mathcal{X} \to \{0, 1\}$ on a held-out split $\mathcal{V}_0$ (disjoint from $\mathcal{V}_1, \mathcal{V}_2$ and the test point) using only pre-generation features of x. Define the predicted-matchable subsample $\mathcal{S} = \{i \in \mathcal{V}_2 : \hat{m}(x_i) = 1\}$ and let $r = \lceil (1-\alpha)(|\mathcal{S}| + 1) \rceil$. For nonempty $\mathcal{S}$ with $r \le |\mathcal{S}|$, the stratified threshold is

$$\hat{\tau}_{\mathrm{strat}}(\lambda) = Q_{\lceil (1-\alpha)(|\mathcal{S}|+1) \rceil / |\mathcal{S}|}\big(\{\tilde{s}_{\lambda,i}\}_{i \in \mathcal{S}}\big). \tag{11}$$

If $\mathcal{S}$ is empty or $r > |\mathcal{S}|$, use threshold $+\infty$ and return all sampled clusters. The resulting set $\mathcal{C}^{\mathrm{strat}}_{\lambda_0}$, for any λ_0 fixed or selected using $\mathcal{V}_1$, then satisfies the following two-sided guarantee.

Theorem 1 (Matchability-stratified coverage with false matchability penalty)**.** *Let $\hat{m} : \mathcal{X} \to \{0, 1\}$ be any binary classifier trained on $\mathcal{V}_0$ (disjoint from $\mathcal{V}_2$ and the test point), let λ_0 be fixed or selected using $\mathcal{V}_1$ (also disjoint from $\mathcal{V}_2$ and the test point), and let $\mathcal{C}^{\mathrm{strat}}_{\lambda_0}(x)$ use the stratified threshold $\hat{\tau}_{\mathrm{strat}}(\lambda_0)$ from Equation* (11)*, with the $+\infty$ convention above. Condition on $\mathcal{V}_0$, $\mathcal{V}_1$, and any randomness used to fit $\hat{m}$ or select λ_0. Assume that the calibration and test examples, including their generated samples and any score randomization, are exchangeable under this conditioning. All probabilities below, including $\rho_{\hat{m}}$, are understood under the same conditioning. Assume $\mathbb{P}(\hat{m} = 1) > 0$. Then*

$$1 - \alpha - \rho_{\hat{m}} \;\le\; \mathbb{P}\big(y^*_{N+1} \in \mathcal{C}^{\mathrm{strat}}_{\lambda_0}(x_{N+1}) \,\big|\, \hat{m}(x_{N+1}) = 1\big) \;\le\; 1 - \rho_{\hat{m}}, \tag{12}$$

where $\rho_{\hat{m}} = \mathbb{P}(M = 0 \mid \hat{m} = 1)$ is the false matchability rate in the predicted-matchable stratum. Without conditioning on membership in the predicted-matchable stratum, coverage satisfies

$$\mathbb{P}\big(y^*_{N+1} \in \mathcal{C}^{\mathrm{strat}}_{\lambda_0}(x_{N+1})\big) \ge (1-\alpha)\,\mathbb{P}\big(\hat{m}(x_{N+1}) = 1\big) - \mathbb{P}\big(M = 0,\, \hat{m}(x_{N+1}) = 1\big). \tag{13}$$

Our bound complements two related formulations. Li et al. [18] concurrently study feasibility-aware coverage at the population level. Noorani et al. [19] estimate unseen probability mass after sampling and combine set construction with an adaptive query policy. Our taxonomy is instead a per-query prediction learned before generation, and $\rho_{\hat{m}}$ measures its errors within the selected population. Unseen mass for one sampled distribution need not identify whether that query's reference is absent (Table 2).

If $\rho_{\hat{m}} = 0$, the bound gives nominal coverage within the predicted-matchable stratum, which may still exclude some matchable queries. If $\rho_{\hat{m}} > \alpha$, the upper bound $1 - \rho_{\hat{m}}$ is below the nominal target $1 - \alpha$, so no threshold over sampled clusters can attain that target within the selected stratum. Using the learned matchability stratum as the Mondrian taxonomy expresses the support ceiling through a measurable parameter. We use $\lambda_0 = 0$ (the aleatoric-only baseline labeled *Vanilla CP*) for the empirical evaluation in Section 3. Appendix P describes the extension to $\lambda_0 = \lambda^\star$, and the proof and corollary on the upper bound are in Appendix B.4.

2.6 Diagnostics

The fitted $\lambda^\star$ controls how much within-clarification variability enters the score. An interpretation can be made only when the set-size objective distinguishes weights and the selected region is stable across splits. On a flat objective, neither a large coefficient nor a tie-breaking value of zero identifies the source of uncertainty. The per-query ratio

$$R(x) = \log(\lambda^\star U_{\mathrm{epi}}(x) + \epsilon) - \log(U_{\mathrm{ale}}(x) + \epsilon), \qquad \epsilon = 10^{-6}, \tag{14}$$

describes the relative weighted components. It is also affected by instability in the coefficient and loses its epistemic contribution when $\lambda^\star = 0$. We therefore evaluate U_{epi} directly for support failure ranking and treat R as a descriptive statistic. Clarification, retrieval, and abstention are possible downstream interventions, but their benefits must be evaluated separately and cannot be inferred from the sign of R.

3 Experiments

3.1 Experimental setup

Datasets. Six QA benchmarks, 2,104 questions total. *Factual:* TriviaQA (300) [20], OpenNQ (300) [21], SimpleQA (500) [22], PopQA (500) [23]. *Ambiguity benchmarks:* AmbigQA (200) [24],

AmbigInst (304) [14]. We use the curated subsets of Walha et al. [11] for TriviaQA, OpenNQ, AmbigQA and AmbigInst for direct comparability. The main table uses a 40% $\mathcal{V}_1$ / 30% $\mathcal{V}_2$ / 30% Test split. Multi-seed summaries use four random splits, with seed values listed in Appendix E. Stratified matchability experiments use a separate $\mathcal{V}_0/\mathcal{V}_2/\mathrm{Test}$ split. Full dataset descriptions are in Appendix D.

Models, baselines, metrics. The main open-weight 4–9B LLMs are **Qwen3-8B** [25], **Ministral-3-8B-Instruct-2512** [26] (FP8), and **Gemma-4-E4B-it** [27]. Each model self-generates $n = 5$ clarifications at temperature 0.7 and $m = 10$ answers per clarification at temperature 0.5. Answers are clustered with all-mpnet-base-v2 at $\delta = 0.7$ (Appendix G examines encoder sensitivity). The reference is matched to cluster representatives, and to raw cluster members for lexical tiers, by a multi-tier criterion combining substring, ROUGE-L, Jaccard, and embedding-cosine tests (Appendix E). The six set rules are **CAELUM** ($\lambda = \lambda^\star$, ours), **Vanilla CP** (the aleatoric-only weight $w = 1 + U_{\mathrm{ale}}$, i.e., $\lambda = 0$), **Equal-weight** ($\lambda = 1$), **Epistemic-only** ($\lambda = 10^6$), a **self-consistency proxy** (SC proxy), and **Semantic Entropy CP** (our APS baseline weighted by $1 + U_{\mathrm{total}}$). The SC proxy calibrates $1 - \max_k p_k$ and uses the resulting cumulative-mass cutoff. It is inspired by ConU [12], but does not reproduce ConU's reference-aligned candidate scores or set rule. Table 6 additionally reports **TrueAPS**, the textbook unweighted APS baseline $w(x) \equiv 1$, since the "Vanilla" label is still uncertainty-weighted by U_{ale}. The main table uses deterministic APS. In multi-seed comparisons, CAELUM, Vanilla, Equal-weight, Epistemic-only, and TrueAPS use randomized calibration scores, while the SE-CP baseline uses deterministic APS. Prediction sets remain deterministic. SE-CP and Equal-weight have the same weight whenever $U_{\mathrm{total}} = U_{\mathrm{ale}} + U_{\mathrm{epi}}$, and their different performance across seeds cannot be attributed to the decomposition alone. Hallucination ranking compares the four signals U_{total} (the spectral-entropy baseline of Walha et al. [11]), U_{epi}, U_{ale}, and $R(x)$. We report marginal coverage (target $1 - \alpha = 0.9$), APSS, AUROC, and $\lambda^\star$. The λ grid is specified in Appendix E. Additional analyses include Qwen3-14B and Qwen3-32B [28].

For Table 1, both CAELUM and the SC proxy calibrate their thresholds on the full $\mathcal{V}_2$ split and use the same unfiltered test questions, including unmatchable items. This differs from the support condition in ConU's correctness-coverage formulation (Appendix A).

3.2 Results

Support limits coverage. Small gains in set size may be possible. Table 1 summarizes experimental results. CAELUM prediction sets are smaller on average than Vanilla prediction sets in 9 of 18 cells, the same average size in 7 of 18 cells, and only slightly larger by no more than 0.01 in 2 of 18 cells. The largest reductions are 4.5% for Ministral–AmbigQA and about 3.3% for Qwen3–PopQA and Ministral–TriviaQA. The mean differences are smaller than variation across random splits, according to Table 6. SC proxy prediction sets have coverage that is up to 6.5 percentage points lower than CAELUM prediction sets. Comparisons of set sizes should consider differences in coverage. Analysis of post-hoc set-size comparisons in Table 8 shows no clear winners and identifies some infeasible coverage levels.

Test coverage for CP methods is close to the test support fraction in most cells (Figure 2). A prediction set can contain all sampled clusters for a given query. Once this happens, weighting cannot add a reference absent from those clusters. Although estimates of $\lambda^\star$ vary greatly by dataset and generator, the small differences in APSS and variability in $\lambda^\star$ across splits suggest that $\lambda^\star$ cannot reliably be interpreted as a measure of intrinsic ambiguity in a dataset.

Hallucination ranking. Table 9 presents AUROC for support failure classification, with "ground truth not in any cluster" as the positive class. AUROC for U_{epi} is higher than for U_{total} in 11 out of 17 non-degenerate cells in the main dataset split, with an average AUROC improvement of $+0.014$ and a maximum improvement of $+0.059$. Averaged over four random splits, AUROC for U_{epi} is higher than for U_{total} in 15 out of 17 non-degenerate cells, with an average improvement of $\bar{\Delta}_{\mathrm{EPI}} = +0.012$. We use U_{epi} as an indicator of support failures and $R(x)$ as a descriptive statistic for each query. See Appendix N for results of paired tests and other comparisons of AUROC with $R(x)$.

Coverage in the predicted-matchable subpopulation. We use Theorem 1 with $\lambda = 0$ (Vanilla-CP score) and a logistic matchability model $\hat{m}$ trained on $\mathcal{V}_0$ (30% of each cell). The inputs to $\hat{m}$ are question embeddings and surface features (Appendix E). The same fixed matchability model $\hat{m}$ is

Table 1: Main results with a target nominal 90% coverage rate (deterministic APS). Test matchability $\hat{\pi}_{\text{test}}$, the fraction of references matched to sampled clusters, and other test results are evaluated on the same held-out questions. **Bold** and underline designate the smallest and second-smallest displayed set sizes among methods with the highest test coverage in that row, including ties, and † marks sizes below that coverage. Equal/SE refers to numerically identical results from Equal-weight and SE-CP methods. SC proxy is a simple self-consistency heuristic. The fitted $\lambda^\star$ (deterministic seed SD) does not by itself identify an uncertainty regime. Full comparison results are reported in Table 5.

	Support / CAELUM			Average set size ↓				SC proxy	
Dataset	$\hat{\pi}_{\text{test}}$	Cov.	$\lambda^\star$	CAELUM	Vanilla	Equal/SE	Epi.-only	Cov.	Size
Qwen3-8B									
TriviaQA	0.811	0.811	47.42 (14.5)	1.82	1.84	1.84	**1.81**	0.778	1.02†
AmbigQA	0.667	0.667	7.54 (33.9)	**1.30**	**1.30**	**1.30**	**1.30**	0.650	1.00†
OpenNQ	0.700	0.700	0.00 (0.0)	1.20	1.20	1.20	1.20	0.700	**1.00**
AmbigInst	1.000	0.935	0.00 (0.0)	**1.00**	**1.00**	**1.00**	**1.00**	0.935	1.05
SimpleQA	0.367	0.367	0.00 (12.2)	1.77	1.77	1.77	**1.75**	0.327	1.02†
PopQA	0.533	0.533	3.15 (38.7)	**1.74**	1.80	1.79	1.75	0.520	1.02†
Ministral-3-8B-Instruct-2512									
TriviaQA	0.811	0.811	26.96 (25.5)	**2.27**	2.34	2.37	2.28	0.767	1.07†
AmbigQA	0.733	0.733	18.41 (24.5)	**3.90**	4.08†	4.05†	3.92	0.717	1.90†
OpenNQ	0.733	0.733	6.19 (14.8)	**1.61**	1.63	1.63	1.60†	0.711	1.00†
AmbigInst	0.880	0.859	0.00 (25.5)	**2.32**	**2.32**	2.75	2.99	0.859	2.63
SimpleQA	0.340	0.340	0.32 (20.9)	2.31	2.30	2.31	**2.29**	0.287	1.00†
PopQA	0.487	0.487	1.43 (22.5)	**3.57**	3.62	**3.57**	**3.57**	0.447	1.12†
Gemma-4-E4B-it									
TriviaQA	0.678	0.678	45.74 (13.6)	1.74	1.77	1.77	**1.73**	0.656	1.01†
AmbigQA	0.633	0.633	9.32 (5.2)	**2.17**	2.18	**2.17**	**2.17**	0.583	1.05†
OpenNQ	0.556	0.556	0.00 (15.0)	**1.51**	**1.51**	**1.51**	**1.51**	0.544	1.00†
AmbigInst	0.935	0.935	6.17 (18.9)	1.96	1.97†	1.99	**1.89**	0.870	1.42†
SimpleQA	0.207	0.207	0.28 (0.1)	**1.66**	**1.66**	1.67	1.67	0.180	1.00†
PopQA	0.373	0.373	8.51 (7.9)	1.94	**1.93**	1.95	**1.93**	0.347	1.02†

used for $\mathcal{V}_2$ (30%) and Test (40%). Using $\lambda = 0$ separates the effect of stratification by matchability from the effect of λ weighting in Table 1. A $\lambda^\star$-weighted variant is described in Appendix P.

Across 30 cells (the 18 cells of Table 1 plus Qwen3-14B and Qwen3-32B on the same six datasets) and 4 random seeds ($n = 120$ cell×seed evaluations, omitting the duplicate 50-question AmbigInst cache), estimated prediction-set coverage in the learned matchability strata is 0.74 ± 0.15, and overall estimated coverage is 0.65 ± 0.17 (78 rows from 27 cells, learned classifier $\mathcal{V}_2$ AUC 0.67 ± 0.08). Rows with only one class or $\mathcal{V}_2$ AUC less than 0.55 are labeled oracle-fallback reference results rather than learned-classifier results. Three cells (Qwen3-8B on AmbigInst and SimpleQA, Qwen3-14B on SimpleQA) use the oracle fallback on all four seeds. Including oracle-fallback rows, estimated coverage in matchability strata is 0.83 ± 0.17, and overall estimated coverage is 0.65 ± 0.21. The $\pm$ values are standard deviations across the corresponding cell×seed evaluations.

Mean coverage gains across all rows include $+20$ percentage points for PopQA across the five models and $+12$ percentage points for TriviaQA. These summaries include oracle fallbacks. The largest learned gains occur for PopQA data. Despite these gains, 19 of the 20 model and seed evaluations on PopQA fail the empirical coverage check, which uses a heuristic tolerance. This check is distinct from the coverage bound in Theorem 1 using $\rho_{\hat{m}}$. Neither more samples nor escalation to larger models closes this apparently knowledge-bound gap. Other information about operating points and deployment challenges is available in Appendix S.2. Point estimates for individual cells are available in Figure 4 (Appendix Q) and the supplement.

Cross-LLM transfer of matchability. To assess whether $\hat{m}$ reflects question difficulty shared across models or issues specific to the source LLM, we fit logistic regression models predicting match

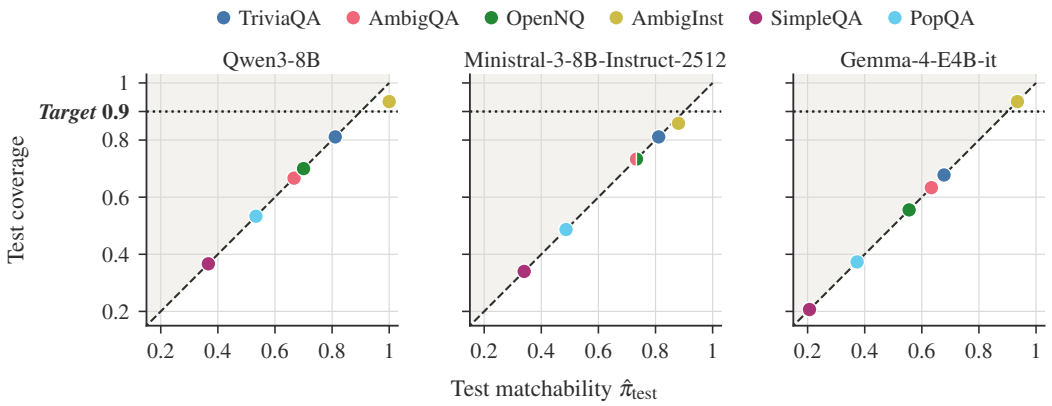


Figure 2: Test coverage and matchability for 18 main model–dataset cells, one panel per LLM. Colors identify datasets. The shaded region above the diagonal is unreachable using sampled clusters. A two-color circle marks overlapping datasets. The horizontal line marks the 0.9 target.

labels for one LLM on 2,104 questions in our six datasets and apply these models to predict match labels for another LLM. Appendix R reports all six cross-generator directions with bootstrap intervals. The average in-sample AUC for these tasks is 0.80, and the average cross-generator AUC is 0.78 ($\bar{\Delta} = -0.02$). Overall, observed label agreement is 0.81, well above independent-label agreement levels of 0.50–0.53 at the observed marginal match rates of the two LLMs. The 95% bootstrap lower confidence bound on target-label AUC for all six cross-generator directions exceeds 0.74 (see Table 11 and Appendix R for complete confidence intervals). These results suggest some evidence of cross-generator transfer of match labels for these questions. Note that the logistic regression models use the same question records for both fitting and scoring. We are not assessing cross-generator transfer to new questions and topics.

Selective answering. Used as a routing signal (rank inputs by score and route the top-k to a simulated oracle), $R(x)$ yields AUC improvements of $+0.040$ to $+0.061$ over random routing on three TriviaQA cells and $+0.052$ AUC improvement on Qwen3–AmbigQA, but no improvement over random routing on some tasks with low values of $\lambda^\star$. Here, AUC measures the area under the accuracy curve as the escalation rate varies. U_{epi} yields more consistent improvements ($17/18$ tasks, per-task change in AUC of $[-0.005, +0.071]$, mainly in the range $[+0.02, +0.07]$). See Table 10 (Appendix O) for details.

Detection and abstention. At a cost of 50 generated responses per query, the learned CAELUM detector adds an abstention label less often than two baseline rules, Good–Turing and CPQ. The average augmented coverage of the CAELUM detector is greater than 0.9, as detailed in Table 2. The abstention label contributes to augmented coverage if the reference answer does not match any sampled cluster. Test results for 29 model–dataset cells with judged support labels are separate from the pre-generation classifier studied in Theorem 1. For more information about experimental procedures and measures of coverage, see Appendix S.7.

Table 2: Detection and abstention on 29 cells. CAELUM uses the variant in Appendix K. Means are over four splits per cell, then over cells. Parentheses count cells with defined precision. **Bold** and underline mark the best and second-best AUROC, abstention rate, and precision. Arrows indicate preferred directions. The augmented coverage target is 0.9.

Trigger signal / rule	AUROC ↑	Abstention rate ↓	Abstention precision ↑	Aug. coverage
CAELUM detector (ours)	**0.780**	**0.459**	**0.569** (28)	0.924
U_{epi} trigger [11]	0.696	0.506	0.481 (28)	0.918
Good–Turing trigger [19]	0.550	0.878	0.371 (28)	0.989
CPQ calibration module [19]	—	0.848	0.384 (27)	0.987

4 Discussion

Support and calibration have different goals. A score can satisfy a conformal rank bound without the prediction set covering references absent from sampled support. Proposition 1 addresses calibration conditional on true support. Theorem 1 demonstrates how false matchability limits coverage in a learned stratum. The difference between overall coverage of 0.65 and learned-stratum coverage of 0.74 compares different populations within the same evaluations, not an overall 90% coverage rate. Both classifier precision and selection rate should be reported.

Weight stability and uncertainty ranking. Mean cell APSS values for Vanilla and CAELUM are similar across four random seeds, each being about 2.21. The mean difference in APSS, Vanilla minus CAELUM, is about -0.005. Differences between Vanilla and CAELUM are smaller than variation across splits in every cell. Flatness of the APSS grid search objective function can result in large differences in estimated coefficients and small differences in prediction sets. We find that in a 30-cell experiment, an average of 3.6% of test items have different set sizes across the examined grid weights up to $\lambda = 100$ (Appendix S.4). Results support the movement diagnostic rather than a regime-based interpretation of every $\lambda^\star$ value. U_{epi} performs slightly better than U_{total} on average in ranking support failures. R is less consistent and does not provide a good substitute for U_{epi}.

Scale of generator and interventions. A comparison using new generations for the same questions at the same sampling budget indicates that pooled heuristic match rates increase with model size from 61.1% for Qwen3-8B to 66.0% for Qwen3-32B. Match rates for Qwen3-32B are 36.6% for SimpleQA and 57.4% for PopQA. Separate 14B and 32B experiments inform transfer and ranking criteria, and coverage tracks $\hat{\pi}$ within 0.030 on all 30 judged cells (Appendix S.5). There are advantages and disadvantages to using larger generator models. Adding an abstention label to a prediction set allows augmented coverage to be evaluated alongside ordinary semantic coverage. Prediction sets with an abstention option have a different output space. Reports should include ordinary semantic coverage, augmented coverage, and abstention rate to evaluate coverage and abstention together.

5 Conclusion, Limitations, and Future Work

Conclusion. CAELUM applies kernel von Neumann uncertainty quantification and conformal methods to produce semantic prediction sets for open-ended natural language problems. Across four random splits of 30 model–dataset combinations, matchability, or the proportion of questions whose reference matches a sampled cluster, limits achievable coverage. Mathematical results in this paper, such as Theorem 1, give bounds on coverage that incorporate a penalty term for false matchability, $\rho_{\hat{m}}$. Classifiers trained on one generator have a mean transfer AUC of 0.78 when used on the same questions from a different generator. U_{epi} performs better than total entropy on average at ranking support failures.

Limitations and future work. The decomposition method measures uncertainty within the clarification ensemble, which need not correspond to ground-truth ambiguity. The association between U_{ale} and labeled ambiguity is weak. Additional ambiguity audit results show that U_{ale} achieves an AUROC of 0.53–0.63. CAELUM and Vanilla have similar set sizes across splits, and there can be several values of $\lambda^\star$ that minimize the objective function, so a selected value need not identify a particular uncertainty regime. Improvements in answer generation may increase matchability, while changes in match evaluation may alter its measured value. The judge audit revisits nonmatches identified heuristically but does not validate all positive matches. Learned-stratum coverage on some factoid datasets, including PopQA, remains below the nominal target. Transfer evaluations reuse questions across generators. Generating 50 answers per question is time-consuming. Current work focuses on generating short-answer QA responses with external references. We do not evaluate long-form responses or multi-turn agentic settings. One potential extension is a multimodal approach that uses source disagreement and other information about modalities to create calibration strata, as in modality-aware conformal regression by Azizi [29]. Audits within each stratum would be needed to assess whether the reference answer is present in the sampled support. Treating prediction sets with limited support as having unconditional nominal coverage may encourage users to trust generated answers too readily. High refusal rates can prevent users from accessing potentially useful information.

Acknowledgments and Disclosure of Funding

This work used computational resources from the DCSR clusters at the University of Lausanne. BegooAI supported the LLM-as-judge evaluations by providing Gemini API keys and covering the associated API usage costs. The author has no additional competing financial interests to disclose.

References

[1] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. URL https://arxiv.org/abs/2303.08774.

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL https://arxiv.org/abs/2302.13971.

[3] Sebastian Farquhar, Jannik Kossen, Lorenz 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://www.nature.com/articles/s41586-024-07421-0.

[4] Michael Kirchhof, Gjergji Kasneci, and Enkelejda Kasneci. Position: Uncertainty quantification needs reassessment for large language model agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 81665–81677. PMLR, 2025. URL https://proceedings.mlr.press/v267/kirchhof25b.html.

[5] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005. doi: 10.1007/b106715.

[6] Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: 10.1561/2200000101. URL https://doi.org/10.1561/2200000101.

[7] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html.

[8] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021. doi: 10.1007/s10994-021-05946-3. URL https://arxiv.org/abs/1910.09457.

[9] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

[10] 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, pages 8901–8929, 2024. doi: 10.52202/079017-0283. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/10c456d2160517581a234dfde15a7505-Abstract-Conference.html.

[11] Nassim Walha, Sebastian G. Gruber, Thomas Decker, Yinchong Yang, Alireza Javanmardi, Eyke Hüllermeier, and Florian Buettner. Fine-grained uncertainty decomposition in large language models: A spectral approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 26090–26098, 2026. doi: 10.1609/aaai.v40i31.39811. URL https://ojs.aaai.org/index.php/AAAI/article/view/39811. Issue 31.

[12] Zhiyuan Wang, Jinhao Duan, Lu Cheng, Yue Zhang, Qingni Wang, Xiaoshuang Shi, Kaidi Xu, Heng Tao Shen, and Xiaofeng Zhu. ConU: Conformal uncertainty in large language models with correctness coverage guarantees. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6886–6898. Association for Computational Linguistics,

2024. doi: 10.18653/v1/2024.findings-emnlp.404. URL https://aclanthology.org/2024.findings-emnlp.404/.

[13] Ilia Azizi, Juraj Bodik, Jakob Heiss, and Bin Yu. CLEAR: Calibrated learning for epistemic and aleatoric risk. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fff035a88e93fff416f66f8a52dd9067-Abstract-Conference.html.

[14] Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decomposing uncertainty for large language models through input clarification ensembling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 19023–19042. PMLR, 2024. URL https://proceedings.mlr.press/v235/hou24b.html.

[15] Yaniv Romano, Matteo Sesia, and Emmanuel Candès. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems*, volume 33, pages 3581–3591, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/244edd7e85dc81602b7615cd705545f5-Abstract.html.

[16] Vladimir Vovk, David Lindsay, Ilia Nouretdinov, and Alex Gammerman. Mondrian confidence machine. On-line Compression Modelling Project, Working Paper No. 4, 2003. URL https://alrw.net/old/04.pdf.

[17] Henrik Boström and Ulf Johansson. Mondrian conformal regressors. In *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 128 of *Proceedings of Machine Learning Research*, pages 114–133. PMLR, 2020. URL https://proceedings.mlr.press/v128/bostrom20a.html.

[18] Ye Li, Anqi Hu, Yuanchang Ye, Shiyan Tong, Zhiyuan Wang, and Bo Fu. Set-valued prediction for large language models with feasibility-aware coverage guarantees. *arXiv preprint arXiv:2603.22966*, 2026. URL https://arxiv.org/abs/2603.22966.

[19] Sima Noorani, Shayan Kiyani, George J. Pappas, and Hamed Hassani. Conformal prediction beyond the seen: A missing mass perspective for uncertainty quantification in generative models. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-1084. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2e5911cb61db978c225cf62b6c029192-Abstract-Conference.html.

[20] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

[21] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, et al. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

[22] Jason Wei, Karina Nguyen, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024. URL https://arxiv.org/abs/2411.04368.

[23] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 9802–9822. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

[24] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 5783–5797. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.466. URL https://aclanthology.org/2020.emnlp-main.466/.

[25] Qwen Team. Qwen3-8B model card. Hugging Face model card, 2025. URL https://huggingface.co/Qwen/Qwen3-8B.

[26] Mistral AI. Ministral-3-8B-Instruct-2512 model card. Hugging Face model card, 2025. URL https://huggingface.co/mistralai/Ministral-3-8B-Instruct-2512.

[27] Google DeepMind. Gemma 4 model card: gemma-4-E4B-it. Hugging Face model card, 2026. URL https://huggingface.co/google/gemma-4-E4B-it.

[28] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[29] Ilia Azizi. Conformal calibration for multi-modal regression with missing modalities. In *Proceedings of the Fifteenth Symposium on Conformal and Probabilistic Prediction with Applications*, volume 329 of *Proceedings of Machine Learning Research*, pages 343–377. PMLR, 2026. URL https://proceedings.mlr.press/v329/azizi26a.html.

[30] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

[31] Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1184–1193. PMLR, 2018. URL https://proceedings.mlr.press/v80/depeweg18a.html.

[32] Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 2282–2292. PMLR, 2023. URL https://proceedings.mlr.press/v216/wimmer23a.html.

[33] Freddie Bickford Smith, Jannik Kossen, Eleanor Trollope, Mark Van Der Wilk, Adam Foster, and Tom Rainforth. Rethinking aleatoric and epistemic uncertainty. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 4345–4359. PMLR, 2025. URL https://proceedings.mlr.press/v267/bickford-smith25a.html.

[34] Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. To believe or not to believe your LLM: Iterative prompting for estimating epistemic uncertainty. In *Advances in Neural Information Processing Systems*, volume 37, pages 58077–58117, 2024. doi: 10.52202/079017-1852. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6aebba00fff5b6de7b488e496f80edd7-Abstract-Conference.html.

[35] 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://proceedings.iclr.cc/paper_files/paper/2026/hash/83e9c3ea5d0b5de331708058637ada10-Abstract-Conference.html.

[36] Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, Guangji Bai, Liang Zhao, and Haifeng Chen. Uncertainty quantification for in-context learning of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, pages 3357–3370. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.184. URL https://aclanthology.org/2024.naacl-long.184/.

[37] Gustaf Ahdritz, Aravind Gollakota, Parikshit Gopalan, Charlotte Peale, and Udi Wieder. Provable uncertainty decomposition via higher-order calibration. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/122ea6470232ee5e79a2649243348005-Abstract-Conference.html.

[38] Adam X. Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/07c256a163a7559186ec1c71e95b9ec9-Abstract-Conference.html.

[39] Yibin Wang, Haizhou Shi, Ligong Han, Dimitris Metaxas, and Hao Wang. BLoB: Bayesian low-rank adaptation by backpropagation for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 67758–67794, 2024. doi: 10.52202/079017-2164. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7d53575463291ea6b5a23cf6e571f59b-Abstract-Conference.html.

[40] Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/31421b112e5f7faf4fc577b74e45dab2-Abstract-Conference.html.

[41] Bhawesh Kumar, Charlie Lu, Gauri Gupta, Anil Palepu, David Bellamy, Ramesh Raskar, and Andrew Beam. Conformal prediction with large language models for multi-choice question answering. *arXiv preprint arXiv:2305.18404*, 2023. URL https://arxiv.org/abs/2305.18404.

[42] Ahmed Karim, Qiao Wang, and Zheng Yuan. Beyond the score: Uncertainty-calibrated LLMs for automated essay assessment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19631–19636. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.992. URL https://aclanthology.org/2025.emnlp-main.992/.

[43] Kwan Ho Ryan Chan, Yuyan Ge, Edgar Dobriban, Hamed Hassani, and René Vidal. Conformal information pursuit for interactively guiding large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3021. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/823a3d2cf462fb815978314023e48f65-Abstract-Conference.html.

[44] Yusuf Sale, Alireza Javanmardi, and Eyke Hüllermeier. Aleatoric and epistemic uncertainty in conformal prediction. In *Proceedings of the Fourteenth Symposium on Conformal and Probabilistic Prediction with Applications*, volume 266 of *Proceedings of Machine Learning Research*, pages 784–786. PMLR, 2025. URL https://proceedings.mlr.press/v266/sale25a.html.

[45] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

[46] Jakob Heiss, Sören Lambrecht, Jakob Weissteiner, Hanna Wutte, Žan Žurič, Josef Teichmann, and Bin Yu. JUCAL: Jointly calibrating aleatoric and epistemic uncertainty in classification tasks. *arXiv preprint arXiv:2602.20153*, 2026. URL https://arxiv.org/abs/2602.20153.

[47] Ilia Azizi, Marc-Olivier Boldi, and Valérie Chavez-Demoulin. SEMF: Supervised expectation-maximization framework for predicting intervals. In *Proceedings of the Fourteenth Symposium on Conformal and Probabilistic Prediction with Applications*, volume 266 of *Proceedings of Machine Learning Research*, pages 250–281. PMLR, 2025. URL https://proceedings.mlr.press/v266/azizi25a.html.

[48] Luben Miguel Cruz Cabezas, Vagner Silva Santos, Thiago Ramos, and Rafael Izbicki. Epistemic uncertainty in conformal scores: A unified approach. In *Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pages 443–470. PMLR, 2025. URL https://proceedings.mlr.press/v286/cruz-cabezas25a.html.

[49] Alireza Javanmardi, Soroush H. Zargarbashi, Santo M. A. R. Thies, Willem Waegeman, Aleksandar Bojchevski, and Eyke Hüllermeier. Optimal conformal prediction under epistemic uncertainty. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pages 2461–2479. PMLR, 2026. URL https://proceedings.mlr.press/v337/javanmardi26a.html.

[50] Jiayi Xin, Evan Qiang, Zihan Zhu, Xiang Li, Weijie J. Su, and Qi Long. Conformal prediction with paraphrase-aware scoring for LLM uncertainty quantification. *arXiv preprint arXiv:2610.04239*, 2026. URL https://arxiv.org/abs/2610.04239.

[51] Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021. doi: 10.1093/imaiai/iaaa017. URL https://arxiv.org/abs/1903.04684. First published online in 2020.

[52] Ikram Belmadani, Oumaima El Khettari, Pacôme Constant dit Beaufils, Richard Dufour, and Benoit Favre. Who judges the judge? Evaluating LLM-as-a-judge for French medical open-ended QA. In *Proceedings of the 1st Workshop on Linguistic Analysis for Health*, pages 142–157. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.healing-1.12. URL https://aclanthology.org/2026.healing-1.12/.

A Extended related work

Uncertainty estimation in LLMs. Token-level probabilities fail to capture semantic confidence. Paraphrases of the same answer can have very different likelihoods [9]. Semantic entropy [9] clusters generations via bidirectional entailment and computes entropy over clusters, and Farquhar et al. [3] scale this to 30 model-task combinations, using it to detect confabulations. Nikitin et al. [10] generalize this construction with kernel-based von Neumann entropy on a positive-semidefinite semantic-similarity kernel, and Manakul et al. [30] detect hallucinations via cross-response consistency without token probabilities. These methods yield uncertainty scores for ranking, but give no coverage guarantees and no signal on the source of uncertainty.

Uncertainty decomposition for LLMs. The classical decomposition $H(Y|X) = \mathbb{E}_\theta[H(Y|X,\theta)] + \mathcal{I}(Y;\theta|X)$ [8, 31] marginalizes over model parameters, which is infeasible for large LLMs and has been criticized in the parametric setting [32, 33]. Hou et al. [14] sidestep both issues with Input Clarification Ensembling (ICE), replacing parameter variability with input-context variability so the procedure stays in the output space. Walha et al. [11] extend this with von Neumann entropy in a kernel-induced semantic space, outperforming ICE on ambiguity detection and correctness prediction. Other routes include information-theoretic epistemic estimators via iterative prompting [34], inter-model disagreement [35], mutual-information decomposition for in-context learning [36], and provable decomposition via higher-order calibration [37]. Weight-space Bayesian methods like Laplace-LoRA and BLoB [38, 39] require fine-tuning and do not apply to frozen LLMs. Kirchhof et al. [4] argue that uncertainty quantification for interactive LLM agents must account for underspecification, interaction, and how uncertainty is communicated. CAELUM returns prediction sets together with uncertainty diagnostics. Our analysis studies the generated-support limitation when these uncertainty signals are used inside conformal set construction.

Conformal prediction for language models. Conformal prediction [5, 6] gives distribution-free coverage under exchangeability. Quach et al. [40] calibrate a sampling stopping rule and a rejection rule for sets of sampled open-ended responses, a sequence-level candidate-set mechanism rather than token-level conformal prediction. Kumar et al. [41] construct prediction sets for multiple-choice QA, and Karim et al. [42] calibrate LLM-based essay scoring. Conformal Information Pursuit [43] uses conformal set sizes to drive interactive question answering, but treats uncertainty as a single

signal. Wang et al. [12] propose ConU, applying conformal prediction to free-form generation via self-consistency without decomposing uncertainty. Sale et al. [44] analyze how aleatoric and epistemic uncertainty enter conformal prediction. CAELUM pairs conformal set construction for LLMs with a decomposed uncertainty weight. Concurrent work by Li et al. [18] constructs set-valued predictions for LLMs with feasibility-aware coverage at the population level. Noorani et al. [19] allow prediction sets to include a catch-all element for unseen outputs, using Good–Turing estimates of the missing probability mass. In our case, in Theorem 1 we condition on a learned stratum of queries predicted to be matchable before generation. We account for false matchability in this stratum through $\rho_{\hat{m}}$. In our experiments with a fixed sampling budget, the Good–Turing score detected support failure at AUROC 0.550, while our learned detector operating after generation achieved 0.780 (see Table 2).

ConU and the self-consistency proxy. ConU [12] scores each sampled answer using its semantic frequency and its agreement with other answers. During calibration, it takes the score of a candidate that matches the reference and computes a conformal quantile of these scores. At test time, it includes each candidate whose score falls below or equals that threshold. Its correctness-coverage formulation requires a correct candidate for every calibration query and assumes one exists for every test query. The guarantee therefore relies on the answer already being present among the samples; it does not cover references missing from the sampled support.

Our SC proxy instead takes a quantile of $1 - \max_k p_k$ over all calibration questions and uses it as a cumulative-mass cutoff over frequency-ranked clusters. The calibration score does not use the identity of the correct cluster, so this quantile alone gives no correctness-coverage guarantee. The proxy can return small sets when most sampled answers fall in one cluster, whether or not that cluster is correct. Its coverage and set size in Table 1 are empirical results on the full test split. Comparisons with ConU require the same generated answers, correctness criterion, evaluation population, and set-size units.

Joint calibration of aleatoric and epistemic uncertainty. Temperature scaling is a scalar post-hoc calibration method [45]. Because it has only one control parameter, it does not by itself provide independent controls for aleatoric and epistemic components. CLEAR jointly calibrates aleatoric and epistemic uncertainty in regression. Its conformalized variant selects a mixture coefficient on a validation split and a final scale on a separate calibration split [13]. JUCAL jointly optimizes two probability-calibration parameters by negative log-likelihood for classifier ensembles, whose outputs can subsequently be conformalized [46]. Azizi et al. [47] introduce SEMF for predictive intervals through supervised expectation maximization, also in a scalar output space. CAELUM adapts the split-calibration construction of conformalized CLEAR to estimates from an LLM, using a *prompt ensemble* of n clarifications of the same query [14], each producing m samples. Following ICE, mean within-clarification entropy is the epistemic proxy, and the pooled-minus-within residual is the aleatoric proxy. The relevant prior theory has distinct scopes. Sale et al. [44] analyze the aleatoric–epistemic distinction in conformal prediction, Cruz Cabezas et al. [48] place epistemic uncertainty in conformal scores, and Javanmardi et al. [49] optimize under second-order epistemic information. Xin et al. [50] also vary the input via paraphrasing in conformal prediction, but apply it to multiple-choice QA without a decomposition step.

A.1 Comparison with CLEAR and JUCAL

Table 3: Joint calibration of aleatoric and epistemic uncertainty across output spaces.

	CLEAR	**JUCAL**	**CAELUM**
Output space	$\mathbb{R}$	$\{1, \ldots, K\}$	Open-ended text
Output	Interval	Calibrated probabilities	Semantic clusters
Aleatoric proxy	Quantile regression	Temperature (c_1)	Between-clarification entropy
Epistemic proxy	Ensemble diversity	Diversity adjustment (c_2)	Within-clarification entropy
Calibration loss	Quantile loss	NLL	Set size (APSS)
Requires ensemble	Yes	Yes	No (single LLM)

B Theoretical analysis

B.1 Proof of Proposition 1 (support-conditional finite-sample coverage)

Assumption 1 (Exchangeability of items, generations, and matchability)**.** The calibration items and the test point are i.i.d. draws of the augmented tuple $(x, y^*, \mathcal{G}(x))$, where $\mathcal{G}(x)$ denotes the LLM's generated clarifications, sampled answers, semantic clusters, and seeded randomized-APS uniforms produced by the same fixed pipeline applied independently to each item. Matchability $M(x, y^*, \mathcal{G}(x))$, written $M(x, y^*)$ in Section 2.1, is computed from this tuple. Conditioning on the post-generation event $\{M = 1\}$ therefore preserves exchangeability of the conditional sub-sample. For Proposition 1, the stratum is true matchability $\{M = 1\}$. For Theorem 1, the stratum is the learned event $\{\hat{m}(x) = 1\}$, where $\hat{m}$ is trained on $\mathcal{V}_0$ and uses only x.

Under Assumption 1, the support-conditional guarantee of Proposition 1 holds with the usual split-conformal finite-sample slack on the quantile level.

Remark 1 (Fixed preprocessing). The support-conditional guarantee requires that the nonconformity score construction is fixed with respect to $\mathcal{V}_2$ except for the exchangeable calibration scores themselves. This holds because (a) $\lambda^\star$ is selected on $\mathcal{V}_1$ alone; (b) the sentence encoder ϕ and clustering threshold δ are fixed independently of $\mathcal{V}_2$ and the test point (they may be chosen on $\mathcal{V}_1$); (c) each instance's uncertainty estimates are computed from its own generated answers, with no cross-instance dependencies; and (d) randomized APS uses independent seeded uniforms, which are part of the exchangeable score randomness. If any of these components were fine-tuned on $\mathcal{V}_2$, the exchangeability argument would break.

Proof. Let $\mathcal{V}_2^M = \{i \in \mathcal{V}_2 : M(x_i, y_i^*, \mathcal{G}(x_i)) = 1\}$ be the calibration subset whose reference answers fall in the generated support, and let $n_M = |\mathcal{V}_2^M|$. For this support-restricted variant define

$$\hat{\tau}_M = Q_{\lceil (1-\alpha)(n_M+1) \rceil / n_M}\big(\{\tilde{s}_{\lambda^\star, i} : i \in \mathcal{V}_2^M\}\big).$$

Once $\lambda^\star$ is fixed from $\mathcal{V}_1$, the normalized nonconformity score $\tilde{s}_{\lambda^\star}(x, y^*) = s(x, y^*)/w_{\lambda^\star}(x)$ is fixed relative to $\mathcal{V}_2^M$. For the smoothed-randomized APS we replace s by s_{rand}, which is part of $\mathcal{G}(x)$ via the seeded uniforms, and treat the uniforms as part of the exchangeable score randomness. By Assumption 1, conditioning on $\{M = 1\}$ preserves exchangeability, so the scores $\{\tilde{s}_i : i \in \mathcal{V}_2^M\} \cup \{\tilde{s}_{N+1} : M_{N+1} = 1\}$ are exchangeable. The conformal quantile $\hat{\tau}_M$ ensures $\mathbb{P}(\tilde{s}_{N+1} \le \hat{\tau}_M \mid M = 1) \ge 1 - \alpha$. The bound conditions on n_M and is uniform in its realization by exchangeability of the ranks. On $\{M = 1\}$, the deterministic APS event $\tilde{s} \le \hat{\tau}_M$ is equivalent to $s \le \hat{\tau}_M \cdot w_{\lambda^\star}(x)$ and implies that the matching cluster is included in $\mathcal{C}_{\lambda^\star}(x)$. For the smoothed-randomized score, $s_{\text{rand}} \ge s$, so the randomized event $\{\tilde{s}^{\text{rand}} \le \hat{\tau}_M\}$ is a sufficient condition for $\{s \le \hat{\tau}_M\, w_{\lambda^\star}(x)\}$ and hence for inclusion (we sacrifice some efficiency for deterministic prediction sets). When $\eta_{\lambda^\star}(x; \hat{\tau}_M) = 1$, Equation (7) contains every generated cluster, including the matching cluster on $\{M = 1\}$, so the implication still holds at saturation. This yields the support-conditional lower bound. □

Remark 2 (Coverage over joint randomness). The APS nonconformity score $s(x, y^*)$ depends on the LLM's stochastic generation (clarifications and answers sampled with temperature > 0). The support-conditional guarantee in Proposition 1 holds over *both* the randomness of the calibration/test split *and* the LLM's sampling randomness, conditional on the generated support containing the answer. The nonconformity scores are therefore not deterministic functions of (x, y^*) alone.

Remark 3 (Role of the weight function). The weight function $w_\lambda(x)$ serves two purposes. First, by normalizing the raw scores as in Equation (8) before computing the conformal quantile, it makes $\hat{\tau}_\lambda$ depend on λ, enabling the optimization in Step 1 to distinguish between different uncertainty weightings. Without normalization, the calibration scores are λ-independent so $\hat{\tau}$ is constant. The per-item threshold $\eta_\lambda(x; \hat{\tau}) = \min(\hat{\tau} \cdot w_\lambda(x), 1)$ is then monotonically non-decreasing in λ (because w_λ is, for $U_{\text{epi}} \ge 0$), so APSS is monotonically non-decreasing and the argmin returns $\lambda^\star = 0$. Second, through the prediction set criterion in Equation (7), the weight scales the effective threshold, expanding sets more aggressively for high-uncertainty inputs and redistributing set sizes across inputs. This mirrors the role of two-parameter joint calibration in conformalized CLEAR [13], where a calibration constant varies with the aleatoric/epistemic mixing coefficient to balance coverage and set efficiency.

B.2 On asymptotic score stability (informal remark)

Remark 4 (Asymptotic score stability). Exact *conditional* coverage cannot generally be guaranteed in a distribution-free, finite-sample setting without making the prediction sets uninformative [51]. We do not claim it. As a heuristic asymptotic, if (i) the number of independently generated answers $m \to \infty$, (ii) the answer partition is fixed and finite, with the clarifications held fixed, and (iii) calibration and test data are i.i.d., then by the law of large numbers the empirical cluster proportions $p_k(x)$ converge to their population counterparts, and their Shannon entropies converge by continuity. Convergence of VNE in an infinite-dimensional kernel space requires additional assumptions. Convergence of the conformal quantile $\hat{\tau}$ also requires the calibration sample size to grow and regularity at the population quantile. We do not establish those convergence results here. This does not imply distribution-free conditional coverage for every fixed x. Our greedy clustering is a computationally efficient approximation that is order-dependent. Condition (ii) therefore need not hold. We report empirical coverage in Table 1 and validate per-cell behavior rather than relying on this asymptotic remark.

B.3 Coverage given matchability

Proposition 1 is conditional on generated support. An item x is *matchable* if some generated cluster $G_k(x)$ contains the reference answer in the sense of Appendix E. Unmatched items receive the maximal nonconformity score $s = 1$, are not included in any prediction set, and are therefore never covered by construction. Letting π denote the population matchability rate, $\mathbb{P}(y^*_{N+1} \in \mathcal{C}_{\lambda^\star}(x_{N+1})) \leq \pi$ regardless of α. Figure 2 compares coverage and empirical support on the same test split. Table 1 reports the match rate on that same test split. The support ceiling π alone precludes nominal $1 - \alpha = 0.9$ coverage on hard cells (e.g. Gemma-SimpleQA, $\hat{\pi}_{\text{test}} = 0.21$). Strengthening the generation or matching pipeline (retrieval, stronger generators, NLI-based equivalence, LLM judges) can lift π, but it lifts the ceiling for all conformal methods rather than uniquely improving CAELUM. Theorem 1 (Section 2.5) makes this dependence explicit by replacing population matchability with a learnable stratum and a false matchability penalty.

B.4 Proof of Theorem 1 (stratified coverage)

Proof of Theorem 1. Let $\hat{m} : \mathcal{X} \to \{0, 1\}$ be a binary classifier trained on $\mathcal{V}_0$, which is disjoint from both $\mathcal{V}_2$ and the test point. Define the predicted-matchable stratum $\mathcal{S} = \{i \in \mathcal{V}_2 : \hat{m}(x_i) = 1\}$. Because $\hat{m}$ depends on x only through a fixed function (no calibration or test labels) and was trained on a disjoint split, $\hat{m}(\cdot)$ is constant in distribution under any permutation of $\mathcal{V}_2 \cup \{(x_{N+1}, y^*_{N+1})\}$.

Score coverage on the stratum follows from Mondrian split conformal prediction [16, 17] applied to the taxonomy $\{\hat{m} = 0, \hat{m} = 1\}$. Specifically, conditioning on the event $\hat{m}(x_{N+1}) = 1$, the calibration sub-sample $\mathcal{S}$ together with the test point is exchangeable. Their joint distribution is invariant to permutations among items with $\hat{m} = 1$. Applying split conformal to this conditional distribution with the conformal quantile $\hat{\tau}_{\text{strat}}$ defined in Equation (11) yields

$$\mathbb{P}\big(\tilde{s}_{\lambda_0}(x_{N+1}, y^*_{N+1}) \leq \hat{\tau}_{\text{strat}}(\lambda_0) \,\big|\, \hat{m}(x_{N+1}) = 1\big) \geq 1 - \alpha,$$

where $\tilde{s} = 1/w_{\lambda_0}(x)$ for unmatched hard-score items. Let M denote true matchability and $\rho_{\hat{m}} = \mathbb{P}(M = 0 \mid \hat{m} = 1)$. For deterministic APS, coverage is the event $\{M = 1, \tilde{s} \leq \hat{\tau}_{\text{strat}}(\lambda_0)\}$. With randomized calibration scores and deterministic prediction sets, this score event is a subset of coverage because $s_{\text{rand}} \geq s$. In both cases

$$\begin{aligned}\mathbb{P}(\text{covered} \mid \hat{m} = 1) &\geq \mathbb{P}(\tilde{s} \leq \hat{\tau}_{\text{strat}} \mid \hat{m} = 1) - \mathbb{P}(M = 0, \tilde{s} \leq \hat{\tau}_{\text{strat}} \mid \hat{m} = 1) \\ &\geq 1 - \alpha - \mathbb{P}(M = 0 \mid \hat{m} = 1) = 1 - \alpha - \rho_{\hat{m}},\end{aligned}$$

which is Equation (12). Since coverage implies $M = 1$, the upper bound $\mathbb{P}(\text{covered} \mid \hat{m} = 1) \leq 1 - \rho_{\hat{m}}$ follows immediately.

For the marginal lower bound Equation (13), decompose the coverage event by the $\hat{m}$ taxonomy:

$$\begin{aligned}\mathbb{P}\big(y^*_{N+1} \in \mathcal{C}^{\text{strat}}_{\lambda_0}(x_{N+1})\big) &= \mathbb{P}\big(y^*_{N+1} \in \mathcal{C}^{\text{strat}}_{\lambda_0}(x_{N+1}) \,\big|\, \hat{m}(x_{N+1}) = 1\big) \cdot \mathbb{P}\big(\hat{m}(x_{N+1}) = 1\big) \\ &\quad + \mathbb{P}\big(y^*_{N+1} \in \mathcal{C}^{\text{strat}}_{\lambda_0}(x_{N+1}),\, \hat{m}(x_{N+1}) = 0\big) \\ &\geq (1 - \alpha - \rho_{\hat{m}})\, \mathbb{P}\big(\hat{m}(x_{N+1}) = 1\big).\end{aligned}$$

This is equivalent to Equation (13) because $\rho_{\hat{m}}\mathbb{P}(\hat{m} = 1) = \mathbb{P}(M = 0, \hat{m} = 1)$. The corollary below makes the upper-bound implication explicit. □

Corollary 1 (Ceiling on hard cells)**.** *On the predicted-matchable stratum, coverage cannot exceed* $1 - \rho_{\hat{m}}$*. Thus low classifier precision lowers attainable coverage within the selected stratum, regardless of the threshold. A low population matchability rate* π *instead caps marginal coverage and may limit the fraction of queries a precise classifier can select.*

Sketch. Direct from the fact that every covered item must be matchable. □

C Algorithm

Algorithm 1 summarizes the full CAELUM pipeline used for the main experiments, including the optional matchability-stratified phase.

Algorithm 1 CAELUM: Calibrated Aleatoric-Epistemic Language Uncertainty Modeling

Require: LLM $\mathcal{M}$; calibration data $\mathcal{D}_{\text{cal}}$; n clarifications; m answers per clarification; nonempty finite grid $\Lambda \subset [0, \infty)$; error rate $\alpha \in (0, 1)$.

Phase 1: Two-Stage Sampling and Decomposition
for each $(x_i, y_i^*) \in \mathcal{D}_{\text{cal}}$ **do**
  Generate clarifications $W_1, \ldots, W_n$ from $\mathcal{M}$ given x_i
  for each clarification W_j **do**
    Sample m answers $y_{j1}, \ldots, y_{jm} \sim \mathcal{M}(\cdot \mid W_j)$
  end for
  Cluster all nm answers semantically $\rightarrow$ sorted proportions $\{p_k\}$, representatives $\{\hat{y}_k\}$
  Compute $U_{\text{total}}(x_i)$ as pooled VNE and $U_{\text{epi}}(x_i)$ as mean per-clarification VNE (Appendix F); set $U_{\text{ale}}(x_i) = \max(U_{\text{total}}(x_i) - U_{\text{epi}}(x_i), 0)$
  Apply the fixed reference matcher to record M_i and, if matched, k_i^*
  Set $s_i \leftarrow 1$ if $M_i = 0$, otherwise $s_i \leftarrow \sum_{k<k_i^*} p_k(x_i)$, adding $U_i p_{k_i^*}(x_i)$ for randomized scoring with independent $U_i \sim \text{Unif}(0, 1)$ fixed across λ ▷ Equation (5)
end for

Phase 2: Split Calibration
Split $\mathcal{D}_{\text{cal}}$ into nonempty $\mathcal{V}_1$ and $\mathcal{V}_2$, plus a disjoint $\mathcal{V}_0$ if Phase 3 is used
for each $\lambda \in \Lambda$ **do** ▷ λ selection on $\mathcal{V}_1$
  $\tilde{s}_{\lambda,i} \leftarrow s_i \,/\, w_\lambda(x_i)$ for all $i \in \mathcal{V}_1$ ▷ Score normalization in Equation (8)
  $\hat{\tau}_\lambda \leftarrow Q_{\lceil (1-\alpha)(|\mathcal{V}_1|+1) \rceil / |\mathcal{V}_1|}(\{\tilde{s}_{\lambda,i}\}_{i \in \mathcal{V}_1})$
  $\text{APSS}(\lambda) \leftarrow \frac{1}{|\mathcal{V}_1|} \sum_{i \in \mathcal{V}_1} |\mathcal{C}_\lambda(x_i)|$ using $\hat{\tau}_\lambda$ and Equation (7)
end for
$\lambda^\star \leftarrow \arg\min_{\lambda \in \Lambda} \text{APSS}(\lambda)$, choosing the smallest minimizing weight
$\tilde{s}_{\lambda^\star,i} \leftarrow s_i \,/\, w_{\lambda^\star}(x_i)$ for all $i \in \mathcal{V}_2$ ▷ Score normalization
$\hat{\tau} \leftarrow Q_{\lceil (1-\alpha)(|\mathcal{V}_2|+1) \rceil / |\mathcal{V}_2|}(\{\tilde{s}_{\lambda^\star,i}\}_{i \in \mathcal{V}_2})$ ▷ Full-population empirical threshold
  For Proposition 1, replace $\mathcal{V}_2$ by $\mathcal{V}_2^M$ in this threshold

Phase 3 (optional, for the matchability-stratified variant of Theorem 1):
Train binary classifier $\hat{m} : \mathcal{X} \rightarrow \{0, 1\}$ on $\mathcal{V}_0$ (disjoint from $\mathcal{V}_1, \mathcal{V}_2$, test) using question features available before generation
Restrict $\mathcal{V}_2$ to $\mathcal{S} = \{i \in \mathcal{V}_2 : \hat{m}(x_i) = 1\}$ and recompute the threshold $\hat{\tau}_{\text{strat}}(\lambda^\star)$ via Equation (11) on $\mathcal{S}$

For a new x, repeat Phase 1 sampling, clustering, and uncertainty estimation without a reference, apply Equation (7) with $\hat{\tau}$, or with $\hat{\tau}_{\text{strat}}$ in Phase 3, and report coverage on $\{\hat{m} = 1\}$ separately

Ensure: Prediction set $\mathcal{C}_{\lambda^\star}(x)$ via Equation (7); diagnostic $\lambda^\star$; per-instance $R(x)$ via Equation (14); with Phase 3, predicted-matchable set $\mathcal{C}_{\lambda^\star}^{\text{strat}}(x)$.

D Dataset descriptions

Factual benchmarks. These benchmarks test factual recall. A dataset label does not determine which component of prompt variability will be largest for a particular generator.

- **TriviaQA** [20]: Factual questions and answers authored by trivia enthusiasts. We use the 300-item curated subset released by Walha et al. [11] for direct comparability.
- **OpenNQ** [21]: Factual questions drawn from real Google search queries (Natural Questions development set). We use the 300-item curated subset released by Walha et al. [11].
- **SimpleQA** [22]: Short-form factual questions designed as a challenging factuality benchmark, subsampled to 500 items. The benchmark was adversarially collected against GPT-4 responses to stress-test short-form factuality.
- **PopQA** [23]: Entity-centric factual questions annotated with subject popularity, subsampled to 500 items.

Ambiguous benchmarks. These benchmarks include questions or instructions with multiple valid interpretations, together with unambiguous items. Their annotations allow a separate check of the relationship between prompt variability and ambiguity.

- **AmbigQA** [24]: Open-domain questions from NQ, including questions that admit different interpretations. We use the 200-item curated subset of Walha et al. [11].
- **AmbigInst** [14]: Synthetic instruction-following tasks with ambiguous and unambiguous instructions. We use all 304 samples in the release of Walha et al. [11].

E Implementation details

Notation summary. For reference, the symbols used in the body and proofs are:

X, Y, C	Query, pooled answer-cluster label, and clarification index in the Shannon identity, with C uniform over clarifications and $\mathbb{P}(Y = k \mid C = i, X = x) = p_k^{(i)}(x)$
$K = K(x)$, $p_k^{(i)}(x)$	Number of clusters and fraction of clarification i's answers assigned to pooled cluster k
$\log$, $0 \log 0$	Natural logarithm, with the entropy convention $0 \log 0 = 0$
$U_{\text{total}}, U_{\text{epi}}, U_{\text{ale}}$	Pooled, epistemic, and aleatoric VNE signals, Equation (4) and Appendix F
$w_\lambda(x) = 1 + U_{\text{ale}} + \lambda U_{\text{epi}}$	Per-instance weight in the conformal score, Equation (6)
$\lambda^\star$	Per-cell APSS-minimizing mixing coefficient, Equation (9)
$\tilde{s}_\lambda = s / w_\lambda$	Weight-normalized APS score, Equation (8)
$\hat{\tau}$	Empirical full-$\mathcal{V}_2$ threshold in Equation (10); $\hat{\tau}_M$ is the support-restricted threshold in Proposition 1; $\hat{\tau}_{\text{strat}}$ on $\mathcal{S} \subseteq \mathcal{V}_2$
$M(x, y^*)$	True matchability (reference appears in some sampled cluster)
$\hat{m}(x) \in \{0, 1\}$	Pre-generation logistic classifier predicting M
$\pi = \mathbb{P}(M = 1)$	Population matchability rate
$\rho_{\hat{m}} = \mathbb{P}(M = 0 \mid \hat{m} = 1)$	False-matchability rate inside the predicted-matchable stratum
$\mathcal{S} = \{i \in \mathcal{V}_2 : \hat{m}(x_i) = 1\}$	Calibration sub-sample for the stratified threshold
$\mathcal{V}_0, \mathcal{V}_1, \mathcal{V}_2$	Calibration splits: $\hat{m}$ training, $\lambda^\star$ selection, $\hat{\tau}$ calibration
$R(x) = \log(\lambda^\star U_{\text{epi}} + \epsilon) - \log(U_{\text{ale}} + \epsilon)$	Diagnostic ratio for each instance, Equation (14)

Lambda grid. The grid Λ consists of 10 linearly spaced values in $[0, 0.09]$ concatenated with 4001 log-spaced values in $[10^{-1}, 10^2]$, yielding 4011 candidate values. This provides fine resolution near $\lambda = 0$ (where the transition from aleatoric-only to mixed behavior occurs) and broader coverage at larger scales.

Matchability classifier features. The inputs to $\hat{m}$ include the all-mpnet-base-v2 sentence embedding of the question and other sentence features: character length and token length, number of capitalized and numeric tokens, a year-pattern feature, and one-hot encoding of the first word of the question using a fixed vocabulary of 18 words (*what, who, when, where, why, how, which, is, are, was, were, did, do, does, can, name, list, describe*) and an *other* category.

Data splits. The main table uses a fixed 40%/30%/30% split ($\mathcal{V}_1$/$\mathcal{V}_2$/Test) with seed 42 and deterministic APS, and Table 6 repeats that split for seeds 42–45 with seeded randomized APS. The coefficient SDs in Tables 1 and 7 instead use deterministic APS on those four splits. For 200 samples per dataset, the seed 42 main split yields 80 $\mathcal{V}_1$ / 60 $\mathcal{V}_2$ / 60 Test. The pre-generation matchability experiment uses a separate 30% $\mathcal{V}_0$ / 30% $\mathcal{V}_2$ / 40% Test split, and the integrated $\lambda^\star$ matchability ablation uses four 25% splits.

Cluster alignment. After clustering, the cluster proportions and representative answers are sorted in descending order of proportion. The correct cluster index must point into this sorted array, otherwise APS scoring is corrupted.

Correctness matching. The reference answer is matched under the following tiers, applied in order to cluster representatives and then to raw cluster members for the lexical/overlap tiers: (1) exact substring (in either direction, after lower-casing and punctuation stripping); (2) normalized-substring match after Unicode and diacritic folding; (3) depluralized whole-word match for short answers; (4) single-token containment in short candidates (≤ 6 tokens) to handle entity answers; (5) ROUGE-L > 0.15; (6) Jaccard > 0.25. If no lexical/overlap tier matches, tier (7) uses embedding cosine > 0.4 (all-mpnet-base-v2) on cluster representatives. If no cluster matches, the sample is unmatched and receives the maximal nonconformity score ($s = 1.0$). These tiers are heuristic and can accept false positives as well as miss valid paraphrases. The embedding tier is permissive. A higher measured match rate is not by itself evidence of a better matcher. Appendix G stress-tests the resulting match rates across encoders.

F von Neumann entropy decomposition

We use an RBF kernel as in Walha et al. [11], fixing $\gamma = 1$ for all datasets; their AmbigInst experiments instead use $\gamma = 100$. L2-normalized sentence embeddings $\phi(y)$ define $k(y, y') = \exp(-\gamma\|\phi(y) - \phi(y')\|^2)$ with $\gamma = 1$. For N answers, let $K_{ij} = k(y_i, y_j)/N$, so $\operatorname{Tr} K = 1$. The von Neumann entropy is $H_{\mathrm{VN}}(K) = -\sum_r \mu_r \log \mu_r$, with non-negative eigenvalues μ_r and natural logarithms. We use unscaled entropy for both the pooled and per-clarification kernels.

The normalized Gram matrix has the same nonzero spectrum as the empirical density operator in the common reproducing-kernel Hilbert space. For the shared kernel feature map ψ, where $\langle \psi(y), \psi(y') \rangle = k(y, y')$, let $\rho_i = m^{-1} \sum_j \psi(y_{ij}) \otimes \psi(y_{ij})$, where $u \otimes u$ is the operator $v \mapsto u\langle u, v\rangle$. With equal answer counts, $\rho_{\mathrm{pool}} = n^{-1} \sum_i \rho_i$, so entropy concavity gives $H_{\mathrm{VN}}(\rho_{\mathrm{pool}}) \geq n^{-1} \sum_i H_{\mathrm{VN}}(\rho_i)$. Computing each Gram block separately does not alter this identity. The encoder, embeddings, kernel, and bandwidth are shared. The implementation clips negative numerical eigenvalues and floors the residual at zero. Unequal answer counts require size-weighted rather than uniform averaging for this identity. Finite-sample entropy estimates are biased, and this decomposition measures response variability under the chosen clarification procedure rather than ground-truth ambiguity.

G Embedding-sensitivity ablation

We recomputed clustering, VNE, and reference matching for three cells (TriviaQA / AmbigQA / PopQA on Qwen3-8B) with four sentence encoders, holding generations, splits, and numerical thresholds fixed. The default protocol uses all-mpnet-base-v2 with similarity threshold $\delta = 0.7$.

On the three audited cells, the default all-mpnet-base-v2 encoder ($\delta = 0.7$) and an alternative MiniLM-L12 encoder differ in match rate by less than three percentage points and in AUROC for U_{epi} by less than 0.02. MiniLM-L12 produces larger sets on all three cells. Larger encoders (bge-large-en-v1.5, e5-large-v2) result in excessive merging at $\delta = 0.7$ and, for e5-large-v2, degenerate clustering. We follow the encoder used in Walha et al. [11] for direct comparability.

Table 4: Embedding-sensitivity ablation. We vary only the sentence encoder, which is used for clustering, VNE, and reference matching. mpnet (our default) and MiniLM-L12 give similar match rates and U_{epi} AUROC, but MiniLM-L12 produces larger sets on all three cells. bge-large becomes too permissive at $\delta = 0.7$ and degrades U_{epi} AUROC on two of three cells. e5-large yields 100% match (degenerate clustering) at the same threshold. Results are unranked because the encoders also change the support labels. The coefficients use deterministic APS and the split with seed 42.

Encoder	Match%	$\lambda^\star$	Coverage	APSS	U_{epi} AUROC
TriviaQA					
all-mpnet-base-v2	78.0	47.4	0.811	1.82	0.709
all-MiniLM-L12-v2	75.3	9.58	0.767	2.02	0.724
bge-large-en-v1.5	93.7	17.1	0.933	1.28	0.519
e5-large-v2	100.0	0.00	1.000	1.00	n/a
AmbigQA					
all-mpnet-base-v2	67.0	7.54	0.667	1.30	0.632
all-MiniLM-L12-v2	66.5	78.0	0.617	1.55	0.636
bge-large-en-v1.5	88.0	24.8	0.883	1.00	0.736
e5-large-v2	100.0	0.00	1.000	1.00	n/a
PopQA					
all-mpnet-base-v2	53.6	3.15	0.533	1.74	0.780
all-MiniLM-L12-v2	52.2	43.9	0.527	2.13	0.778
bge-large-en-v1.5	76.2	56.4	0.727	1.65	0.772
e5-large-v2	100.0	0.00	1.000	1.00	n/a

H Clarifier-quality sensitivity

The decomposition reflects the LLM's clarification diversity, so clarifier quality is part of the measured pipeline. All main experiments use the answer LLM itself as the clarifier, which keeps the support-ceiling analysis attributable to a single sampled generator. External clarifiers are not used in the reported experiments. An external clarifier may change matchability, and its benefit would require a separate evaluation.

I Evaluation metrics

- **Marginal coverage.** Fraction of test queries where $y^* \in \mathcal{C}(x)$. For sampled semantic sets this rate is at most the test support rate. The support-conditional and learned stratum guarantees are stated separately (we use $\alpha = 0.1$).
- **APSS (Average Prediction Set Size).** Mean number of semantic clusters in prediction sets across test queries indexed by T: $\text{APSS} = \frac{1}{|T|} \sum_{i \in T} |\mathcal{C}(x_i)|$. Lower is better at equal coverage, as smaller sets are more informative.
- **Support-failure / hallucination AUROC.** Area under the ROC curve using U_{epi} (or U_{total}, U_{ale}) as a predictor of support failure, where a sample is labeled positive if the reference answer does not match any generated cluster under the fixed matching rule. This often indicates hallucination, but the measured event is absence from generated support.
- **Ambiguity AUROC.** Area under the ROC curve using U_{ale} as a predictor of the binary ambiguity label, available for AmbigQA and AmbigInst in the main benchmark.
- **Diagnostic $\lambda^\star$.** The calibrated epistemic weight selected on $\mathcal{V}_1$ via Equation (9). The value describes the fitted score weight. A regime interpretation requires a nonflat objective and split stability.

J Full comparison of semantic prediction sets

Table 5 expands the main comparison with paired coverage and APSS for all six methods. It uses the same generated answers and deterministic seed-42 split. Set size alone cannot establish efficiency when the methods achieve different coverage. Table 8 provides a separate comparison at matched coverage.

Table 5: Coverage and average set size for all six set rules at nominal $\alpha = 0.1$. Every pair of columns reports both quantities for the same method and test split. SC proxy is the simplified self-consistency baseline, not original ConU. As in Table 1, **bold** and underline mark the smallest and second-smallest displayed sizes among methods tied for the highest coverage in each row.

	CAELUM		Vanilla		Equal		Epi. only		SC proxy		SE-CP	
Dataset	Cov.	APSS	Cov.	APSS	Cov.	APSS	Cov.	APSS	Cov.	APSS	Cov.	APSS
Qwen3-8B												
TriviaQA	0.811	1.82	0.811	1.84	0.811	1.84	0.811	**1.81**	0.778	1.02	0.811	1.84
AmbigQA	0.667	**1.30**	0.667	**1.30**	0.667	**1.30**	0.667	**1.30**	0.650	1.00	0.667	**1.30**
OpenNQ	0.700	1.20	0.700	1.20	0.700	1.20	0.700	1.20	0.700	**1.00**	0.700	1.20
AmbigInst	0.935	**1.00**	0.935	**1.00**	0.935	**1.00**	0.935	**1.00**	0.935	1.05	0.935	**1.00**
SimpleQA	0.367	1.77	0.367	1.77	0.367	1.77	0.367	**1.75**	0.327	1.02	0.367	1.77
PopQA	0.533	**1.74**	0.533	1.80	0.533	1.79	0.533	1.75	0.520	1.02	0.533	1.79
Ministral-3-8B-Instruct-2512												
TriviaQA	0.811	**2.27**	0.811	2.34	0.811	2.37	0.811	2.28	0.767	1.07	0.811	2.37
AmbigQA	0.733	**3.90**	0.717	4.08	0.717	4.05	0.733	3.92	0.717	1.90	0.717	4.05
OpenNQ	0.733	**1.61**	0.733	1.63	0.733	1.63	0.722	1.60	0.711	1.00	0.733	1.63
AmbigInst	0.859	**2.32**	0.859	**2.32**	0.859	2.75	0.859	2.99	0.859	2.63	0.859	2.75
SimpleQA	0.340	2.31	0.340	2.30	0.340	2.31	0.340	**2.29**	0.287	1.00	0.340	2.31
PopQA	0.487	**3.57**	0.487	3.62	0.487	**3.57**	0.487	**3.57**	0.447	1.12	0.487	**3.57**
Gemma-4-E4B-it												
TriviaQA	0.678	1.74	0.678	1.77	0.678	1.77	0.678	**1.73**	0.656	1.01	0.678	1.77
AmbigQA	0.633	**2.17**	0.633	2.18	0.633	**2.17**	0.633	**2.17**	0.583	1.05	0.633	**2.17**
OpenNQ	0.556	**1.51**	0.556	**1.51**	0.556	**1.51**	0.556	**1.51**	0.544	1.00	0.556	**1.51**
AmbigInst	0.935	1.96	0.924	1.97	0.935	1.99	0.935	**1.89**	0.870	1.42	0.935	1.99
SimpleQA	0.207	**1.66**	0.207	**1.66**	0.207	1.67	0.207	1.67	0.180	1.00	0.207	1.67
PopQA	0.373	1.94	0.373	**1.93**	0.373	1.95	0.373	**1.93**	0.347	1.02	0.373	1.95

K Weight selection on a separate validation subset

The procedure for selecting $\lambda^\star$ in Equation (9) computes the conformal threshold $\hat{\tau}_\lambda$ on $\mathcal{V}_1$ and measures APSS *on the same* $\mathcal{V}_1$ for each candidate λ. With a 4011-point grid on a small $\mathcal{V}_1$ (≈ 80–200 items per cell), this is susceptible to selection bias.

A held-out variant splits $\mathcal{V}_1$ into $\mathcal{V}_1^{\text{train}}$ (60%) and $\mathcal{V}_1^{\text{eval}}$ (40%) and computes $\hat{\tau}_\lambda$ on $\mathcal{V}_1^{\text{train}}$ but evaluates the APSS for the argmin on $\mathcal{V}_1^{\text{eval}}$. This separates threshold estimation from weight evaluation, whereas the original procedure uses all of $\mathcal{V}_1$ for both.

The main 18-cell comparison uses selection on all of $\mathcal{V}_1$. Across seeds, CAELUM and Vanilla have nearly identical cell-mean APSS, and their differences lie within the reported split variation (Table 6).

Standardized weight and set-score selection. The judged rerun in Appendix S.5 and the semantic sets in Appendix S.7 use a different variant. Let μ_{ale} and μ_{epi} be the means of U_{ale} and U_{epi} over all of $\mathcal{V}_1$, replacing a mean below 10^{-8} by 1. This variant uses

$$w_\lambda^{\text{std}}(x) = 1 + \frac{U_{\text{ale}}(x)}{\mu_{\text{ale}}} + \lambda \frac{U_{\text{epi}}(x)}{\mu_{\text{epi}}}.$$

It uses randomized APS and the 60/40 split of $\mathcal{V}_1$ described above. For each weight, the threshold is fitted on $\mathcal{V}_1^{\text{train}}$, and the selection objective on $\mathcal{V}_1^{\text{eval}}$ is mean set size plus $2/\alpha$ times the fraction of references absent from the prediction set. The first minimizing grid weight is selected. The judged rerun uses all items in each subset. The abstention experiment instead restricts both this threshold calculation and the selection objective to matched items, and fits the final semantic threshold on matched items in the first half of $\mathcal{V}_2$. The uncertainty means still use all of $\mathcal{V}_1$. These weights and their selection objective differ from Table 1, so their fitted coefficients are not directly comparable.

Table 6: Per-cell APSS (mean $\pm$ SD across seeds 42–45) and $\lambda^\star$ at nominal $\alpha = 0.1$. **Bold** and underline mark the smallest and second-smallest displayed mean APSS, including ties. APSS values and their ranks are descriptive because coverage is not matched across methods. CAELUM and Vanilla agree within their per-cell split SD on all 18 cells. The largest CAELUM APSS SD is 0.37 and the largest Vanilla APSS SD is 0.38. SE-CP and TrueAPS have larger SD on a few high-variance cells. $\lambda^\star$ varies widely across seeds on cells with a nearly flat objective (e.g. Qwen3-TriviaQA 14.6 ± 23.8), even though the resulting APSS is much more stable. Flat objectives do not identify a regime from $\lambda^\star$.

		Average set size (mean ± split SD)				
Dataset	Match%	CAELUM	Vanilla	SE-CP	TrueAPS	$\lambda^\star$
Qwen3-8B						
TriviaQA	78.0	1.82±0.13	1.82±0.12	**1.81**±0.12	1.84±0.12	14.6±23.8
AmbigQA	67.0	**1.30**±0.08	**1.30**±0.08	**1.30**±0.08	**1.30**±0.08	13.4±8.7
OpenNQ	63.7	**1.19**±0.03	**1.19**±0.03	**1.19**±0.03	**1.19**±0.03	0.0±0.0
AmbigInst	100.0	1.66±0.05	1.67±0.05	**1.00**±0.00	1.59±0.07	0.0±0.0
SimpleQA	35.6	**1.73**±0.09	**1.73**±0.09	**1.73**±0.09	**1.73**±0.09	8.4±14.5
PopQA	53.6	1.82±0.04	1.82±0.04	**1.81**±0.03	1.83±0.04	4.2±4.5
Ministral-3-8B-Instruct-2512						
TriviaQA	84.7	2.84±0.28	2.84±0.29	**2.45**±0.36	2.85±0.28	9.6±13.2
AmbigQA	67.0	3.72±0.37	3.74±0.38	**3.70**±0.40	3.76±0.40	21.1±33.7
OpenNQ	67.0	**1.74**±0.08	**1.74**±0.08	**1.74**±0.08	**1.74**±0.08	17.0±10.0
AmbigInst	91.4	4.55±0.15	4.42±0.34	**2.64**±0.78	3.83±0.74	1.2±1.6
SimpleQA	42.6	**2.45**±0.15	**2.45**±0.15	**2.45**±0.15	2.46±0.15	0.2±0.1
PopQA	51.6	3.36±0.36	3.36±0.36	**3.33**±0.36	3.39±0.35	0.4±0.5
Gemma-4-E4B-it						
TriviaQA	68.7	**1.75**±0.02	**1.75**±0.03	**1.75**±0.03	**1.75**±0.02	4.7±6.2
AmbigQA	58.0	2.18±0.04	2.17±0.04	**2.16**±0.04	2.18±0.04	6.4±7.4
OpenNQ	51.7	**1.56**±0.06	**1.56**±0.06	**1.56**±0.06	**1.56**±0.06	0.0±0.0
AmbigInst	96.1	2.48±0.20	2.50±0.19	**1.90**±0.09	2.25±0.16	24.6±22.9
SimpleQA	21.2	**1.74**±0.05	**1.74**±0.05	1.75±0.05	1.75±0.05	0.1±0.1
PopQA	38.0	**1.92**±0.06	**1.92**±0.06	1.93±0.06	1.93±0.06	3.8±6.0

L Selected weight coefficients

Table 7 reports the coefficients from Table 1 using deterministic APS with seed 42. Their magnitudes alone do not establish stability or identify an uncertainty regime.

Table 7: Selected $\lambda^\star$ at seed 42 under deterministic APS, rounded to two decimals. Parentheses give the population SD of $\lambda^\star$ over seeds 42–45 under the same deterministic APS protocol.

Dataset	**Qwen3-8B**	**Ministral-3-8B Instruct-2512**	**Gemma-4-E4B-it**
TriviaQA	47.42 (14.5)	26.96 (25.5)	45.74 (13.6)
AmbigQA	7.54 (33.9)	18.41 (24.5)	9.32 (5.2)
OpenNQ	0.00 (0.0)	6.19 (14.8)	0.00 (15.0)
AmbigInst	0.00 (0.0)	0.00 (25.5)	6.17 (18.9)
SimpleQA	0.00 (12.2)	0.32 (20.9)	0.28 (0.1)
PopQA	3.15 (38.7)	1.43 (22.5)	8.51 (7.9)

M Matched-coverage APSS (NCAPSS)

Smaller sets in Table 1 can reflect lower coverage. To compare sizes at matched coverage, *normalized-coverage APSS* (NCAPSS) sets $q^\star = \min(1-\alpha, \text{validation match rate})$ for each cell, sweeps each method over $\alpha \in [0.05, 0.4]$, and interpolates APSS on its test coverage curve at $q^\star$; if every swept coverage exceeds $q^\star$, it reports the smallest swept APSS. A dagger marks methods whose sweep cannot reach the target. This use of test coverage makes NCAPSS a post-hoc diagnostic, not a deployable selection rule. Like the main table, the sweep uses deterministic APS.

Table 8: APSS at matched coverage $q^\star$ (smallest swept APSS when every swept coverage exceeds $q^\star$). **Bold** and underline mark the smallest and second-smallest displayed sizes among methods that reach the target. Daggers mark unattainable targets and are excluded from the ranking. Vanilla beats CAELUM on $2/11$ matched cells. The SC proxy cannot reach the target on $9/18$.

		Average set size at matched coverage ↓					
Dataset	$q^\star$	CAELUM	Vanilla	Equal	Epi.	SC proxy	SE-CP
Qwen3-8B							
TriviaQA	0.77	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
AmbigQA	0.67	1.20	**1.07**	1.10	1.20	**1.07**	1.10
OpenNQ	0.61	1.06	1.17	1.14	1.06	**1.00**	1.14
AmbigInst	0.90	**1.00**	**1.00**	**1.00**	**1.00**	1.01	**1.00**
SimpleQA	0.35	**1.68**	**1.68**	1.73	**1.68**	1.00†	1.73
PopQA	0.54	1.64	1.66	**1.56**	1.67	1.00†	**1.56**
Ministral-3-8B-Instruct-2512							
TriviaQA	0.86	1.92†	1.92†	2.37†	2.28†	1.00†	2.37†
AmbigQA	0.64	**1.07**	1.08	1.08	**1.07**	1.38	1.08
OpenNQ	0.64	1.31	1.31	1.22	1.13	**1.00**	1.22
AmbigInst	0.90	3.92†	2.32†	3.25†	3.86†	2.32†	3.25†
SimpleQA	0.46	2.27†	2.29†	2.27†	2.27†	1.00†	2.27†
PopQA	0.53	3.17†	3.31†	3.15†	3.23†	1.08†	3.15†
Gemma-4-E4B-it							
TriviaQA	0.69	1.71†	1.61†	1.66†	1.71†	1.00†	1.66†
AmbigQA	0.56	1.57	1.92	1.90	1.55	**1.00**	1.90
OpenNQ	0.50	1.26	1.39	1.34	1.19	**1.00**	1.34
AmbigInst	0.90	1.65	1.57	**1.51**	1.66	1.66	**1.51**
SimpleQA	0.21	1.65†	1.63†	1.65†	1.65†	1.00†	1.65†
PopQA	0.38	1.81†	1.86†	1.86†	1.80†	1.00†	1.86†

N Support failure AUROC

Table 9: Support failure AUROC at seed 42. Each Δ subtracts the AUROC of U_{total} on the [0,1] scale. **Bold** and underline mark the best and second-best AUROC.

		Support-failure AUROC ↑				Difference vs. U_{total}	
Dataset	Match%	U_{total}	U_{epi}	U_{ale}	$R(x)$	Δ_{epi}	Δ_R
Qwen3-8B							
TriviaQA	78.0	0.658	0.709	0.571	**0.730**	+0.050	+0.072
AmbigQA	67.0	0.574	0.632	0.449	**0.677**	+0.059	+0.104
OpenNQ	63.7	0.629	**0.650**	0.581	0.419	+0.021	−0.210
SimpleQA	35.6	**0.588**	0.583	0.558	0.442	−0.005	−0.146
PopQA	53.6	0.761	**0.780**	0.641	0.764	+0.019	+0.003
Ministral-3-8B-Instruct-2512							
TriviaQA	84.7	0.765	**0.823**	0.632	0.803	+0.058	+0.038
AmbigQA	67.0	**0.666**	0.665	0.632	0.453	−0.001	−0.213
OpenNQ	67.0	0.751	**0.802**	0.584	0.763	+0.051	+0.011
AmbigInst	91.4	**0.735**	0.724	0.682	0.318	−0.011	−0.418
SimpleQA	42.6	0.528	0.565	0.470	**0.598**	+0.037	+0.069
PopQA	51.6	0.707	**0.729**	0.630	0.721	+0.022	+0.014
Gemma-4-E4B-it							
TriviaQA	68.7	0.727	**0.773**	0.606	0.733	+0.046	+0.006
AmbigQA	58.0	0.600	**0.605**	0.541	0.529	+0.005	−0.072
OpenNQ	51.7	**0.672**	0.665	0.606	0.395	−0.007	−0.278
AmbigInst	96.1	**0.849**	0.766	0.787	0.574	−0.083	−0.275
SimpleQA	21.2	0.389	0.399	0.409	**0.457**	+0.010	+0.068
PopQA	38.0	**0.785**	0.749	0.761	0.607	−0.036	−0.177

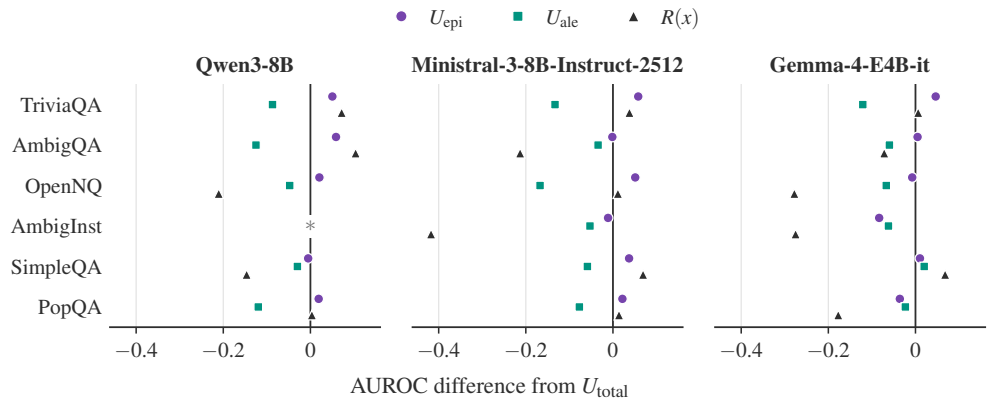


Figure 3: Support failure AUROC differences relative to U_{total} at seed 42. Positive values indicate better ranking. *Qwen3-8B/AmbigInst has no support failures and therefore no defined AUROC.

Paired AUROC comparisons. An inverse-variance meta-analysis using DeLong estimates across the 17 cells gives a fixed-effect $\Delta_{\text{EPI}} = 0.019$ (95% CI $[0.007, 0.032]$, $p = 0.003$) and a random-effect estimate of 0.019 ($p = 0.004$), with $I^2 = 0.04$ indicating negligible between-cell heterogeneity. The DeLong test accounts for per-cell AUROC variance, whereas a one-sided sign test uses only the $11/17$ wins ($p \approx 0.17$). Both summaries treat cells as independent, although the three generators answer the same questions.

Variation across splits. The multi-seed evaluation (seeds 42–45) gives $U_{\text{epi}} > U_{\text{total}}$ on $15/17$ cells with mean $\bar{\Delta}_{\text{EPI}} = +0.012$. U_{epi} seed std is at most 0.094 (Gemma-AmbigInst) and is ≤ 0.052 on the other cells.

The ratio score. $R(x)$ obtains gains over U_{total} on TriviaQA (Qwen3 $+0.072$, Ministral $+0.038$, Gemma $+0.006$) and on Qwen3-AmbigQA ($+0.104$), but underperforms U_{total} on $8/17$ cells, including all four with $\lambda^\star = 0$ (mean $\Delta_R = -0.083$ across the 17 cells). On a separate selective-answering routing AUC evaluation that uses randomized APS for weight selection, $R(x)$ loses to U_{epi} on $16/18$ cells (mean $\Delta_{R-U_{\text{epi}}} = -0.046$, two-sided sign-test $p = 0.001$, Wilcoxon $p = 5\times10^{-5}$). Away from the numerical floor, $\log \lambda^\star$ contributes a cell-wide constant to R, while the varying aleatoric denominator can change the ordering. At $\lambda^\star = 0$, the epistemic term disappears.

O Selective answering (full table)

Table 10: Selective-answering experiment with deterministic APS for weight selection. AUC of accuracy as a function of escalation rate, when items are ranked by each signal (higher AUC = better). *Random* is a reference policy. *Oracle* (escalating incorrect items first) is the upper bound. $\Delta_{R(x)}$ is the absolute AUC gain of $R(x)$ over Random (*not* a percentage). **Bold** and underline mark the best and second-best signals in each row among $\{U_{\text{total}}, U_{\text{epi}}, R(x)\}$. We aggregate across the three models per dataset (mean over models).

Dataset (mean over 3 LLMs)	Random	U_{total}	U_{epi}	$R(x)$	Oracle	$\Delta_{R-\text{random}}$
TriviaQA	0.865	0.917	**0.923**	0.915	0.963	+0.049
AmbigQA	0.801	**0.854**	0.852	0.819	0.928	+0.018
OpenNQ	0.834	0.868	**0.871**	0.827	0.936	−0.007
AmbigInst	0.924	**0.963**	0.956	0.910	0.986	−0.014
SimpleQA	0.626	0.645	**0.647**	0.631	0.728	+0.005
PopQA	0.738	**0.787**	0.785	0.766	0.838	+0.028

P Matched-stratum coverage with $\lambda^\star$-weighted scores

The empirical stratified-coverage results in Section 3 use the Vanilla-CP score ($\lambda = 0$) on the matchable stratum, to isolate the effect of stratifying by matchability. Replacing the score with the $\lambda^\star$-weighted form $\tilde{s}_{\lambda^\star,i} = s_i / w_{\lambda^\star}(x_i)$ also satisfies the bound because Theorem 1 holds for any λ_0 fixed or selected using $\mathcal{V}_1$, and $\lambda^\star$ selected on $\mathcal{V}_1$ is by construction independent of $\mathcal{V}_2$ and the test point.

Integrated pipeline. We (i) split each cell 25/25/25/25 into $\mathcal{V}_0$ (matchability classifier on pre-generation features), $\mathcal{V}_1$ ($\lambda^\star$ argmin), $\mathcal{V}_2$ (matched-stratum $\hat{\tau}$), and Test; (ii) run the standard two-parameter CAELUM calibration on $\mathcal{V}_1, \mathcal{V}_2$ to obtain $\lambda^\star$; (iii) compute $\hat{\tau}_{\text{strat}}(\lambda^\star)$ via Equation (11) on the predicted-matchable subset of $\mathcal{V}_2$; (iv) construct $\mathcal{C}^{\text{strat}}_{\lambda^\star}(x)$ on items with $\hat{m}(x) = 1$ in test. Because the only change is the per-item weight and $\lambda^\star$ is fixed before $\mathcal{V}_2$, the support-aware bound of Theorem 1 is unchanged.

Empirical instantiation. Across the same 18 cells (seed 42, 4-way split), the integrated variant passes the slack-adjusted empirical diagnostic on $10/18$ cells and gives matched-stratum coverage 0.72 ± 0.21, including three oracle-fallback rows. On the 15 learned-classifier rows, matched-stratum coverage is 0.67 ± 0.19 and classifier $\mathcal{V}_2$ AUC is 0.64 ± 0.07. These $\pm$ values are standard deviations across cells. Selected $\lambda^\star$ values include Qwen3-TriviaQA 29.8, Ministral-AmbigQA 11.9, and Gemma-AmbigQA 37.5, with 0 selected on 12 cells. On the same 18 cells, the body's 3-way variant (pre-gen classifier, $\lambda = 0$ on the predicted-matchable stratum, 4 random splits) gives learned-only matched-stratum coverage 0.71 ± 0.14 and full support-audit coverage 0.82 ± 0.17 when flagged oracle-fallback rows are included. The integrated variant checks the stratified construction with a $\lambda^\star$-weighted score. The main empirical evaluation uses $\lambda = 0$ to isolate matchability stratification. Per-cell numbers are reported in the supplementary release.

Q Per-cell stratified coverage figure

Figure 4 reports per-cell stratified coverage under the pre-generation matchability classifier, with learned and oracle-fallback evaluations shown separately.

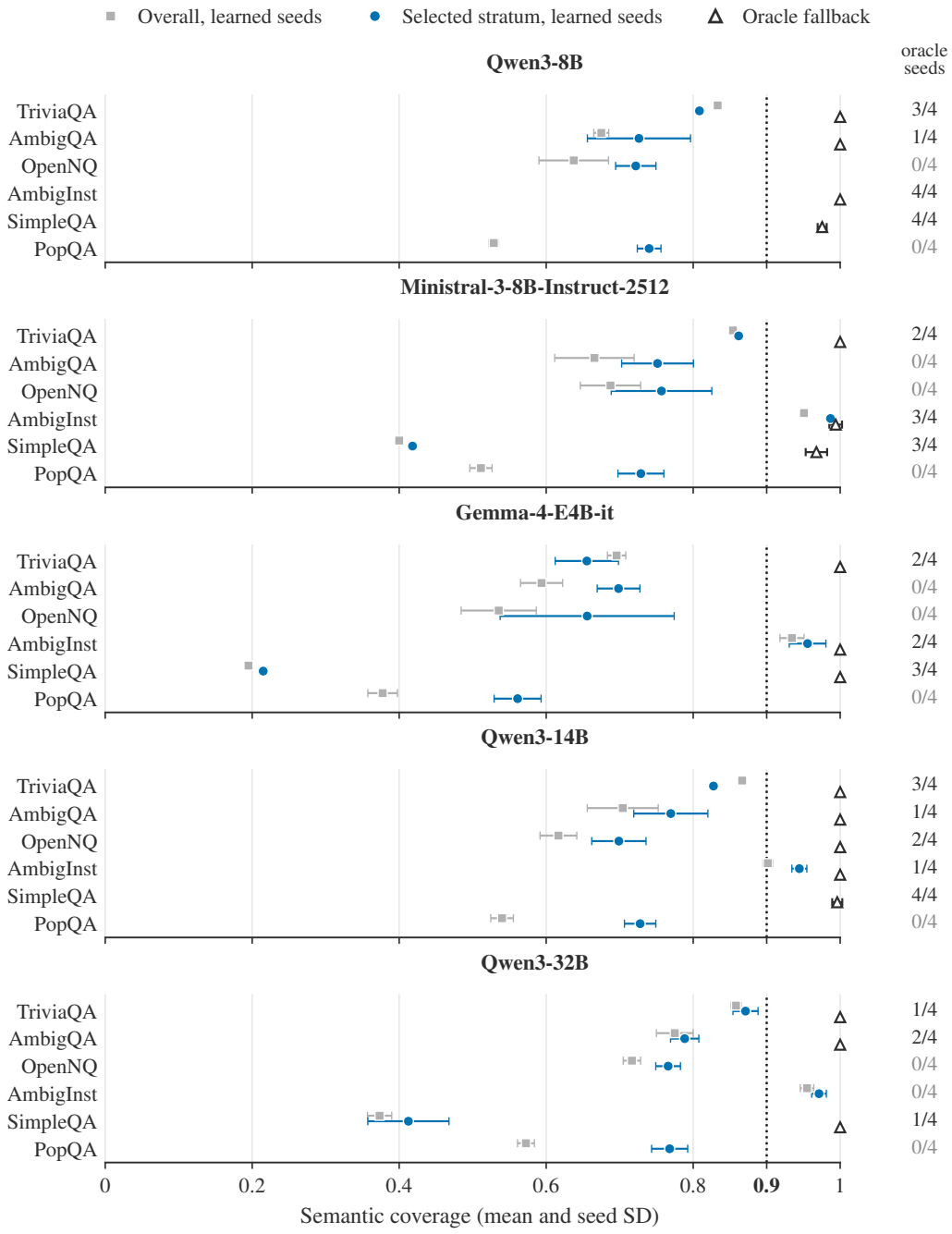


Figure 4: Matchability stratification across four random splits. Blue circles and gray squares compare selected-stratum and overall coverage using only seeds with a learned classifier. Hollow black triangles show oracle-fallback stratum coverage separately. Counts at right give oracle seeds out of four. Error bars are split SD within each subset, so a one-seed subset has zero SD. The dotted line marks nominal 0.9 coverage, not the $\rho_{\hat{m}}$-adjusted theorem bound. Missing learned points mean all four seeds used oracle labels.

R Cross-model matchability transfer (full table with bootstrap CIs)

Table 11 reports the point AUCs, bootstrap CIs, Δ, and label agreement. Pre-generation question features are the all-mpnet-base-v2 sentence embedding, character/token length, capitalized- and numeric-token counts, a year-pattern indicator, and a one-hot question-prefix bucket. We train a logistic regression on the matchability labels from one 4–9B LLM in our family, then evaluate AUC against another LLM's matchability labels on the same 2,104 question records across the six datasets, with all six ordered transfer directions reported below. We align records by dataset and question ID and verify their question text and reference answers. Repeated question texts retain their own matchability labels, including when their references differ.

Table 11: Cross-LLM transfer on $n = 2{,}104$ aligned questions per model pair, reused for fitting and scoring. $\mathrm{AUC}_{\mathrm{src}}$ and $\mathrm{AUC}_{\mathrm{tgt}}$ score the same classifier against source and target LLM labels. Pairwise 95% bootstrap CIs use $B = 1000$ resamples of (label, score) pairs with the fitted model held fixed, so they omit uncertainty from fitting the model. $\Delta = \mathrm{AUC}_{\mathrm{tgt}} - \mathrm{AUC}_{\mathrm{src}}$. Agreement is the fraction of identical source and target labels. The mean row weights the six directions equally and reports point estimates only. These results measure label transfer on the same questions, not unseen-question generalization.

Source LLM → Target LLM	n	$\mathrm{AUC}_{\mathrm{src}}$ ↑ (95% CI)	$\mathrm{AUC}_{\mathrm{tgt}}$ ↑ (95% CI)	Δ	Agreement
Qwen3 → Ministral	2104	0.793 [0.772, 0.813]	0.771 [0.750, 0.791]	−0.022	0.831
Qwen3 → Gemma	2104	0.793 [0.772, 0.813]	0.797 [0.778, 0.815]	+0.004	0.817
Ministral → Qwen3	2104	0.788 [0.769, 0.807]	0.771 [0.750, 0.790]	−0.017	0.831
Ministral → Gemma	2104	0.788 [0.769, 0.807]	0.782 [0.763, 0.801]	−0.006	0.796
Gemma → Qwen3	2104	0.821 [0.803, 0.838]	0.782 [0.762, 0.801]	−0.039	0.817
Gemma → Ministral	2104	0.821 [0.803, 0.838]	0.766 [0.745, 0.785]	−0.055	0.796
Mean over 6 pairs	—	0.801	0.778	−0.023	0.815

Across model pairs, roughly four out of five questions receive the same matchability label. For comparison, let p_{src} and p_{tgt} denote the source and target matchability rates. Independent-label agreement $p_{\mathrm{src}}p_{\mathrm{tgt}} + (1 - p_{\mathrm{src}})(1 - p_{\mathrm{tgt}})$ is 0.50–0.53 at these marginal match rates. The agreement exceeds that independent-label baseline. Since the same questions and datasets are reused, shared benchmark structure and question difficulty may both contribute. Figure 5 visualizes the per-pair AUCs.

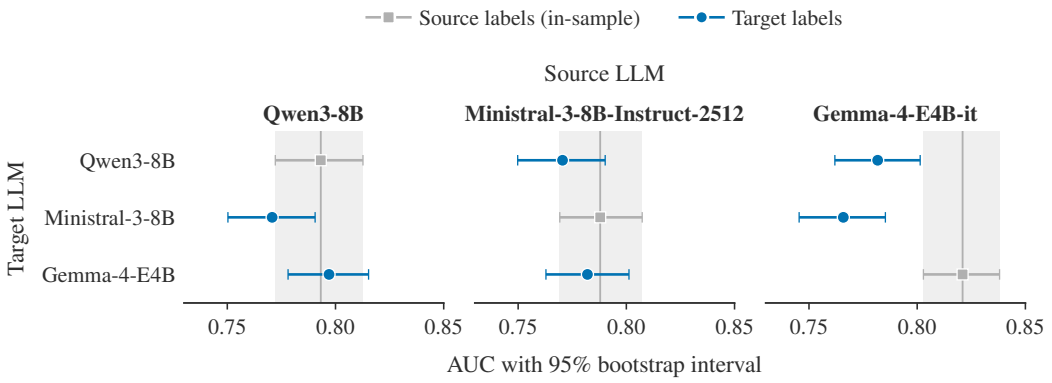


Figure 5: Transfer of matchability labels across generators on the same 2,104 question records. Each panel names the source LLM whose matchability labels train the classifier. Rows name the target LLM whose labels are used for evaluation. Gray squares show AUC against source labels and blue circles show AUC against target labels, with 95% bootstrap intervals. The source result sits in the row of the source LLM, and its interval is shaded across the panel for comparison. Fitting and evaluation reuse the same questions, so this is not an estimate of generalization to unseen questions.

S Reproducibility and further analyses

S.1 Experiment protocols and reproducibility

The main experiments and additional analyses differ in their scores, matching rules, and data splits. A sweep over split seeds reuses generated answers. In the sweep of set sizes, CAELUM's calibrator seed follows the split seed, while Vanilla, Equal-weight, Epistemic-only, and TrueAPS retain the default calibrator seed of 42. SE-CP uses deterministic APS. These sweeps do not measure variation across independent generation runs.

Table 12: Protocols used for the main analyses. Both columns of the comparison at equal sample counts use newly generated answers. That comparison is separate from the other scale and judge analyses. Stratum rows that substitute oracle labels are reported separately as oracle reference results.

Evidence	Questions / labels	Scores	Split / seeds
Main comparison and NCAPSS	18 cells, heuristic	Deterministic APS	40/30/30; 42, with $\lambda^\star$ SD over 42–45
Sets across splits	Same generations	Randomized APS for CAELUM, Vanilla, Equal, Epi., TrueAPS; deterministic SE-CP	40/30/30; 42–45
Support failure ranking	18 cells, heuristic	$U_{\text{total}}, U_{\text{epi}}, U_{\text{ale}}, R(x)$. For $R(x)$, deterministic APS in Table 9, randomized APS across splits	40/30/30; 42 for Table 9, 42–45 across splits
Selective answering	18 cells, heuristic	$R(x)$ with deterministic APS in Table 10; randomized APS for comparison with U_{epi}	40/30/30; 42
Learned stratum	30 cells, heuristic	Randomized APS, $\lambda = 0$	$\mathcal{V}_0/\mathcal{V}_2$/test 30/30/40; 42–45
Integrated stratum	18 cells, heuristic	Randomized APS, fitted λ	Four 25% splits; 42
Transfer across generators	Aligned questions, heuristic	Logistic probability	Fit and score the same 2,104 questions
Scale comparison	30 cells, judged	Standardized variant (Appendix K), randomized APS	40/30/30; 42–45
Classifier audit	30 cells, judged	Question-only logistic probability	$\mathcal{V}_0/\mathcal{V}_2$/test 30/30/40; 42–45
Abstention	29 cells, judged	Standardized variant; matched semantic calibration (Appendix S.7)	Four 25% splits; $\mathcal{V}_2$ halved; 42–45
Scale at equal sample counts	12 cells, heuristic	Support counts over all questions	Generation seed 42; no calibration split

Finite-sample quantiles. The theoretical construction uses the conformal rank $k = \lceil (1-\alpha)(n+1) \rceil$, with threshold $+\infty$ when $k > n$ (including an empty stratum). For $k \leq n$, the implementation used for these results calls NumPy's `quantile` at k/n with `method='higher'`, which selects rank $k+1$ whenever $k < n$. For example, $n = 60$, $\alpha = 0.1$ gives rank 56 rather than 55. This is conservative at ordinary sample sizes. The stratified procedure returns $+\infty$ when fewer than ten calibration items are selected. The reported results use these numerical conventions.

Matching and tie handling. Two further implementation details affect a few reported values. A PopQA reference alias consisting only of a flag emoji normalizes to an empty string and therefore matches any answer, which concerns six of the $10{,}520$ match labels. Set construction compares cumulative mass with $\hat{\tau}\, w_\lambda(x)$ rather than the normalized score, so a cluster can be excluded at an exact floating-point tie. Changing both leaves every coverage value in Table 1 and the stratified summaries unchanged at the reported precision; it changes two APSS values in Table 1 by at most 0.02, one deterministic coverage value at split seed 44 (0.880 to 0.870), a few PopQA-derived values

in the third decimal, and several fitted $\lambda^\star$ values, consistent with the nearly flat selection objective (Appendix S.4). The released code keeps both details, so rerunning it in the documented software environment reproduces the reported values.

Compute. The main experiments use 2,104 questions per generator, with five clarifications and 50 answers per question. We used DCSR clusters at the University of Lausanne for generation and uncertainty computation. Code, results, and details of GPU hardware and compute usage are available at `https://github.com/Unco3892/caelum`.

Assets and licenses. The repository documents the datasets, models, and software used in our experiments, with links to their sources and licenses. CAELUM's original code and software documentation use the MIT License. Third-party assets retain their respective licenses and attribution requirements. Model weights are obtained from their original publishers.

S.2 Classifier operating points and PopQA

The learned-only summaries in Section 3 exclude rows whose $\mathcal{V}_2$ AUC is below 0.55. This filter uses $\mathcal{V}_2$ labels, so Theorem 1 does not apply conditional on passing it.

The main matchability experiment uses a logistic classifier with $C = 0.1$, class-balanced training, and threshold 0.5 on question features. Its oracle substitution after single-class training or calibration AUC below 0.55 uses labels unavailable at deployment. We exclude these rows from learned-only summaries. The learned-only summary is selected using calibration labels. Theorem 1 therefore does not apply conditional on passing that filter. A deployment evaluation must fix the classifier and any retention rule before $\mathcal{V}_2$. The empirical coverage check compares selected-stratum test coverage with $1 - \alpha - 1/\sqrt{N}$, where N is the number of calibration questions in the selected stratum. This tolerance is heuristic, not a confidence bound or a test of Theorem 1.

A separate audit uses the 30 cells with judged labels, seeds 42–45, and a 30/30/40 train/calibration/test split with no oracle substitution. With matchable as the positive class, macro AUROC is 0.620 ± 0.107 over 116 test folds containing both classes. Brier score is 0.216 ± 0.056 and ten-bin ECE is 0.180 ± 0.090 over all 120 folds. The $\pm$ values are fold standard deviations, not confidence intervals.

The pre-generation AUROC of 0.649 in Table 17 uses a different allocation, with 50% of each cell for training and 25% for testing, rather than 30% and 40% here. It is not a second estimate from the same fitted classifiers and test items.

Table 13: Classifier audit with judged labels, macro-averaged over cell–seed folds. Precision and ρ are defined only when the selected subset is nonempty (120 folds at thresholds 0.3 and 0.5, and 104 at 0.7). **Bold** and underline mark the best and second-best displayed precision, recall, and false matchability rate. Higher thresholds reduce the selected population as well as its false matchability penalty.

Threshold	Fraction selected	Precision ↑	Recall ↑	$\rho_{\hat{m}}$ ↓
0.3	0.966	0.671	**0.970**	0.329
0.5	0.578	<u>0.735</u>	<u>0.636</u>	<u>0.265</u>
0.7	0.192	**0.839**	0.207	**0.161**

For PopQA at threshold 0.5, pooling the held-out predictions over 20 overlapping model–seed folds gives selection fraction 0.424, precision 0.715, recall 0.593, and $\rho = 0.285$. These numbers describe a different judged protocol from the stratum results based on heuristic labels. They do not close the PopQA coverage gap. An explanation based on the popularity tail requires an analysis conditioned on popularity. Observations about model size and sampling budget alone cannot identify that cause. Augmented abstention is a separate task, not a repair of the classifier using question features.

Sampling budget, escalation, and held-out model families. In three additional analyses, we ask whether more samples, a larger fallback generator, or a different training set would change our results. In the README of our repository, we describe commands for doing this. First, we take $m \in \{1, 2, 4, 6, 8, 10\}$ previously generated answers per clarification, so between 5 and 50

answers per question. We then re-cluster the answers and apply our heuristic matcher. On the 26 cells where the match rate with 5 answers is below 0.9, the match rate increases by between 3.0 and 13.8 percentage points as we increase m from 1 to 10. But the mean gain per additional answer per clarification falls from 2.2 percentage points between $m = 1$ and $m = 2$ to 0.3 percentage points between $m = 8$ and $m = 10$. The highest match rate on PopQA with 50 answers is 0.572 (Qwen3-32B). The four remaining AmbigInst cells all begin with a match rate of 0.9 or more, and none improve by more than 0.7 percentage points with additional answers.

Second, we evaluate an escalation policy using judged labels. When the base model's learned support-failure score reaches a threshold, the policy uses Qwen3-14B or Qwen3-32B as a fallback. We choose the escalation budget on $\mathcal{V}_1$ and then calibrate abstention using the outcomes of the combined base and fallback policy. On SimpleQA, the selected escalation budget was zero for every base model. So no query was escalated. On PopQA, some queries were escalated. But none of the items that would otherwise abstain became answered and covered with the Qwen3-14B fallback. With Qwen3-32B, at most 1.3% did so on average over split seeds, under both the switching and merged-set policies.

Third, when we use the learned support-failure score from Appendix S.7, with all cells of the target's model family excluded from training (all sizes of Qwen3 for a Qwen3 target), its mean detection AUROC across cells is 0.751. If we exclude only the target cell, it is 0.780. Both means are over 29 cells. They exclude Qwen3-8B–AmbigInst, which has no support failures under judged labels and therefore no defined AUROC.

A tenfold increase in budget raises PopQA matchability by 3.4 to 9.6 percentage points, but not enough to reach our target. Escalation converts almost no abstentions into covered answers. Taken together, these results are consistent with gaps in the generators' knowledge of these questions, but they do not establish why the failures occur.

Exploratory popularity probe. We examine popularity as a possible predictive feature by comparing logistic classifiers using the non-embedding question metadata with and without $\log(1 + s_{\text{pop}})$, where s_{pop} is PopQA's supplied subject-popularity value. The probe uses five PopQA cells with judged labels, seeds 42–45, 30/30/40 splits, threshold 0.5, and the same $C = 0.1$ and class balancing. Scaling is fitted only on the training portion. No oracle labels replace predictions.

Across 20 overlapping test folds, mean AUROC rises from 0.688 to 0.795, precision from 0.710 to 0.757, and recall from 0.552 to 0.679. Selection rises from 0.397 to 0.458 and ρ falls from 0.290 to 0.243. This exploratory comparison uses metadata features rather than the question embeddings used by the main classifier. Popularity is supplied dataset metadata and may not be available from a new question alone. The selected population's precision remains below 0.9. The association motivates further study of popularity as a feature, but deployment and a causal explanation remain unresolved.

S.3 Reference-matching sensitivity

The support ceiling applies to any fixed matching rule, but its measured value will depend on that rule. In an audit of 4,118 cases labeled "unmatched" by the heuristic matcher across the 30 cells and six datasets, Gemini 2.5 Flash-Lite found 168 matches, and measured matchability in each cell increased by 0–5 percentage points. Gemini 2.5 Flash independently looked at the same 4,118 cases. In 97.4% of them, the two judges agreed on whether they were matched or not. Flash agreed with 73 of the 168 new matches found by Flash-Lite. And in no cell did their matchability estimates differ by more than 2.2 percentage points. This is a false-negative sensitivity check of the heuristic matcher. It does not measure the false-positive rate among matches accepted by the heuristic. This agreement applies only to cases adjudicated by the judges, not to all generated answers. And it would not be sufficient to assure the semantic correctness of any particular match they accepted.

Each judge is presented with the question, acceptable reference answers, and a numbered representative of each cluster. They are asked to select a cluster that asserts the same person, date, number, fact or entity as one of the reference answers. The cluster answer could use paraphrase, abbreviations or additional context, but not assert a different fact. They were asked to respond with `ANSWER: N`, where N is the number of the cluster (starting at 1) or 0 if there was no match. Temperature was set to 0. A valid positive index corresponding to one of the clusters would replace the heuristic "unmatched" label. Otherwise, their response (if unparseable, out of range or zero) would leave the case unmatched.

Encoder and matching choices must be fixed before using $\mathcal{V}_2$ or test labels. Tuning them on a separate training or validation split preserves the split-conformal setup. Tuning them on $\mathcal{V}_2$ or test invalidates that setup. Maximizing match rate is not a sufficient tuning objective, since permissive similarity thresholds can increase apparent support while destroying correctness discrimination. The bge-large and e5-large results in Table 4 illustrate this problem. The French medical QA judge study of Belmadani et al. [52] concerns a different domain and does not validate this matcher.

S.4 Weight movement and stability

In Section 2.4, we check whether any set in the selection grid changes. If none of the sets change, then the coefficient cannot be identified using APSS. With $V = \mathcal{V}_1$ and $K_i = K(x_i)$, a small fraction of changing sets would allow only a small change in the mean set size. Indeed, $\mathrm{range}(\mathrm{APSS}) \leq |V|^{-1} \sum_i (K_i - 1)\mathbf{1}\{i \text{ moves}\}$.

A separate sweep on held-out test items covers four split seeds and 30 cells, with λ on a sparse grid from 0 to 10^6. At $\lambda = 0$, 97.7% of sets include all of the sampled clusters (96.3% among items with more than one cluster). Over the sweep, 3.7% of the test items experience a change in set size. Over the sweep's points up to 100, 3.6% of them change. Using all questions in each cell and heuristic labels, 25 of 30 cells have support-failure rate above α. On average over the four seeds, none of these cells changes more than 6.3% of its sets. The other five cells, all of which are of type AmbigInst, move between 0.3% and 37.2% of their sets, the most on Ministral–AmbigInst (37.2%) and Qwen3-32B–AmbigInst (12.8%). Changes in set size are measured on the test split. Support failure rates use all questions in each cell.

We also look at how often the reference of a matched question across all splits lies in the highest-ranked cluster (under deterministic APS, this item would get score zero). In the five cells below α, this holds for 93% of matched items in Qwen3-14B and 94% in Qwen3-8B. These sets move at most 1.6%. For the other three cells, it holds for 79–91% of matched items. Their sets move 11.4–37.2%. Qwen3-14B has a failure rate of 0.095, below α. Taken together, across these five cells, we could not reliably set a cutoff for how concentrated the matched scores are at zero. These shares refer to deterministic APS, whereas the sweep used APS with randomization, where the highest-ranked cluster does not get score zero.

Validation audit on the full grid. We also recompute the diagnostic on $\mathcal{V}_1$ using the 18 main cells, deterministic APS, the seed 42 split, and all 4,011 candidate weights. Table 14 shows three completely flat cells. In the remaining cells, the objective distinguishes some weight regions but still has multiple minimizing grid points. All 18 main coefficients equal the first minimizer in the recomputed grid. Tied fractions range from 5 of 4,011 points to the entire grid. These fractions describe this nonuniform grid, not a confidence region or probability. For example, the minimizing region on Ministral–TriviaQA is narrow, approximately $[26.96, 28.94]$, even though it contains more than one grid point.

Failure rates above α do not rule out movement on $\mathcal{V}_1$. Ministral–TriviaQA, Ministral–AmbigQA, and Qwen3-8B–TriviaQA have validation failure rates of 12.5%, 41.25%, and 23.3%, but 29.2%, 16.2%, and 11.7% of their sets move. Ministral–PopQA also has 44.5% failure and 7.5% movement. These validation results differ from the seed averages in the held-out sweep.

Table 14: Validation calculation on the 18 main cells. n_1 is the selection-split size, f_{move} the fraction of items whose set size changes across the grid, ΔAPSS the maximum minus minimum validation APSS, and tied grid the fraction of candidate weights attaining that minimum. No cell has a unique minimizing grid point.

Model	Dataset	n_1	f_{move} (%)	ΔAPSS	Tied grid (%)
Qwen3-8B	TriviaQA	120	11.7	0.092	10.8
	AmbigQA	80	1.2	0.013	37.3
	OpenNQ	120	0.8	0.017	6.9
	AmbigInst	121	0.0	0.000	100.0
	SimpleQA	200	0.0	0.000	100.0
	PopQA	200	3.5	0.020	44.6
Ministral-3-8B	TriviaQA	120	29.2	0.258	0.2
	AmbigQA	80	16.2	0.125	24.5
	OpenNQ	120	4.2	0.033	40.2
	AmbigInst	121	56.2	1.091	1.0
	SimpleQA	200	1.0	0.010	26.0
	PopQA	200	7.5	0.030	12.7
Gemma-4-E4B-it	TriviaQA	120	5.0	0.042	8.1
	AmbigQA	80	6.2	0.050	34.3
	OpenNQ	120	0.0	0.000	100.0
	AmbigInst	121	39.7	0.198	0.1
	SimpleQA	200	1.0	0.005	41.1
	PopQA	200	2.0	0.015	35.6

S.5 Qwen3-14B and Qwen3-32B

These results use different generated samples from the fixed sampling budget comparison in Appendix T. The same six benchmarks were evaluated with Qwen3-14B and Qwen3-32B. Table 15 uses labels obtained by having a judge revisit heuristic negatives, 40/30/30 splits, and seeds 42–45. Coverage, APSS, and $\lambda^\star$ use the standardized variant of Appendix K with randomized APS and selection over all items in $\mathcal{V}_1^{\text{eval}}$. AUROC is computed from the uncertainty signals on the same held-out subsets. The support rate uses all items, so it need not equal the split-mean coverage exactly. Across all 30 cells of this judged rerun, including the 18 main cells, mean CAELUM coverage tracks $\hat{\pi}$ with a maximum absolute gap of 0.030 (Qwen3-14B–AmbigQA, 0.700 versus 0.730).

The five-model transfer matrix uses heuristic labels for the same 2,104 records, aligned by dataset and question ID. The classifiers are fitted and scored without a held-out question split. Mean source-label AUC is 0.793 and target-label AUC 0.772 over 20 ordered pairs. Every transfer interval has a lower endpoint above 0.72. Table 16 gives the complete matrix. This extension supports cross-generator label transfer for these questions. It does not establish transfer to new domains, unseen questions, or API models.

Table 15: Results for larger generators. Values are means over four overlapping split seeds, except all-question $\hat{\pi}$. Coverage, APSS, and $\lambda^\star$ use the standardized variant of Appendix K. The $\lambda^\star$ column gives the mean $\pm$ SD over these seeds and is not directly comparable with Table 1. **Bold** and underline mark the larger and smaller displayed AUROCs, and displayed ties are both in bold. Win counts use unrounded values. U_{epi} exceeds U_{total} on 8 of these 12 cells, with mean difference $+0.031$. It does not improve ranking on every larger-model cell.

	Support / coverage		Support-failure AUROC ↑				
Dataset	$\hat{\pi}$	Cov.	U_{epi}	U_{total}	Δ_{epi}	APSS	$\lambda^\star$
Qwen3-14B							
TriviaQA	0.840	0.822	**0.744**	**0.744**	-0.001	1.56	0.0±0.0
AmbigQA	0.730	0.700	**0.756**	0.681	+0.075	1.45	0.0±0.0
OpenNQ	0.690	0.686	**0.718**	0.648	+0.070	1.16	0.0±0.0
AmbigInst	0.918	0.910	**0.801**	0.702	+0.098	1.92	0.0±0.0
SimpleQA	0.370	0.365	**0.547**	0.543	+0.005	1.48	0.0±0.0
PopQA	0.550	0.542	**0.746**	0.744	+0.002	1.94	2.1±2.1
Qwen3-32B							
TriviaQA	0.873	0.858	**0.766**	0.741	+0.025	1.61	0.9±1.2
AmbigQA	0.805	0.817	**0.771**	0.711	+0.060	1.53	0.9±1.6
OpenNQ	0.740	0.744	**0.761**	0.701	+0.061	1.22	0.0±0.0
AmbigInst	0.961	0.962	0.696	**0.718**	-0.021	2.63	21.3±19.0
SimpleQA	0.406	0.390	0.566	**0.568**	-0.002	1.80	0.0±0.0
PopQA	0.576	0.563	0.768	**0.770**	-0.002	2.43	1.3±2.2

Table 16: Matchability label transfer across five generators on aligned questions. Source AUC is in-sample, and target AUC uses the same fitted scores. Bootstrap intervals use 1,000 resamples of record labels and fixed fitted scores. These estimates use saved generations from five models, aligned by dataset and question ID. The three-model table uses the corresponding 18-cell subset. The scale comparison in Appendix T uses different generations.

Source LLM	**Target LLM**	**Source AUC↑**	**Transfer AUC↑** (95% CI)	**Agreement**
Qwen3-8B	Ministral-3-8B-Instruct-2512	0.793	0.771 [0.750, 0.791]	0.831
	Gemma-4-E4B-it	0.793	0.797 [0.778, 0.815]	0.817
	Qwen3-14B	0.793	0.752 [0.730, 0.772]	0.854
	Qwen3-32B	0.793	0.763 [0.742, 0.782]	0.846
Ministral-3-8B-Instruct-2512	Qwen3-8B	0.788	0.771 [0.750, 0.790]	0.831
	Gemma-4-E4B-it	0.788	0.782 [0.763, 0.801]	0.796
	Qwen3-14B	0.788	0.747 [0.724, 0.767]	0.822
	Qwen3-32B	0.788	0.764 [0.742, 0.784]	0.831
Gemma-4-E4B-it	Qwen3-8B	0.821	0.782 [0.762, 0.801]	0.817
	Ministral-3-8B-Instruct-2512	0.821	0.766 [0.745, 0.785]	0.796
	Qwen3-14B	0.821	0.755 [0.733, 0.775]	0.797
	Qwen3-32B	0.821	0.769 [0.749, 0.790]	0.786
Qwen3-14B	Qwen3-8B	0.778	0.781 [0.762, 0.801]	0.854
	Ministral-3-8B-Instruct-2512	0.778	0.773 [0.752, 0.791]	0.822
	Gemma-4-E4B-it	0.778	0.800 [0.781, 0.818]	0.797
	Qwen3-32B	0.778	0.782 [0.762, 0.802]	0.870
Qwen3-32B	Qwen3-8B	0.783	0.769 [0.748, 0.788]	0.846
	Ministral-3-8B-Instruct-2512	0.783	0.766 [0.744, 0.785]	0.831
	Gemma-4-E4B-it	0.783	0.787 [0.767, 0.806]	0.786
	Qwen3-14B	0.783	0.756 [0.735, 0.777]	0.870

S.6 Detection of support failure

We compare the additional support-failure scores in Table 17 on the same 29 cells with judged labels, excluding Qwen3-8B/AmbigInst because it has no support failures. The evaluation uses seeds 42–45 and the four-way split described below. CAELUM's learned detector is trained on other cells and evaluated on the omitted cell. The per-cell logistic and gradient-boosting scores use features available after generation. The pre-generation classifier uses question features only, so it has access to less information.

Table 17: Additional detector comparison on 29 cells with judged labels. References identify score definitions. All values come from our evaluation. For the frequency baseline, p_1 is the largest cluster mass. Values are mean ± population SD across cell means, after averaging over seeds 42–45. Higher AUROC and AUPRC and lower FPR at 90% TPR are better. AUPRC's chance baseline is the support-failure prevalence, which varies by cell. **Bold** and underline indicate the best and second-best displayed mean, including ties. These are different splits and labels from the main 18-cell AUROC table.

Support-failure score	AUROC ↑	AUPRC ↑	FPR at 90% TPR ↓
CAELUM detector (ours)	**0.780**±0.08	**0.648**±0.16	**0.542**±0.17
Logistic regression (per cell)	<u>0.716</u>±0.09	<u>0.573</u>±0.17	0.661±0.16
Gradient boosting (per cell)	0.709±0.10	0.566±0.17	0.659±0.16
U_{epi} [11]	0.696±0.10	0.529±0.17	<u>0.636</u>±0.15
U_{total} [11]	0.674±0.10	0.510±0.16	0.678±0.16
U_{ale} [11]	0.604±0.09	0.457±0.16	0.749±0.14
$1 - p_1$ (cluster frequency)	0.585±0.09	0.434±0.17	0.958±0.12
Pre-generation classifier (ours)	0.649±0.09	0.506±0.17	0.720±0.12
Good–Turing missing mass [19]	0.550±0.05	0.400±0.19	1.000±0.00

S.7 Missing mass and abstention at a fixed sampling budget

We compare rules that add a catch-all label for a reference absent from the sampled support. Its augmented coverage counts either a matchable reference included in the semantic set, or an unmatchable reference accompanied by that label. This is an enlarged output space. It does not increase the rate at which correct semantic answers are generated.

The comparison considers 29 non-degenerate cells, omitting Qwen3-8B–AmbigInst. Four split seeds are used for each cell, with judged support labels. To form semantic sets, the standardized variant of Appendix K is used, including its restriction to matched items. From each cell, four splits are taken at 25% each. For weight selection, $\mathcal{V}_1$ is used. To fit the semantic threshold, the first half of $\mathcal{V}_2$ is used. To calibrate triggers, the second half is used. The final 25% is used as test data. CAELUM's detector is trained on cells other than this one. It therefore does not see the first quarter of this cell. To compare against the calibration module in CPQ, the full $\mathcal{V}_2$ and a fixed budget of 50 answers are used.

Learned support-failure detector. This comparison is separate from the main results across the 18 cells. The detector is a logistic regression (L2-regularized, with $C = 1$, balanced class weights) of features standardized over the pool of records from the other cells. Its positive label is lack of a reference match in the sampled clusters under the judged support labels. The 15 features are $U_{\text{epi}}, U_{\text{ale}}, U_{\text{total}}, p_1$ and p_2 (the two largest cluster masses, with $p_2 = 0$ for a single cluster), their difference, K, $1/K$, the Shannon entropy of the cluster masses, the maximum and standard deviation of the entropies of the clarifications, a pre-generation matchability probability of the question, its length in characters and in tokens separated by whitespace, and the proportion of generated answers that refuse the question. For each of the other 29 cells in the full 30-cell collection, all its records are included in the pool. But the cell in question is excluded. One of the features is a pre-generation matchability probability. This comes from a logistic regression using question-only features to predict class (match vs. no match), with $C = 0.1$ and balanced class weights on the pool. On this pool, therefore, the probabilities are in-sample. Both regressions use seed 42 and a 5,000-iteration limit. No hyperparameter tuning was done in either regression. And the scores of items in the target cell were kept fixed across the four splits of the data. Even if a cell is excluded, it is possible that the same

questions from another generator will appear in another cell in the pool. The family-held-out analysis in Appendix S.2 tests a different exclusion rule.

Abstention threshold. Let $a(x)$ be the support-failure score. The catch-all label is added when $a(x) \geq t$. On the second half of $\mathcal{V}_2$, let n_b be the sample size and $\widehat{R}_b(t)$ the fraction not covered by either the semantic set or a correctly added catch-all label. Of the scores 0, $1 + 10^{-9}$, and those actually observed, the rule picks the largest value of t for which

$$\frac{n_b}{n_b + 1} \widehat{R}_b(t) + \frac{1}{n_b + 1} \leq \alpha.$$

If none of these values satisfies this inequality, the catch-all label is added to all sets. In this case, it could still be that a semantic set misses a matched reference, which the catch-all label does not cover. The same threshold rule supports the Good–Turing trigger.

We evaluate the Good–Turing missing mass score used by Noorani et al. [19] and their CPQ calibration module. The Good–Turing trigger uses our risk-controlled threshold rule. Neither row reproduces their adaptive querying policy.

Table 2 reports means across cell means. Abstention rate is the frequency of including the added label. Precision is the fraction of such cases where the reference is absent from support. Parentheses count cells with defined precision. The other means use all 29 cells. Ordinary semantic coverage is 0.652 ± 0.19 and semantic APSS is 2.021 ± 0.77 (cell SD). The CPQ calibration rule has no standalone ranking score, hence the dash. Its higher augmented coverage also comes with more abstention.

At the fixed sampling budget, the calibration module attaches the catch-all label to every set on 24 of 29 cells. CAELUM's learned detector reduces that frequency in this experiment, while its 0.459 abstention rate remains substantial. This extension operates after generation and addresses a different decision from the classifier using question features in Theorem 1. It does not close that classifier's PopQA gap.

T Additional support audits and escalation pilots

Generator scale at a fixed sampling budget. To test whether scaling the generator removes the sampled-support limitation, we regenerate answers with Qwen3-8B [25] and Qwen3-32B [28] on the same $2{,}104$ questions and reference answers as the main experiment. Both models use FP16 without 4-bit quantization, generation seed 42, $n = 5$ self-generated clarifications at temperature 0.7, and $m = 10$ answers per clarification at temperature 0.5 (up to 512 new tokens for the clarification list and 128 per answer, with thinking mode disabled). We retain all-mpnet-base-v2 with clustering threshold 0.7, VNE $\gamma = 1$, and the original heuristic reference matcher (Appendix E). Both columns of Table 18 come from this generation campaign. Neither reuses the main table's generations. The ordered questions and references are identical across the two columns.

Table 18: Support rates at a fixed generation budget. $\hat{\pi}$ is the fraction of references matched to any generated cluster under the heuristic matcher, not calibrated semantic coverage. Counts cover all queries, not a held-out test split. Percentages and differences are descriptive estimates from one generation seed. The final row pools all $2{,}104$ questions with equal weight. **Bold** and underline mark the higher and lower support rates within each row.

Dataset	N	**Matched references /** N **(rate)**		$\Delta\hat{\pi}$ **(pp)**
		Qwen3-8B	Qwen3-32B	
TriviaQA	300	232/300 (77.3%)	253/300 (**84.3%**)	+7.0
AmbigQA	200	128/200 (64.0%)	155/200 (**77.5%**)	+13.5
OpenNQ	300	184/300 (61.3%)	216/300 (**72.0%**)	+10.7
AmbigInst	304	301/304 (**99.0%**)	295/304 (97.0%)	−2.0
SimpleQA	500	175/500 (35.0%)	183/500 (**36.6%**)	+1.6
PopQA	500	265/500 (53.0%)	287/500 (**57.4%**)	+4.4
All questions	2,104	1285/2104 (61.1%)	1389/2104 (**66.0%**)	+4.9

The 32B generator has higher empirical matchability on five of six datasets, with the largest gains on AmbigQA, OpenNQ, and TriviaQA. AmbigInst is already near saturation and decreases slightly. The pooled rate rises from 61.1% to 66.0%, but SimpleQA and PopQA remain at 36.6% and 57.4%. Scaling therefore does not remove the missing-support problem in this fixed sampling budget audit. These measurements do not establish calibrated coverage, cross-scale classifier transfer, or an advantage in epistemic ranking. Those require separate evaluations. They also do not identify the cause of the remaining errors, such as entity popularity.

Scope of the escalation pilots. The following escalation and abstention pilots use previously generated answers and different protocols from the scale comparison above. They are *not* used for the main claims or pooled with those support rates.

Phi-4 fallback escalation. We evaluate a model-escalation policy on 96 pair-seed evaluations using four base LLMs (the three main models plus Qwen3-14B), six datasets, and seeds 42–45, with Phi-4 as the fallback answer generator. The policy answers with the base model unless a risk score computed from the base model triggers escalation. No non-oracle policy uses test labels. Phi-4 is stronger on average. Base top-1 accuracy is 0.589 versus fallback 0.669, and base support is 0.629 versus fallback support 0.713. For set-valued escalation, base coverage is 0.623 versus fallback coverage 0.706, base APSS is 1.88 versus fallback APSS 1.83, and the union support is 0.766. Phi-4 has higher mean support in this pilot, but the useful-escalation pool is only 0.143 for top-1 accuracy and 0.139 for set coverage, while the harmful-escalation pool is non-negligible (0.062 and 0.056 respectively).

Table 19: Phi-4 escalation pilot using previously generated answers. Each entry averages the 96 combinations of base generator, dataset, and split equally. Best non-oracle rows are selected separately at each fixed escalation budget. Oracle rows use test labels and are upper bounds, not deployable policies. Full summaries are in the supplementary release.

Policy	Escalation budget	Accuracy / coverage	Δ vs base	Oracle Δ
Answer top-1				
LearnedNet-RF	20%	0.615 acc.	$+0.027$	$+0.134$
CAELUM margin	50%	0.642 acc.	$+0.054$	$+0.143$
Set coverage				
LearnedNet-RF	20%	0.646 cov.	$+0.023$	$+0.131$
Learned logistic	50%	0.674 cov.	$+0.051$	$+0.139$

The ranking scores only partially identify helpful escalation. For answer escalation, the best helpful-escalation AUROC is 0.625 for U_{total} and 0.613 for U_{epi}. For set-valued escalation, U_{epi} is best at 0.603. A useful escalation rule must distinguish cases where the stronger model adds missing support from those where it replaces a correct base answer with a wrong one.

Qwen3.5-35B abstain pilot. We also ran a small Qwen3.5-35B-A3B pilot on OpenNQ, SimpleQA, and PopQA with an augmented label space that adds "abstain (reference absent from sampled support)". This tests abstention when the reference is absent from sampled support. The augmented set reaches 0.978 mean coverage over the three cells, compared with 0.689 semantic coverage on generated clusters, but it does so with a high abstain rate (0.822) and low abstain precision for unmatched items (0.358). Frequent inclusion of the abstention label raises recall for support failures, so high augmented coverage alone does not establish a deployable answering policy.

Table 20: Qwen3.5-35B-A3B pilot with an added abstention label on three factual datasets. Semantic coverage and APSS count only generated clusters. Augmented metrics also count the "abstain (not in support)" label. Abstention rate is its frequency of inclusion. The final row averages the three datasets equally, each with 30 test questions. Lower APSS and abstention rates are preferable at comparable coverage.

Dataset	Match rate	Coverage		APSS		Abstention rate
		Semantic	Augmented	Semantic	Augmented	
OpenNQ	0.767	0.767	1.000	1.47	2.30	0.833
SimpleQA	0.633	0.633	0.967	2.33	3.30	0.967
PopQA	0.667	0.667	0.967	4.17	4.83	0.667
Dataset mean	0.689	0.689	0.978	2.66	3.48	0.822

U Examples of generated answers

Table 21 contains three Qwen3-8B/TriviaQA records checked against the source uncertainty and raw generation records. Questions, reference answers, and representative responses are reproduced verbatim. Cluster proportions and uncertainty values come from those records. We recompute R with the main cell's $\lambda^\star$ and $\epsilon = 10^{-6}$, and determine set size using its main threshold. These examples illustrate the recorded outputs, not identified uncertainty regimes.

Table 21: Examples from the Qwen3-8B/TriviaQA cell in the main experiments. Labels describe whether the reference matches sampled support. They do not infer an intervention's benefit from R or treat prompt variability as annotated ambiguity. A generated response can be incorrect even when it is reproduced faithfully here.

1. Reference in a single sampled cluster	
Source	Qwen3-8B / TriviaQA, question ID 251; zero-based cache row 251.
Question	In which city was Pablo Picasso born, in 1881?
Reference	Malaga
Clusters	(1.00) "Pablo Picasso was born in Malaga, Spain, in 1881."
Signals	$U_{\text{ale}} = 0.212$, $U_{\text{epi}} = 0.007$, $R(x) = +0.45$, $\lvert\mathcal{C}(x)\rvert = 1$.
2. Reference absent from sampled clusters	
Source	Qwen3-8B / TriviaQA, question ID 39; zero-based cache row 39.
Question	The Museum of Bags and Purses is in which European city?
Reference	Amsterdam
Clusters	(0.72) "The Museum of Bags and Purses is located in the European city of Antwerp, Belgium." (0.28) "The Museum of Bags and Purses is located in the European city of Madrid, Spain."
Signals	$U_{\text{ale}} = 0.270$, $U_{\text{epi}} = 0.392$, $R(x) = +4.23$, $\lvert\mathcal{C}(x)\rvert = 2$.
3. Reference present among three sampled clusters	
Source	Qwen3-8B / TriviaQA, question ID 35; zero-based cache row 35.
Question	Who directed the Noughties films One Day in September, The Last King of Scotland and Touching the Void?
Reference	KEVIN MACDONALD
Clusters	(0.96) "The director responsible for the Noughties films *One Day in September*, *The Last King of Scotland*, and *Touching the Void* is Kevin Macdonald." (0.02) "The three Noughties films were directed by Andy Anderson." (0.02) "The three Noughties films were directed by Mike Leigh."
Signals	$U_{\text{ale}} = 0.429$, $U_{\text{epi}} = 0.412$, $R(x) = +3.82$, $\lvert\mathcal{C}(x)\rvert = 3$.

Scope of the checks. Raw generation strings verify the origin of the displayed representatives. The uncertainty records contain proportions but not a member-to-cluster assignment for every answer. Multiplying each proportion by 50 recovers integer cluster counts, but is not an independent rerun of the embedding clustering. The examples therefore verify the source records and arithmetic rather than certify every semantic-clustering decision.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the matchability ceiling, learned-stratum bound, empirical findings, and limits.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 5 and Appendices S.2, S.3, and S.1.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Appendix B states the assumptions and proves the support-conditional and learned-stratum results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 and Appendices E and S.1 give the sampling settings, splits, seeds, grids, matching rules, and classifier settings. The GitHub repository documents the software environment for each experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code, evaluation inputs, and reproduction instructions are available at `https://github.com/Unco3892/caelum`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sections 3 and S.1 specify sampling, splits, scores, tuning, and classifier settings. Generators are not fine-tuned.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report split SDs and transfer bootstrap intervals, distinguishing split from generation variability.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix S.1 describes the workload and cluster, and the repository README reports the GPU types, host memory, and GPU-hours of one complete generation run.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We evaluate reliability on existing QA benchmarks, without human-subject experiments.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss reliability benefits and risks from nominal coverage, matcher errors, and excessive abstention.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No new foundation model or high-risk dataset is released.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Appendix S.1 and the accompanying asset record identify the sources, licenses, and service terms. Third-party assets retain their upstream licenses and notices.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The repository documentation and Appendix S.1 describe the assets, use, provenance, limitations, and license status.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or human-subject research was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human-subject research was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Sections 2.2 and 3 describe experimental LLM use. LLM tools also assisted writing, proof verification, and code checks.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.