Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

XL-CARR: Cross-Lingual Constrained Adaptive Retrieval Routing with Conformal Risk Control for Low-Resource Question Answering

Hariom Pandya
Computer Engineering Department, Dharmsinh Desai University, Nadiad, Gujarat, India.

E-mail: hariompandya.ce@ddu.ac.in

ORCID ID: 0000-0001-5766-7293

Abstract

Adaptive retrieval-augmented generation (RAG) decides for each query whether and how to retrieve, but routers such as MBA-RAG choose only the number of retrieval steps and trade accuracy against cost through hand-tuned reward weights. For low-resource languages (LRLs), whose native corpora are small, the more consequential decision is where to retrieve from: pivoting through English widens coverage at the risk of translation errors, and multilingual readers often answer confidently but incorrectly. This paper proposes XL-CARR, which casts cross-lingual adaptive RAG as a constrained online decision over 18 feasible combinations of retrieval source (closed-book, native, English pivot or transliterated), tool and reader, chosen from a language-, typology- and retrievability-aware context. Primal–dual Thompson sampling maximizes answer quality under budgets on latency and incorrect answers, and a two-stage conformal cascade, calibrated per language, accepts, escalates or abstains with a finite-sample bound on the probability of emitting an incorrect answer. On 23 languages in 29 test sets from TyDiQA-GoldP, XQuAD, MLQA and IndicQA, the policy alone improves average F1 over an adapted MBA-RAG with the same sources and readers by 1.4–1.7 points on the first three benchmarks and by 2.5 points on IndicQA, at a 37–38% lower cost. With the same cascade applied to both routers, XL-CARR answers 2.9–3.3 more questions per hundred while the marginal error probability stays below 0.30 in every language; the error rate among answered questions remains 28.0% on average.

Keywords: Retrieval-Augmented Generation, Cross-Lingual Question Answering, Low-Resource Languages, Indic Languages, Constrained Online Learning, Conformal Risk Control.

1. Introduction

Retrieval-augmented generation (RAG) conditions the reader on retrieved passages instead of relying only on its parameters [1, 2] and increasingly underlies deployed assistants [3]. Adaptive retrieval decides when to retrieve from model uncertainty during generation [4] or from the estimated complexity of the query [5]. MBA-RAG [6] treats the choice among no, single-step and multi-step retrieval as a bandit with a step-penalized reward, saving about 20% of retrieval steps on English benchmarks.

DOI: https://doi.org/10.36548/jtcsst.2026.4.008
Received: 18.09.2026, received in revised form: 22.10.2026, accepted: 09.11.2026, published: 23.11.2026
© 2026 Inventive Research Organization. This is an open access article under the Creative Commons Attribution-NonCommercial International (CC BY-NC 4.0) License

These designs rely on assumptions that are difficult to sustain for LRLs. For many languages, the native Wikipedia is one to three orders of magnitude smaller than the English one [7], and many non-English questions are answerable only from English documents [8], so the more consequential decision is which corpus to search. Translation-based pipelines are sensitive to translation errors and document-language preferences [9, 10] precisely where pivoting is most needed: translation degrades when parallel data are scarce [11, 12], and its artifacts propagate to later stages [13], e.g., as paraphrased answer spans on MLQA [14].

Moreover, the reward $A - \lambda C$ of MBA-RAG must be re-calibrated per dataset, controls neither latency nor incorrect answers, and its ε-greedy exploration ignores estimate uncertainty. Multilingual models make QA feasible for LRLs [15, 16], yet without verification every system evaluated here errs on more than half of the TyDiQA questions (Section 5.1), so a guarantee on how often an incorrect answer reaches the user is needed. This paper therefore proposes XL-CARR (Cross-Lingual Constrained Adaptive Retrieval Routing), with three contributions:

(1) Cross-lingual adaptive RAG is cast as a constrained decision problem, akin to contextual bandits with knapsacks [17], over $K = 18$ source–tool–reader actions (Section 3.3, Table 1) with a language-, typology- and retrievability-aware context.

(2) Primal–dual Thompson sampling is combined with a per-language cascade calibrated by conformal risk control [18] that bounds $\Pr(\text{emit} \wedge \text{incorrect}) \leq \alpha$ using natural language inference (NLI), semantic entropy [19] and machine translation (MT) quality estimation (QE) [20]; Proposition 1 and Appendix A prove the bound.

(3) The evaluation spans TyDiQA-GoldP [21], XQuAD [22], MLQA [23] and held-out native and Romanized IndicQA [16] (23 languages, 29 test sets).

Scope of the Claims. The individual components are established techniques; the contribution lies in combining them into a single constrained decision over evidence sources for LRLs and in a controlled evaluation of that decision (Section 2.1); no new regret bound and no new conformal theorem are claimed. The gains are largest for languages with small native corpora and absent where a single inexpensive source suffices, and learning the policy requires a labeled stream during development, after which the policy is frozen. Sections 2–6 cover related work, the method, the setup, the results and the conclusion; Appendix A proves the guarantee and Appendix B gives reproduction details.

2. Related Work

Contextual Bandits. LinUCB [24], NeuralUCB and NeuralTS [25, 26] and neural-linear Thompson sampling [27] learn from bandit feedback with linear, neural and refitted-representation models. Bandits with knapsacks add budgets [28], priced by primal–dual algorithms with Lagrange multipliers [17], also with Thompson sampling [29].

Adaptive Retrieval and Routing. DRAGIN decides during generation when to retrieve [4], so its cost is hard to bound in advance. Adaptive-RAG labels each query with the cheapest strategy that answers it [5], MBA-RAG replaces these labels with bandit feedback [6], and PILOT selects large language models (LLMs) under a knapsack budget [30]; such routers act within one language and do not bound incorrect answers.

Cross-Lingual QA and Multilingual RAG. Indic transfer has been improved by Indic pre-training [16] and language adapters [15], and multilingual RAG prefers high-resource and

query-language documents [9]. CrossRAG translates retrieved documents before generation [10], CORAL re-selects corpora when evidence is insufficient [31], and answer-aligned cross-lingual retrieval improves generative QA on MLQA [14]; All fix the evidence language or switch on hand-set thresholds.

Error Detection and Conformal Guarantees. Semantic entropy [19] and entailment checks estimate uncertainty but do not bound how often an error reaches the user. Conformal risk control extends conformal guarantees to any bounded loss monotone in a threshold [18]; In NLP [32] conformal methods give prediction sets for multiple-choice QA [33], factuality guarantees [34], calibrated generation [35] and answer sets for RAG [36], Fisch et al. [37] combined them with a model cascade. XL-CARR instead bounds the probability of returning a single incorrect answer, per language.

Positioning. Adaptive-RAG and MBA-RAG decide the number of retrieval steps and DRAGIN decides when to retrieve, all over a fixed evidence language. PILOT chooses the generator under a knapsack budget. CrossRAG fixes an English pivot and CORAL re-selects corpora iteratively. TRAQ controls the coverage of answer sets. None of these methods both constrains cost and bounds the rate of incorrect answers. XL-CARR decides where evidence comes from, under an explicit latency budget and a per-language bound $\Pr(\text{emit} \wedge \text{incorrect}) \leq \alpha$, and learns when pivoting is worth its translation risk. Adaptive-RAG and MBA-RAG are re-implemented over the same sources and readers, and DRAGIN, CrossRAG, and CORAL are baselines (Section 4.3).

2.1 Methodological Novelty Relative to Existing Techniques

The novelty of XL-CARR lies in how three established method families are instantiated and coupled for cross-lingual evidence selection, not in new learning theory.

Contextual Bandits. The learning rule is standard neural-linear Thompson sampling with Sherman–Morrison updates [27]. What differs from MBA-RAG [6] and PILOT [30] is the decision: each arm is a constrained triple of source, tool and reader (Section 3.3), so the bandit chooses the language of the evidence; the context contains retrievability predictors and a typological vector that allows transfer to languages without bandit feedback (Section 5.8); Additionally, quality, cost and error are predicted by three heads instead of one scalar reward.

Primal–Dual Thompson Sampling. Pricing budgets with Lagrange multipliers by projected dual ascent [17, 29] is reused as is. Specific to XL-CARR are the simultaneous pricing of an observable cost and an error rate observable only with references, a latency budget offset by the unseen verification cost, and multipliers frozen for deployment so that the conformal guarantee applies. Since the representation is refitted, the regret guarantees of Agrawal and Devanur [17] do not transfer, and none is claimed.

Conformal Risk Control. The finite-sample argument is that of conformal risk control [18], and per-language calibration is Mondrian conformal calibration [38]. Specific are a two-stage cascade whose second stage is a different source–tool–reader action rather than a larger model of the same kind [37], a per-stage risk split proved valid under sequential escalation (Proposition 1), and an error score in which MT quality estimation lets translation errors of pivoted actions enter.

3. Proposed Methodology

Fig. 1 shows the components of XL-CARR: a context encoder, a budgeted policy over source, tool and reader, a verification cascade and an offline warm start.

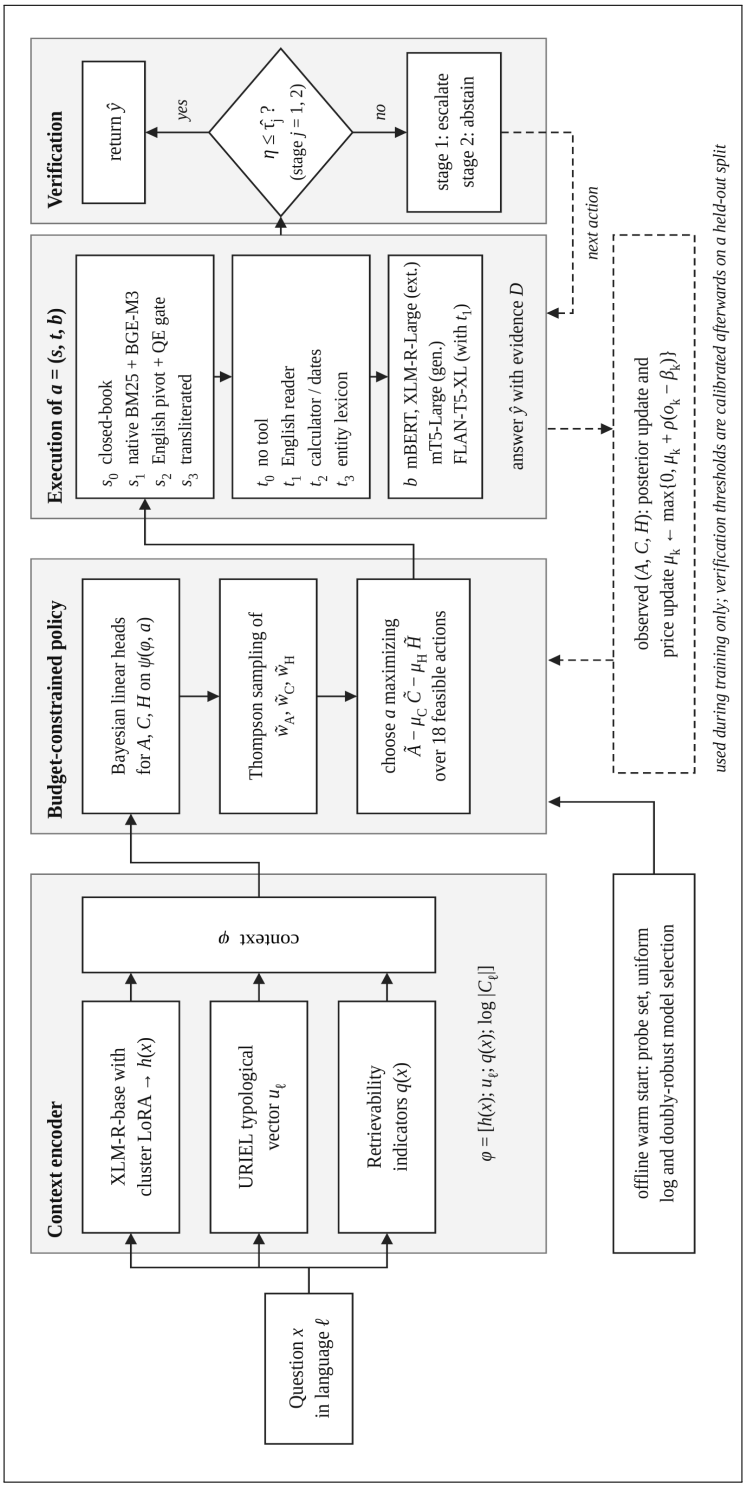

Figure 1. Architecture of XL-CARR. Dashed paths are used only during training. The policy selects one of the 18 feasible actions $a = (s, t, b)$ of Table 1; the verification stage implements the two-stage cascade of Section 3.6, with thresholds $\hat{\tau}_1, \hat{\tau}_2$ calibrated per language after the policy is frozen.

3.1 Problem Statement

For each language $\ell \in \mathcal{L}$, a native corpus $\mathcal{C}_\ell$ and an English corpus $\mathcal{C}_{en}$ are available,typically with $|\mathcal{C}_\ell| \ll |\mathcal{C}_{en}|$. Given a question x in ℓ, the system returns an answer $\hat{y}$ in ℓ, or abstains, by selecting $a = (s, t, b)$ of source s, tool t and reader b from a feasible set $\mathcal{A}$, based on a context ϕ (Section 3.2), yielding quality A, normalized cost C and an emitted-error indicator H in $[0, 1]$ (Section 3.4), observed only for the executed action. A and H need reference answers, so learning uses a labeled development stream; the deployed policy is frozen. The policy π should solve

$$\max_\pi \mathbb{E}[A] \quad \text{subject to} \quad \mathbb{E}[C] \le \beta_C, \quad \mathbb{E}[H] \le \beta_H \tag{1}$$

where β_C is the permitted average latency as a fraction of the most expensive configuration and β_H the tolerated rate of incorrect emitted answers. With $A^\star$ the value of the optimal randomized policy for Eq. (1), learning over T questions (rounds i) is assessed by the regret and constraint violation

$$R_T = \sum_{i=1}^{T} (A^\star - A_i), \qquad V_C(T) = \left[\sum_{i=1}^{T} C_i - T\beta_C \right]_+ \tag{2}$$

with $V_H(T)$ defined analogously and $[z]_+ = \max\{z, 0\}$.

3.2 Typology- and Retrievability-Aware Context

The query is encoded by XLM-R-base with LoRA adapters shared within k-means clusters ($k = 4$) of URIEL vectors: {bn, hi, te, tr}, {ar, id, sw}, {th, vi, zh} and {el, fi, ro}; every IndicQA language falls in the first cluster. Mean pooling gives $h(x) \in \mathbb{R}^{768}$. A typological vector $u_l \in \mathbb{R}^{32}$ is the principal component analysis (PCA) projection of URIEL syntax, phonology and inventory features over the 24 evaluated languages. Pre-retrieval predictors $q(x)$, computed against both indices, indicate whether the native corpus likely holds the answer: IDF statistics, a simplified clarity score, the out-of-vocabulary rate, the top-1 similarity and margin of a BGE-M3 probe, and the share of named-entity tokens. The context is

$$\phi = [h(x); u_\ell; q(x); \log |\mathcal{C}_\ell|] \tag{3}$$

and the cost of the probes is charged to every action. A shared two-layer network maps $[\phi; e(a)]$, with an action embedding $e(a)$, to $\psi(\phi, a) \in \mathbb{R}^{128}$, on which three linear heads predict quality, cost and error.

3.3 Action Space

The sources are s_0 (closed-book); s_1 (native BM25 and BGE-M3 [39] retrieval fused by reciprocal rank fusion, RRF); s_2 (pivot: query translation with IndicTrans2 [40] or NLLB-200, English Wikipedia retrieval and entity-preserving back-translation); and s_3 (transliterated: Romanized tokens converted with IndicXlit [41] and the query expanded with its top-3 variants). The tools are t_0 (none), t_1 (an English reader), t_2 (a calculator and date normalizer) and t_3 (a

bilingual Wikidata lexicon for rare entities). The readers are b_1 = mBERT, b_2 = XLM-R-Large, b_3 = mT5-Large and b_4 = FLAN-T5-XL, the English reader of t_1.

Formal Definition. Let $\mathcal{S} = \{s_0, \dots, s_3\}$, $\mathcal{T} = \{t_0, \dots, t_3\}$ and $\mathcal{B} = \{b_1, \dots, b_4\}$. Of the $|\mathcal{S} \times \mathcal{T} \times \mathcal{B}| = 64$ triples, $a = (s, t, b)$ is feasible if and only if (C1) closed-book generation needs a generative reader and no tool, $s = s_0 \Rightarrow (t, b) = (t_0, b_3)$; (C2) the English reader needs English evidence and replaces the reader, $t = t_1 \Leftrightarrow b = b_4 \wedge s \in \{s_2, s_3\}$; (C3) the calculator and the entity lexicon post-process extracted spans of XLM-R-Large, $t \in \{t_2, t_3\} \Rightarrow b = b_2 \wedge s \neq s_0$; and (C4) without a tool, any multilingual reader may be used, $t = t_0 \wedge s \neq s_0 \Rightarrow b \in \{b_1, b_2, b_3\}$. With $\mathcal{S}^+ = \{s_1, s_2, s_3\}$, the feasible set is

$$\begin{aligned} \mathcal{A} = {} & \{(s_0, t_0, b_3)\} \cup \{(s, t_0, b) : s \in \mathcal{S}^+, b \in \{b_1, b_2, b_3\}\} \\ & \cup \{(s, t_2, b_2) : s \in \mathcal{S}^+, t \in \{t_2, t_3\}\} \cup \{(s, t_1, b_4) : s \in \{s_2, s_3\}\} \end{aligned} \tag{4}$$

so that $K = |\mathcal{A}| = 1 + 9 + 6 + 2 = 18$ (Table 1). The feasible set does not depend on the question; the context enters only through the predicted outcomes. The QE gate acts after selection: if CometKiwi [20] scores the translated query of a selected action with $s \in \{s_2, s_3\}$ below τ_{QE}, the action is replaced by the highest-valued action under Eq. (6) that does not use the translated query, and the translation cost is still charged. The action embedding $e(a) = E_S[s] + E_T[t] + E_B[b] + E_{ST}[s, t]$ shares parameters among actions with a common component.

Table 1. Complete enumeration of the $K = 18$ feasible actions $\alpha = (s, t, b)$ of Eq. (4). Evidence: language of the passages that are read (+En: also English when the query is translated); a_{10} is the Always-Pivot configuration that defines the unit of cost; -L: Large.

Index	**Source** s	**Tool** t	**Reader** b	**Evidence**	**Index**	**Source** s	**Tool** t	**Reader** b	**Evidence**
a_1	s_0 closed-book	t_0 none	mT5-L	none	a_{10}	s_2 pivot	t_1 Engl. reader	FLAN-T5-XL	English
a_2	s_1 native	t_0 none	mBERT	native	a_{11}	s_2 pivot	t_2 calc./date	XLM-R-L	English
a_3	s_1 native	t_0 none	XLM-R-L	native	a_{12}	s_2 pivot	t_3 entity lex.	XLM-R-L	English
a_4	s_1 native	t_0 none	mT5-L	native	a_{13}	s_3 translit.	t_0 none	mBERT	native (+En)
a_5	s_1 native	t_2 calc./date	XLM-R-L	native	a_{14}	s_3 translit.	t_0 none	XLM-R-L	native (+En)
a_6	s_1 native	t_3 entity lex.	XLM-R-L	native	a_{15}	s_3 translit.	t_0 none	mT5-L	native (+En)
a_7	s_2 pivot	t_0 none	mBERT	English	a_{16}	s_3 translit.	t_1 Engl. reader	FLAN-T5-XL	English
a_8	s_2 pivot	t_0 none	XLM-R-L	English	a_{17}	s_3 translit.	t_2 calc./date	XLM-R-L	native (+En)
a_9	s_2 pivot	t_0 none	mT5-L	English	a_{18}	s_3 translit.	t_3 entity lex.	XLM-R-L	native (+En)

3.4 Outcome Signals

Quality is token F1, $A = \mathrm{F1}\,(\hat{y}, y)$ ($A = 0$ for abstention); cost is the latency of all invoked components, probes included, normalized by the most expensive configuration. During training, $H = 1$ if an emitted answer has F1 below 0.5, which also counts surface-form variants as errors (Section 5.9). At test time, H is estimated from the entailment probability p_{ent} of the question and answer given the top passage (XLM-R-Large fine-tuned on XNLI); the normalized entropy SE over meaning classes of five candidates grouped by bidirectional entailment [19] (semantic entropy for generative readers, span-level semantic dispersion over the top-5 spans for extractive readers); and, for translated actions, the CometKiwi estimate $\hat{q}_{MT}$ reused from the QE gate. A logistic calibrator fitted on the probe set gives

$$\eta = \sigma\left(\gamma_0 + \gamma_1(1 - p_{\text{ent}}) + \gamma_2 SE + \gamma_3(1 - \hat{q}_{MT})\right) \tag{5}$$

The closed-book action uses SE alone. A gate on the token entropy ε_1 of the first candidate sets the score to $\sigma\left(\delta_0 + \delta_1 \varepsilon_1\right)$ when $\varepsilon_1 \notin [\kappa_{lo}, \kappa_{hi}]$. Because the gate, the calibrator and all models are fixed before calibration, the resulting score $\tilde{\eta}$ is a deterministic function of the question and the executed action, up to sampling noise of the generative reader drawn independently per question; this is the property used in Proposition 1.

3.5 Budget-Constrained Policy Learning

In the neural-linear design, ψ is fixed between refreshes and each head is a Bayesian linear regression sharing one precision matrix with rank-one (Sherman–Morrison) updates. At each round, $\tilde{w}_k$ is sampled from head $k \in \{A, C, H\}$ and

$$a = \arg\max_{a' \in \mathcal{A}} \langle \tilde{w}_A - \mu_C \tilde{w}_C - \mu_H \tilde{w}_H, \psi(\phi, a') \rangle \tag{6}$$

so that uncertain cost and error estimates are also explored. After each outcome o_k, the multipliers are updated by projected dual ascent,

$$\mu_k \leftarrow \min\left\{M, \max\left\{0, \mu_k + \rho(o_k - \beta_k)\right\}\right\}, \quad k \in \{C, H\} \tag{7}$$

initialized at $\mu_C = \mu_H = 0$, with upper bound $M = 10$ and step size $\rho = M/\sqrt{T} \approx 0.056$ for $T = 31{,}563$ planned rounds. At deployment, the multipliers are frozen at their average over the last 5,000 rounds and posterior means are used. Every T_r rounds the network and adapters are refitted. Because verification adds latency unseen during learning, the policy budget is $\beta_C = \beta_C^{dep} - \hat{c}_{ver}$, with β_C^{dep} the deployment budget and $\hat{c}_{ver}$ the verification and escalation cost estimated on validation data.

3.6 Conformal Accept–Escalate–Abstain Cascade

The action of Eq. (6) is the first of $J = 2$ stages; at stage j, the answer is accepted if $\tilde{\eta}^{(j)} \le \hat{\tau}_j$. A rejected first-stage answer escalates to the action with the highest posterior-mean quality expected to exceed it (or abstains if none exists); a rejected second-stage answer abstains. For calibration example m, let $H_m^{(1)}, \tilde{\eta}_m^{(1)}$ be the error indicator and score of the first-stage action and $H_m^{(2)}, \tilde{\eta}_m^{(2)}$ those of its escalation target, computed for every calibration example whether or

not it would be escalated ($\tilde{\eta}_m^{(2)} = +\infty$, $H_m^{(2)} = 0$ if no target exists). With $\alpha_1 + \alpha_2 = \alpha$ and the candidate set $\Lambda_j = \{-\infty\} \cup \{\tilde{\eta}_m^{(j)}\}_{m=1}^n$, the threshold of stage j is

$$\hat{\tau}_j = \max \left\{ \tau \in \Lambda_j : \frac{1}{n+1} \left(\sum_{m=1}^{n} H_m^{(j)} \mathbf{1} \left[\tilde{\eta}_m^{(j)} \leq \tau \right] + 1 \right) \leq \alpha_j \right\} \tag{8}$$

If no candidate qualifies, which happens exactly when $n < 1/\alpha_j - 1$, $\hat{\tau}_j = -\infty$ and the stage accepts nothing; with $\alpha_1 = 0.20$ and $\alpha_2 = 0.10$, a non-trivial threshold requires $n \geq 4$ and $n \geq 9$. Because pooled calibration bounds only the language-averaged risk, thresholds are calibrated per language (Mondrian calibration [38]), and because the guarantee requires a frozen policy, the cascade is applied post hoc, with $\alpha_1 = 2\alpha/3$ and $\alpha_2 = \alpha/3$.

Proposition 1 (Finite-Sample Risk of the Two-Stage Cascade). Fix a language ℓ Assume that the policy, the escalation rule, the error score $\tilde{\eta}$ and all models are fixed independently of the calibration and test data of ℓ, and that the n calibration examples and a new test question of ℓ are exchangeable. Let the cascade accept the first-stage answer if $\tilde{\eta}^{(1)} \leq \hat{\tau}_1$, otherwise accept the escalated answer if $\tilde{\eta}^{(2)} \leq \hat{\tau}_2$, and otherwise abstain, with $\hat{\tau}_1, \hat{\tau}_2$ from Eq. (8). Then, over the calibration sample and the test question,

$$\Pr(\text{emit} \wedge \text{incorrect} | \ell) \leq \alpha_1 + \alpha_2 = \alpha \tag{9}$$

The emitted-error indicator is bounded by two monotone stage losses, which is mentioned as $L^{(j)}\left(\hat{\tau}_j\right) = H^{(j)} \mathbf{1}\left[\tilde{\eta}^{(j)} \leq \hat{\tau}_j\right]$, each controlled at level α_j by conformal risk control, and linearity of expectation gives the bound without independence between the thresholds (Appendix A). Calibrating $L^{(2)}$ over all n examples, not only escalated ones, preserves exchangeability (Remark 1).

Interpretation. The bound concerns the fraction of all questions that receive an incorrect answer, not the conditional error $\Pr(\text{incorrect} \mid \text{emit}) = \Pr(\text{emit} \wedge \text{incorrect}) / \text{coverage}$ faced by a user who receives an answer, which can exceed α when coverage is low (Section 5.3); the conditional error is therefore always reported alongside.

3.7 Off-Policy Warm Start

N_p probe questions per language are run through all actions to pre-train the network and heads, and a second log is collected under uniform exploration. The posterior scale, M (which also fixes ρ) and τ_{QE} are selected offline with a doubly-robust (DR) estimator, unbiased because the logging probabilities are known.

4. Experimental Setup

4.1 Data and Protocol

As in XOR-TyDi [8], gold passages are withheld and serve only as an oracle. TyDiQA-GoldP (primary; ar, bn, fi, id, sw, te; development set used as test with 921/113/782/565/499/669 items) is natively written, whereas XQuAD (el, hi, ro, th, tr, vi; 1,190 items each) and MLQA (ar, hi, vi, zh; 5,335/4,918/5,495/5,137 test items) derive from English Wikipedia, which may favor pivoting. MLQA also provides generalized cross-lingual

transfer (G-XLT); IndicQA (eleven languages) is held out. Since no Romanized IndicQA exists, two synthetic versions of its questions are built (contexts and answers stay in native script): one transliterated with IndicXlit [41], and one romanized with uroman [42] plus rule-based noise imitating informal typing (vowel omission, variant spellings, English words for common terms). Korean and Russian TyDiQA (large native corpora) are reported in the text.

Data Partitions and Leakage Control. A single controller serves all benchmarks; Table 2 lists the six roles that data play. The 38,070 TyDiQA training questions are divided into probes, validation and a once-traversed stream (39% Arabic), and no XQuAD or MLQA test question is used for training. Three rules separate the roles. First, every evaluation set is used only for calibration and testing, split 20:80 at random within each language for each of 100 splits; no evaluation question is seen by a reader, the controller, the calibrator of Eq. (5) or any selection step. Second, the validation split is disjoint from the probes and the stream. Third, XQuAD translates the SQuAD v1.1 development set, whereas the readers, t_1 and the translate-train probes use only the SQuAD training set. The only overlap is that the readers are fine-tuned on the TyDiQA training split; this inflates stream scores but not test scores, and five-fold cross-fitted readers change the test F1 of the policy by only 0.2 points.

Table 2. Data partitions. Evaluation sets are split 20:80 into calibration and test portions within each language; sizes of these portions are given for TyDiQA.

Role	Source	Size	Used For
Reader training	SQuAD v1.1 train; TyDiQA train per language; NQ, TriviaQA (t_1 only)	full splits	Readers, t_1
Controller probes	500 per language and source: TyDiQA train (6), translate-train SQuAD (6), MLQA dev. (4)	8,000	Warm start, calibrator (Eq. 5)
Validation	10% of the TyDiQA training split after removing the probes	3,507	Hyper-parameters, α, $\hat{c}_{ver}$
Bandit stream	Remaining TyDiQA training questions, one pass	31,563	Online policy learning
Calibration	20% of each evaluation set, per language	710	Thresholds $\hat{\tau}_1$, $\hat{\tau}_2$
Test	80% of each evaluation set, per language	2,839	All reported test results

4.2 Implementation

The deployment budget is $\beta_C^{dep} = 0.59$ and the estimated overhead $\hat{c}_{ver} = 0.27$, hence $\beta_C = 0.32$, with $\beta_H = 0.50$ and $\alpha = 0.30$, i.e., $(\alpha_1, \alpha_2) = (0.20, 0.10)$; latency is measured on one NVIDIA A100 (80 GB). Appendix B gives corpora, preprocessing, indices and all hyper-parameters.

Choice of the Operating Point. Every always-answering system errs on at least 52.3% of the TyDiQA questions, so $\alpha = 0.15$ already leaves half of the questions unanswered (Section 5.7). $\alpha = 0.30$ is the most permissive level at which at least seven in ten delivered answers are correct on the disjoint validation split, fixed before calibration, so the guarantee remains valid. $\beta_H = 0.50$, just above the error of the unconstrained policy (47.2%), guards

against policies that trade accuracy for cost.

4.3 Baselines and Metrics

Baselines are closed-book mT5-Large; Always-Native (XLM-R-Large); Always-Pivot (translate-test with t_1, or XLM-R-Large on English evidence); a merged BGE-M3 index over both corpora; Adaptive-RAG [5] with an XLM-R classifier on least-cost-correct labels; MBA-RAG [6] with ε-greedy selection, reward $A - \lambda C$ and a DistilmBERT encoder; and DRAGIN [4], CrossRAG [10] and CORAL [31] on the same corpora. Adaptive-RAG and MBA-RAG (‡) use the same sources, readers and corpora as XL-CARR. As references, an oracle router picks the best action per question, and a full-information router fits A, C and H on the offline outcomes of all 18 actions.

Two Comparison Regimes. In the policy-only regime, no system uses the cascade and every question is answered; all F1, EM and cost improvements in the abstract and Section 5.1 belong to this regime. In the cascade regime, the same cascade, calibrated identically, is applied to Adaptive-RAG, MBA-RAG and XL-CARR, each escalating by its own value estimates, so differences in coverage and AURC reflect only the routers.

EM and F1 use the MLQA normalization after NFC Unicode normalization; cost is scaled so that Always-Pivot equals 1.00. Reliability is measured by coverage, the emitted-error rate Err. $= 100 \cdot \Pr(\text{emit} \wedge \text{F1} < 0.5)$, the conditional error among answered questions, the area under the risk–coverage curve (AURC) and AUROC. Values after $\pm$ are standard deviations over seeds; † marks $p < 0.05$ against the highest-scoring baseline under paired bootstrap (10,000 resamples) with Holm–Bonferroni correction; coverage and AURC differences carry 95% intervals over the 100 splits, and per-language quantities carry Wilson intervals [43].

5. Results and Discussion

5.1 Main Results

For the policies alone (Tables 3 and 4), XL-CARR improves average F1 over MBA-RAG by 1.5 points on TyDiQA, 1.7 on XQuAD and 1.4 on MLQA, at a 37–38% lower cost (0.31–0.32 against 0.49–0.52). The gains concentrate in the smallest native corpora: on TyDiQA, +2.8 F1 for Swahili and Telugu and +2.1 for Bengali, against +0.9 and +1.2 for Finnish and Indonesian. For Arabic, with the largest native Wikipedia, XL-CARR trails MBA-RAG by 0.3 F1, as it answers most questions natively; likewise for Korean and Russian (38.9 and 43.8 F1 at cost 0.29–0.30, against 39.2 and 44.3 at 0.45–0.47), where pivot shares of 0.12–0.14 show that the router correctly avoids pivoting.

Always-Native beats Always-Pivot for Arabic, Finnish and Indonesian but trails by 5.8–8.0 F1 for Bengali, Swahili and Telugu; since reading English evidence with XLM-R-Large instead of t_1 costs only 1.1 F1, the pivot advantage comes mostly from the corpus. The merged index (40.3) beats both static pipelines, consistent with Pandya et al. [14]. The online policy comes within 0.6 F1 of the full-information router (44.6) while observing one of 18 outcomes per question. As upper bounds, reading the gold passage gives 74.9/72.0/70.0 F1 on TyDiQA/XQuAD/MLQA and an oracle that picks the best of the 18 actions per question gives 54.9/55.3/52.6, so source selection leaves about 11 F1 of headroom.

Table 3. Open-retrieval results on TyDiQA-GoldP (macro averages). Bold: highest per language among the policy-only systems; †: $p < 0.05$ (paired bootstrap, Holm); $\pm$: s.d. over five seeds; ‡: re-implemented on the same sources; shaded: full-information reference router (Section 4.3).

	F1 per language						Average				
Method	ar	bn	fi	id	sw	te	EM	F1	Cost↓	Cov.	Err.↓
Full-information router	45.2	41.8	49.9	49.1	42.6	38.9	32.6	44.6	0.32	100.0	51.9
Closed-book mT5-Large	25.4	16.4	27.7	27.9	17.8	12.4	14.9	21.3 ± 0.6	0.08	100.0	77.0
Always-Native (XLM-R-L)	40.4	29.9	43.8	42.0	31.0	25.9	25.2	35.5 ± 0.7	0.41	100.0	61.7
Always-Pivot (FLAN-T5-XL)	39.6	35.7	41.0	39.8	37.4	33.9	27.3	37.9 ± 1.1	1.00	100.0	59.1
Always-Pivot (XLM-R-L)	38.6	34.4	39.6	38.6	36.5	33.0	26.5	36.8 ± 0.8	0.89	100.0	60.3
Merged index (BGE-M3)	42.0	36.9	45.1	43.9	38.1	35.9	28.6	40.3 ± 0.8	0.45	100.0	56.5
Adaptive-RAG‡	44.7	37.6	46.3	46.1	37.2	33.4	29.4	40.9 ± 0.7	0.58	100.0	55.8
MBA-RAG‡	45.1	39.0	48.5	47.4	39.2	35.5	30.6	42.5 ± 0.9	0.52	100.0	54.1
DRAGIN	**46.2**	36.1	42.3	47.7	38.6	31.5	30.2	40.4 ± 1.0	0.56	100.0	57.7
CrossRAG	45.3	37.1	45.2	46.9	36.8	34.1	29.1	40.9 ± 1.1	0.55	100.0	52.3
CORAL	45.9	38.1	45.9	46.7	36.1	33.9	29.7	41.1 ± 1.0	0.56	100.0	53.6
XL-CARR (policy only)	44.8	**41.1**	**49.4**	**48.6**	**42.0**	**38.3**	**32.1**	$\mathbf{44.0 \pm 0.5}^{\ddagger}$	**0.32**	100.0	52.5
Within the formal cascade ($\alpha = 0.30$)											
MBA-RAG‡ + cascade	44.8	38.7	48.2	47.2	38.8	35.1	30.3	42.1	0.73	67.4	19.5
XL-CARR + cascade	45.5	41.7	49.9	48.9	42.5	38.7	32.5	44.5	0.59	70.4	19.7

Table 4. Open-retrieval F1 and cost on XQuAD (no XQuAD data in training) and MLQA (same-language). Notation as in Table 3; full-information router not run on these benchmarks.

	XQuAD								MLQA					
Method	**el**	**hi**	**ro**	**th**	**tr**	**vi**	**F1**	**Cost**	**ar**	**hi**	**vi**	**zh**	**F1**	**Cost**
Closed-book mT5-Large	23.9	19.5	26.8	15.4	22.5	26.7	22.5 ± 0.9	0.08	18.4	20.7	22.9	22.5	21.1 ± 0.4	0.07
Always-Native (XLM-R-L)	37.9	34.8	41.6	30.1	36.9	40.8	37.0 ± 0.8	0.43	32.9	32.1	37.8	36.1	34.7 ± 0.7	0.40
Always-Pivot (FLAN-T5-XL)	39.1	38.0	40.9	34.2	38.6	40.3	38.5 ± 1.1	1.00	32.5	36.6	36.5	33.8	34.9 ± 0.6	1.00

Always-Pivot (XLM-R-L)	37.6	36.8	39.8	32.9	37.4	38.9	37.2 ± 0.9	0.90	31.2	35.3	34.5	32.4	33.4 ± 0.5	0.88
Merged index	39.8	38.1	42.6	35.0	39.6	42.5	39.6 ± 0.8	0.46	33.5	38.7	38.4	37.4	37.0 ± 0.5	0.44
Adaptive-RAG‡	41.7	38.7	44.7	36.2	43.0	44.0	41.4 ± 0.9	0.56	36.5	38.9	41.1	38.8	38.8 ± 0.6	0.54
MBA-RAG‡	44.5	40.7	46.4	36.7	43.4	46.3	43.0 ± 1.0	0.51	39.0	39.8	42.4	39.3	40.1 ± 0.6	0.49
DRAGIN	41.1	38.6	42.1	35.7	38.2	42.9	39.8 ± 1.0	0.41	32.5	39.1	38.8	38.1	37.1 ± 0.5	0.51
CrossRAG	40.9	36.7	42.1	34.9	38.7	42.2	39.3 ± 0.9	0.41	33.6	38.9	38.2	37.4	37.0 ± 0.5	0.49
CORAL	42.2	38.2	42.9	36.1	40.1	42.3	40.3 ± 0.8	0.42	33.8	38.1	38.7	37.2	37.0 ± 0.5	0.48
XL-CARR (policy only)	**45.9**	**42.9**	**47.4**	**39.1**	**45.0**	**47.6**	**44.7 ±0.4**†	**0.32**	**39.9**	**41.7**	**43.6**	**40.7**	**41.5 ±0.4**†	**0.31**
With the conformal cascade ($\alpha = 0.30$)														
MBA-RAG‡ + cascade	44.1	40.5	46.2	36.5	43.1	46.2	42.8	0.72	38.8	39.4	42.2	38.9	39.8	0.68
XL-CARR + cascade	46.2	43.4	47.9	39.5	45.5	48.0	45.1	0.57	40.4	42.0	44.1	41.1	41.9	0.54

Comparison With Recent Adaptive And Cross-Lingual RAG Methods. DRAGIN, CrossRAG and CORAL reach 40.4–41.1 average F1 on TyDiQA, 39.3–40.3 on XQuAD and 37.0–37.1 on MLQA, below both MBA-RAG and XL-CARR on every benchmark, at costs of 0.55–0.56, 0.41–0.42 and 0.48–0.51. DRAGIN obtains the highest Arabic F1 (46.2): for a large native corpus, when to retrieve matters more than where. The margins of XL-CARR over MBA-RAG exceed the seed standard deviations (0.4–1.1 F1) and remain significant under the paired bootstrap, whereas the differences among DRAGIN, CrossRAG and CORAL are within one standard deviation.

5.2 Separating Routing Gains From Verification Gains

Table 5 decomposes the comparison with MBA-RAG, the strongest baseline, into router and cascade effects. Without the cascade, the router alone accounts for the full F1 gain (+1.5, +1.7 and +1.4) and a cost reduction of 0.18–0.20; the Welch intervals over five training seeds per system all exclude zero. The cascade, applied identically, changes F1 by at most ± 0.5 for either router; its effect is on reliability, lowering the emitted-error rate from 52.5–54.1% to 19.5–19.7% on TyDiQA at an additional cost of 0.19–0.27. With the cascade, the router effect on coverage is +3.0, +3.3 and +2.9 points, with 95% intervals over the 100 splits that exclude zero, and the AURC on TyDiQA falls from 0.281 to 0.263 (-6.4%, interval $[-0.027, -0.009]$), because more first-stage answers of XL-CARR pass the same threshold, i.e., from routing rather than verification.

Two qualifications apply. Part of the cost difference reflects the budget itself (with a fixed λ, the cost is 0.36; Section 5.5), and with the cascade XL-CARR costs more (0.59) than MBA-RAG without

it (0.52). Per-language 95% F1 intervals ($\pm$1.2 for Arabic to $\pm$3.9 for Bengali) are on the order of the per-language gains, so the conclusions rest on the benchmark-level gains and on a positive gain in 26 of the 29 test sets.

Table 5. Decomposition of the comparison with MBA-RAG‡ into routing and cascade effects (F1, normalized cost including verification, coverage in %). Brackets: 95% intervals (F1: Welch over five seeds per system; coverage: over 100 calibration–test splits)

	TyDiQA			XQuAD			MLQA		
System / effect	**F1**	**Cost**	**Cov.**	**F1**	**Cost**	**Cov.**	**F1**	**Cost**	**Cov.**
(a) MBA-RAG, policy only	42.5	0.52	100.0	43.0	0.51	100.0	40.1	0.49	100.0
(b) XL-CARR, policy only	44.0	0.32	100.0	44.7	0.32	100.0	41.5	0.31	100.0
(c) MBA-RAG + cascade	42.1	0.73	67.4	42.8	0.72	68.3	39.8	0.68	66.9
(d) XL-CARR + cascade	44.5	0.59	70.4	45.1	0.57	71.6	41.9	0.54	69.8
Routing, no cascade (b)–(a)	+1.5	-0.20	0.0	+1.7	-0.19	0.0	+1.4	-0.18	0.0
95% interval of F1	[0.4, 2.6]			[0.5, 2.9]			[0.6, 2.2]		
Routing, with cascade (d)–(c)	+2.4	-0.14	+3.0	+2.3	-0.15	+3.3	+2.1	-0.14	+2.9
95% interval of coverage	[1.9, 4.1]			[2.1, 4.5]			[2.0, 3.8]		
Cascade, MBA-RAG (c)–(a)	-0.4	+0.21	-32.6	-0.2	+0.21	-31.7	-0.3	+0.19	-33.1
Cascade, XL-CARR (d)–(b)	+0.5	+0.27	-29.6	+0.4	+0.25	-28.4	+0.4	+0.23	-30.2

5.3 Per-Language Risk and Conditional Error

At essentially the same realized risk (0.194, 0.195 and 0.197), Adaptive-RAG, MBA-RAG and XL-CARR answer 65.0%, 67.4% and 70.4% of the TyDiQA questions, with conditional errors of 29.8%, 28.9% and 28.0%; XL-CARR still leaves 29.6% unanswered, and 28.0% of its delivered answers are incorrect.

Per-language calibration is needed (Table 6): pooled thresholds keep the average risk below α, but Swahili (0.312) and Telugu (0.329) exceed it, whereas language-conditional thresholds keep every language below α (at most 0.230), at lower coverage for the least reliable languages. Because the bound is on answering and being wrong, these same languages carry the highest conditional error. For Telugu the Wilson interval (33.7–44.4%) lies entirely above 30% and for Swahili (29.3–41.0%) almost entirely, so their high conditional error is a property of the system, not of sampling noise; for Bengali (56 answered test questions) the interval is too wide to decide. Their risk is spread over fewer answers because their policies start from a lower first-stage F1 (42.0 and 38.3, against 44.8–49.4).

Table 6. Realized risk, coverage and conditional error per language on TyDiQA at $\alpha = 0.30$ (100 splits). Pooled: one threshold pair for all languages; language-conditional: per-language calibration (XL-CARR). Brackets: 95% Wilson intervals for one test portion; values above α in bold

	Pooled			Language-conditional		
Language	$\mathbf{n}_{\text{cal}}$	$\mathbf{n}_{\text{test}}$	**Realized risk**	**Realized risk**	**Coverage (%)**	**Cond. error (%)**
ar	184	737	0.142	0.175 [0.149, 0.204]	74.6 [71.3, 77.6]	23.5 [20.1, 27.2]
bn	23	90	0.201	0.210 [0.139, 0.305]	62.3 [52.0, 71.6]	33.7 [22.7, 46.8]

fi	156	626	0.151	0.185 [0.157, 0.217]	76.8 [73.3, 79.9]	24.1 [20.5, 28.1]
id	113	452	0.160	0.185 [0.152, 0.223]	76.3 [72.2, 80.0]	24.2 [20.0, 29.0]
sw	100	399	**0.312**	0.220 [0.182, 0.263]	63.1 [58.3, 67.7]	34.9 [29.3, 41.0]
te	134	535	**0.329**	0.230 [0.196, 0.268]	59.2 [55.0, 63.3]	38.9 [33.7, 44.4]
All (question-weighted)	710	2,839	0.208	0.197 [0.183, 0.212]	70.4 [68.7, 72.1]	28.0 [26.1, 30.0]

Practical implications. The marginal bound limits how many questions in a language receive an incorrect answer (at most 30 of every 100 Telugu questions in expectation), but a Telugu user who receives an answer faces an error probability of about 0.39. Where users act on each answer, one can (i) set α per language on the validation split to keep the conditional error below 30%, which lowers the coverage of Telugu, Swahili and Bengali to 39.1%, 51.8% and 53.9%; (ii) show $\tilde{\eta}$ or route such answers to human review; or (iii) control the conditional error directly with Learn-then-Test [44].

Effect of the calibration size. Proposition 1 holds for every n, but small calibration sets matter: the term $1/(n+1)$ in Eq. (8) consumes 41.7% of α_2 for Bengali ($n = 23$), against at most 9.9% elsewhere, making its thresholds conservative; the Bengali risk has a split-to-split standard deviation of about 0.043 (0.014 for Arabic); and a guarantee with probability $1-\delta$ for the particular calibration set [45] would need a Hoeffding margin $\sqrt{\ln(1/\delta)/(2n)}$ of 0.224 for Bengali at $\delta = 0.1$, so only Arabic, Finnish and Telugu (0.079–0.093) would retain both stages.

5.4 Indic Languages and Synthetically Romanized Input

On IndicQA, where the controller has seen data only for Hindi, Bengali and Telugu, XL-CARR improves average F1 over MBA-RAG by 2.5 points (36.4 against 33.9; Always-Native 23.6, Always-Pivot 32.0), more than elsewhere, since most Indic corpora are small. The gain is positive in all eleven languages, from 2.2 for Bengali and Marathi to 3.0 for Assamese, and is largest for the smallest corpora (Assamese, Odia, Gujarati, Kannada; 2.6–3.0). Although pivoting beats native retrieval for all eleven languages in isolation, the pivot share stays between 0.43 (Tamil) and 0.65 (Odia) and falls with corpus size (Spearman $r_s = -0.97$) without any language-specific rule. On the IndicXlit-romanized version, Always-Native drops from 23.6 to 14.8 F1, MBA-RAG from 33.9 to 27.1 and XL-CARR from 36.4 to 31.9; without s_3, XL-CARR drops from 36.1 to 28.2, and the policy selects s_3 for 46% of Romanized questions against 7% of native-script ones. Since IndicXlit both creates and inverts these questions, the uroman-plus-noise version is more realistic: Always-Native reaches 12.3, MBA-RAG 24.8 and XL-CARR 28.6 F1 (26.2 without s_3), so the benefit of s_3 shrinks from 3.4 to 2.1 F1. Neither version represents naturally typed, code-mixed text, and Dakshina [46] romanizes sentences, not questions; all statements about Romanized input therefore refer to synthetic data, and the benefit on natural input may be smaller.

5.5 Exploration Strategy and Budgets

Varying only exploration on the TyDiQA stream (last 10,000 rounds; no cascade; same context), stream F1 rises from 46.9 for ε-greedy (as in MBA-RAG; cost 0.53) and 46.6 for Boltzmann exploration, whose regret grows linearly (Fig. 2a), to 47.4 for LinUCB and 48.0–48.2 for NeuralUCB and NeuralTS, at about three times the GPU hours of neural-linear variants. Neural-linear Thompson sampling reaches 48.5 at a cost of 0.36 with a fixed λ, and the primal–dual XL-CARR 48.9 at 0.32 with a regret of 1.40×10^3 (against 1.65×10^3 and 3.68×10^3 for ε-greedy), the only strategy without measured cost violation;

its running mean cost settles at the budget of 0.32 while $\bar{\mu}_C$ stabilizes at about 0.9 (Fig. 2b), as expected for a binding constraint.

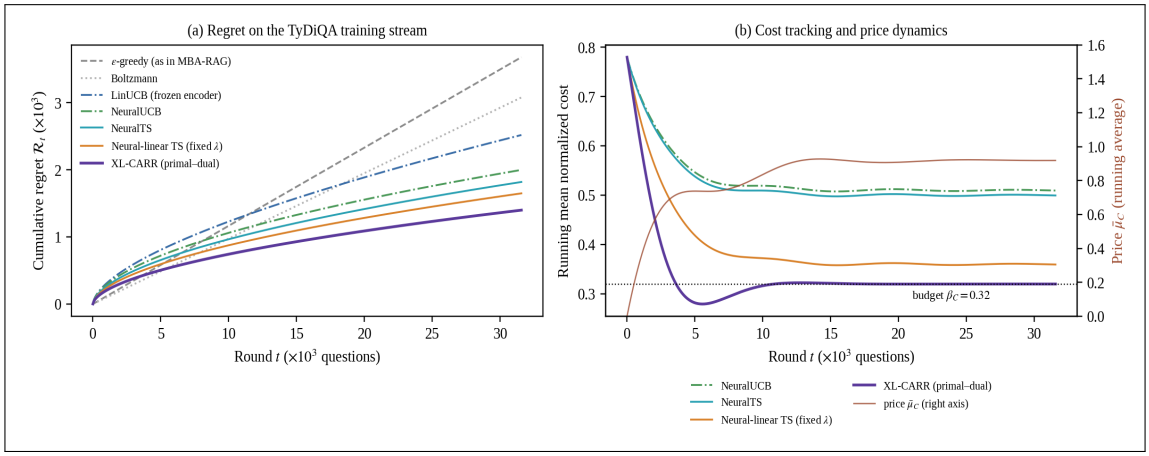

Figure 2. Learning on the TyDiQA stream: (a) cumulative regret (mean over 5 seeds); (b) running mean cost and average price $\bar{\mu}_C$ (right axis).

Sensitivity to the error budget. With $\beta_H = 0.50$, $\bar{\mu}_H$ stays at 0.04; tightening it to 0.47 and 0.44 raises $\bar{\mu}_H$ to 0.17 and 0.58, lowers the stream error to 46.9% and 44.0% at a cost of 0.32 and raises test coverage from 70.4% to 70.9% and 72.1% (test F1 44.5, 44.5 and 44.3). M, the posterior scale and τ_{QE} were selected offline with the DR estimator over the grids listed in Appendix B.

5.6 Ablation Study

In Table 7, evidence access has the largest effect: removing s_2 costs 3.2 F1 on TyDiQA and 2.5 on average, and the tools 1.0 (0.7 from t_1); tools serve about one fifth of TyDiQA questions. Among context features, the retrievability indicators (-1.1) matter more than the typological vector and the cluster adapters (-0.6, -0.5), consistent with corpus size explaining more of the gains than language family. Learning components mainly affect cost (0.63 without the warm start, 0.65 with a fixed λ). Verification components act on reliability rather than F1: an NLI-only score costs 4.0 points of coverage at the same emitted error, removing the CometKiwi gate 1.8 points (and 1.0 F1 on XQuAD, where Thai and Greek translations are noisier), and removing the cascade raises the emitted-error rate from 19.7% to 52.5%.

Removing a source measures the value of that evidence, not of the routing: Adaptive-RAG, MBA-RAG and XL-CARR share the same 18 actions yet range from 40.9 to 44.0 F1 on TyDiQA, so how English evidence is selected matters as much as its availability. The small effect of removing s_3 reflects the near absence of Romanized questions in the three benchmarks; since components interact, the drops are not additive.

Table 7. Ablation study (average F1; cascade enabled unless removed). Cost, Cov. and Err. refer to TyDiQA; Δ: mean F1 change over the three benchmarks.

Variant	TyDiQA	XQuAD	MLQA	Cost↓	Cov.	Err.↓	Δ
XL-CARR (full)	44.5	45.1	41.9	0.59	70.4	19.7	0.0
Action space							
– pivot source s_2	41.3	42.7	39.9	0.50	66.0	19.8	-2.5
– transliterated source s_3	44.2	44.8	41.7	0.59	70.2	19.7	-0.3
– all tools (t_1–t_3)	43.3	44.2	41.0	0.55	68.8	19.9	-1.0
– English reader t_1 only	43.7	44.5	41.3	0.56	69.4	19.8	-0.7

– calculator/date tool t_2 only	44.3	45.0	41.8	0.59	70.3	19.7	-0.1
Context							
– typological vector u_l	43.9	44.5	41.4	0.60	69.8	19.8	-0.6
– retrievability indicators $q(x)$	43.2	44.1	40.8	0.62	69.2	19.8	-1.1
– cluster LoRA (shared adapter)	43.9	44.6	41.4	0.60	69.9	19.8	-0.5
Verification							
η with NLI only	44.1	44.8	41.6	0.55	66.4	19.6	-0.3
– CometKiwi gate	44.0	44.1	41.5	0.60	68.6	19.8	-0.6
– conformal cascade (policy only)	44.0	44.7	41.5	0.32	100.0	52.5	-0.4
Learning							
– probe set and DR warm start	43.4	44.2	40.9	0.63	69.0	19.8	-1.0
budgets replaced by fixed-λ reward	43.9	44.6	41.4	0.65	70.1	19.7	-0.5

5.7 Error Detection, Cascade Calibration and Sensitivity to the Risk Level

The AUROC of emitted-error detection for mBERT/XLM-R-Large/mT5 is 0.68/0.70/0.66 for span or token entropy, 0.74/0.76/0.73 for NLI, 0.72/0.75/0.79 for semantic dispersion or entropy and 0.61/0.62/0.60 for CometKiwi. The combination η reaches 0.80/0.82/0.84, 0.05–0.06 above the best single signal.

Table 8 separates the two stages and treats a rejected first-stage answer as a predicted error. Realized risk stays below the target at every level, at about 0.65α (Fig. 3a). At $\alpha = 0.30$, stage-1 acceptances (57.5% of questions) carry a risk of 0.158 against $\alpha_1 = 0.20$, with a conditional error of 27.5%, whereas stage-2 acceptances (12.9%) carry 0.039 against $\alpha_2 = 0.10$, with 30.2%. The larger stage-2 slack arises because $\widehat{\tau}_2$ is calibrated over all n examples although only 42.5% of questions reach it (Remark 1). 86.4% of rejected answers are incorrect and 69.9% of incorrect answers are rejected, while 12.2% of correct answers are rejected unnecessarily. The score affects coverage, not validity (Fig. 3b): η answers 70.4%, against 59.0% for NLI and 64.4% for semantic dispersion alone.

The risk level is the dominant parameter: from $\alpha = 0.15$ to 0.40, coverage rises from 49.5% to 78.4% and F1 from 34.4 to 46.0, while the conditional error rises from 19.8% to 33.7%, so α is a deployment decision rather than a tuning choice.

Table 8. Stage-wise cascade and first-stage error detection on TyDiQA (100 splits). Shares in % of test questions (coverage = S1 + S2). Cond. err.: error among all accepted answers; Prec./Rec.: precision and recall of rejection for incorrect first-stage answers; False rej.: share of correct first-stage answers rejected. Without the cascade, Err. is 52.5% and F1 44.0.

	Share (%)			**Realized risk**					**First-stage detection (%)**		
Target α	**S1**	**S2**	**Abstain**	**S1**	**S2**	**Total**	**Cond. err. (%)**	**F1 (all)**	**Prec.**	**Rec.**	**False rej.**
0.15	41.5	8.0	50.5	0.080	0.018	0.098	19.8	34.4	76.1	84.8	29.5
0.20	47.8	10.2	42.0	0.106	0.025	0.131	22.6	39.1	80.3	79.8	21.7
0.25	53.0	11.6	35.4	0.131	0.032	0.163	25.2	42.2	83.8	75.0	16.0
0.30	**57.5**	**12.9**	**29.6**	**0.158**	**0.039**	**0.197**	**28.0**	**44.5**	**86.4**	**69.9**	**12.2**
0.40	67.8	10.6	21.6	0.226	0.038	0.264	33.7	46.0	92.9	57.0	4.8

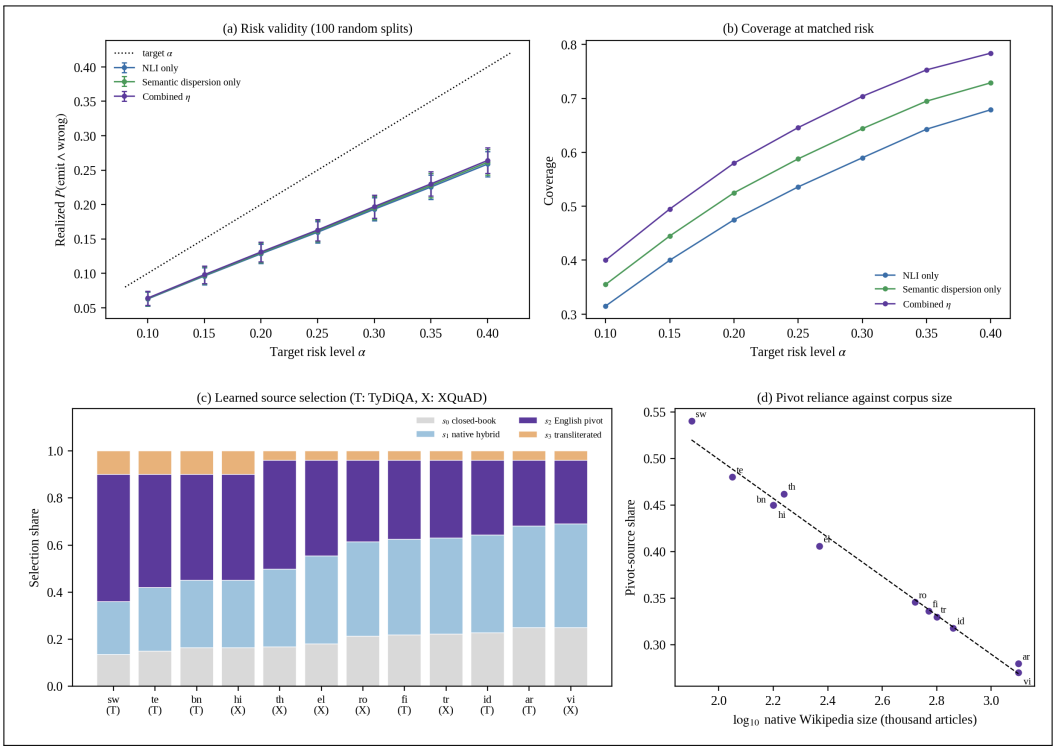

Figure 3. (a) Realized risk against α (100 splits); (b) coverage at matched risk; (c) learned source shares per language; (d) pivot share against log native Wikipedia size.

5.8 Routing Behavior, Typological Diversity and Transfer to Unseen Languages

The pivot share falls almost linearly with log corpus size, from 0.54 for Swahili to 0.27 for Vietnamese (Fig. 3c,d). For the 21.4% of TyDiQA questions whose answer entity lacks an English Wikipedia article, Always-Pivot trails Always-Native (27.8 against 38.7 F1); XL-CARR obtains 39.3 on them by pivoting for only 19%. In MLQA G-XLT, gains over MBA-RAG are larger when question and context languages differ (+2.9) than in the same-language setting (+1.4), in line with Park and Lee [9].

Typological Diversity. Over 23 languages from eleven families and 16 scripts, the mean policy-only gain over MBA-RAG is positive in eleven of twelve family groups, largest for Niger–Congo (+2.8), Dravidian (+2.5) and Indo-Aryan and Kra–Dai (+2.4), and smallest for Afro-Asiatic (+0.3) and Slavic/Koreanic (-0.4), whose native corpora are large; within Indo-Aryan the gain decreases as the corpus grows, so corpus size matters more than family.

Leave-One-Language-Out Transfer. In the leave-one-language-out (LOLO) setting (Table 9), a held-out language loses its probes and stream and relies on u_l, the indicators and its cluster adapter. The policy retains about 97% of the in-language F1 (42.6 against 44.0), loses 1.1–2.6 more points without u_l, and stays ahead of the better static pipeline and of LOLO MBA-RAG in every language, narrowly for Arabic (0.3 F1). 2,000 labeled bandit rounds in the held-out language lift it to 43.5, so further adaptation, like calibration, needs labeled data.

Influence of the Training-Language Distribution. The bandit stream contains only the six TyDiQA languages and is dominated by Arabic (Table 9). Three observations indicate that the improvements are not an artefact of this distribution. First, the F1 gain is negatively rank-correlated with the training share (Spearman $r_s = -0.81$, $p = 0.05$): Arabic (38.9% of the stream) gains nothing, whereas Bengali and Swahili (6.3% and 7.2%) gain 2.1 and 2.8, although share and corpus size are related. Second, on IndicQA, the eight languages without any training data gain 2.6 F1 on average,

against 2.3 for Hindi, Bengali and Telugu, and across the eleven languages the gain decreases with native corpus size ($r_s = -0.84$, $p = 0.001$). Third, Finnish, whose cluster {el, fi, ro} contains no other TyDiQA language, retains the most (98.0%) in the LOLO setting, so transfer does not require a typologically similar language in the stream.

Table 9. Leave-One-Language-Out (LOLO) F1 on TyDiQA (no cascade) and the training-language distribution. Share: fraction of the 38,070 TyDiQA training questions; Gain: in-language policy-only F1 gain over MBA-RAG‡; Static: better static pipeline; Offline: full-information router trained on the other languages; +Online: after 2,000 labeled bandit rounds in the held-out language; Ret.: retention, LOLO F1 as a share of in-language F1. Last row: total share and averages of the other columns.

Held-out	Share (%)	Gain	In-lang	LOLO	LOLO $-u_t$	LOLO MBA	Static	Offline	+Online	Ret. (%)
ar	38.9	-0.3	44.8	43.6	42.4	43.3	40.4	43.9	44.3	97.3
fi	18.0	+0.9	49.4	48.4	47.3	46.9	43.8	48.6	49.0	98.0
id	15.0	+1.2	48.6	47.6	46.4	45.7	42.0	47.9	48.2	97.9
te	14.6	+2.8	38.3	36.5	33.9	33.4	33.9	36.8	37.6	95.3
sw	7.2	+2.8	42.0	40.2	37.7	37.2	37.4	40.6	41.4	95.7
bn	6.3	+2.1	41.1	39.5	37.0	37.1	35.7	39.8	40.6	96.1
All / Avg.	100.0	+1.5	44.0	42.6	40.8	40.6	38.9	42.9	43.5	96.8

5.9 Robustness, Computational Cost and Limitations

Surface-Form Errors and Automated Judges. Entity-normalized F1 (both sides romanized with uroman) raises Always-Pivot by 2.1, Always-Native by 0.3, MBA-RAG to 43.6 and XL-CARR to 45.2. Qwen2.5-72B-Instruct, with Llama-3.3-70B-Instruct as a second judge, assessed 600 sampled answered TyDiQA test questions for each of MBA-RAG and XL-CARR with and without the cascade, deciding whether the evidence supports each answer and whether F1 errors are equivalent to the reference. It judges 9.6% and 10.8% of the F1 errors of MBA-RAG and XL-CARR equivalent to the reference; the cascade raises the share judged supported from 74.8% to 83.6%. The judges agree on 87.9% of decisions ($\kappa = 0.72$; 0.58 for Telugu, 0.61 for Swahili). Against blind native-speaker labels, the primary judge agrees on 64 of 70 Bengali (91.4%, Wilson 82.5–96.0%) and 44 of 50 Telugu answers (88.0%, 76.2–94.4%); as these checks are small and exclude Swahili, the main comparisons rest on EM and F1. Native dumps from December 2018 keep the margin over MBA-RAG at 1.5 F1.

Latency, Memory and Throughput. Costs include probes, translation, verification and escalation and are mean latencies (one A100, batch size 1) relative to Always-Pivot (270 ms, 3.7 questions/s). MBA-RAG needs 140 ms (7.1 q/s) and 197 ms with the cascade; XL-CARR needs 86 ms (11.6 q/s) for the policy alone and 159 ms (6.3 q/s) with the cascade, of which 22.5 ms go to the controller and probes, 63.5 ms to the selected action, 19.6 ms to stage-1 verification, 43.2 ms to stage-2 execution and 10.3 ms to stage-2 verification. Query translation (86 ms) and the English reader (93 ms) dominate pivot actions; native-to-pivot escalations (24% of the 39.4% of questions that run stage 2, about 228 ms each) drive the 95th-percentile latency of 402 ms, so tighter tail targets require the policy alone. The models total about 10.1×10^9 parameters (20 GB in 16-bit precision) and fit on one 80-GB A100; the English dense index (21.0 million BGE-M3 vectors) needs 43 GB in 16-bit precision. Controller cost grows linearly in K, not with corpus size; a new language needs its index, typological vector and calibration set, not reader retraining. Throughput under concurrent load and peak memory were not measured.

Scope and Limitations. Unlike iterative pipelines such as DRAGIN or CORAL, XL-CARR

decides once before retrieval, with at most one escalation, so its cost can be budgeted in advance but it cannot react to evidence gaps found during reasoning; an iterative pipeline could be added as one more action. Limitations: learning and calibration require labeled questions; regret and violation are measured, not guaranteed; the bound is marginal; native-speaker checks cover two languages; the Romanized evaluation is synthetic; the stream is Arabic-dominated; and DRAGIN, CrossRAG and CORAL were compared only without the cascade.

6. Conclusion

XL-CARR extends adaptive retrieval routing to budgeted, cross-lingual and reliability-controlled decisions for low-resource languages. A constrained bandit chooses among 18 source–tool–reader actions, and a per-language two-stage conformal cascade bounds the probability of emitting an incorrect answer in finite samples. Across 23 languages and 29 test sets, the policy alone improves average F1 over an adapted MBA-RAG with the same sources and readers by 1.4–1.7 points on TyDiQA-GoldP, XQuAD and MLQA and by 2.5 points on IndicQA at a 37–38% lower cost. These gains come from routing rather than the cascade. With the same cascade, it reduces AURC by 6.4% and answers 2.9–3.3 more questions per hundred at the same risk. The gains concentrate in languages with small native corpora. The error rate among answered questions remains 23.5–38.9% across TyDiQA languages, and lowering it costs coverage, most notably in the least-resourced languages. Future work will use proxy feedback, control per-language conditional risk, and evaluate on naturally typed Romanized questions.

References

[1] Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler et al. ”Retrieval-augmented generation for knowledge-intensive nlp tasks.” Advances in neural information processing systems 33 (2020): 9459-9474.

[2] Karpukhin, Vladimir, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. ”Dense passage retrieval for open-domain question answering.” In Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp. 6769-6781. 2020.

[3] Pokhrel, Sangita, Bina KC, and Prashant Bikram Shah. ”A practical application of retrieval-augmented generation for website-based chatbots: combining web scraping, vectorization, and semantic search.” Journal of Trends in Computer Science and Smart Technology 6, no. 4 (2025): 424-442.

[4] Su, Weihang, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. ”Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models.” In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 12991-13013. 2024.

[5] Jeong, Soyeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. ”Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity.” In Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers), pp. 7036-7050. 2024.

[6] Tang, Xiaqiang, Qiang Gao, Jian Li, Nan Du, Qi Li, and Sihong Xie. ”Mba-rag: a bandit approach for adaptive retrieval-augmented generation through question complexity.” In Proceedings of the 31st International Conference on Computational Linguistics, pp. 3248-3254. 2025.

[7] Joshi, Pratik, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. ”The state and fate of linguistic diversity and inclusion in the NLP world.” In Proceedings of the 58th annual

meeting of the association for computational linguistics, pp. 6282-6293. 2020.

[8] Asai, Akari, Jungo Kasai, Jonathan H. Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. "XOR QA: Cross-lingual open-retrieval question answering." In Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 547-564. 2021.

[9] Park, Jeonghyun, and Hwanhee Lee. "Investigating language preference of multilingual rag systems." In Findings of the Association for Computational Linguistics: ACL 2025, pp. 5647-5675. 2025.

[10] Ranaldi, Leonardo, Barry Haddow, and Alexandra Birch. "Multilingual retrieval-augmented generation for knowledge-intensive question answering task." In Findings of the association for computational linguistics: EACL 2026, pp. 697-716. 2026.

[11] Koehn, Philipp, and Rebecca Knowles. "Six challenges for neural machine translation." In Proceedings of the first workshop on neural machine translation, pp. 28-39. 2017.

[12] Costa-Jussà, Marta R., James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi et al. "No language left behind: Scaling human-centered machine translation." arXiv preprint arXiv:2207.04672 (2022).

[13] Artetxe, Mikel, Gorka Labaka, and Eneko Agirre. "Translation artifacts in cross-lingual transfer learning." In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 7674-7684. 2020.

[14] Pandya, Hariom, Jatayu Baxi, and Apurva Mehta. "Late-interaction Answer-aligned Retrieval for Multilingual RAG-based Question Answering." International Journal of Intelligent Engineering & Systems 19, no. 6 (2026): 1040.

[15] Pfeiffer, Jonas, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. "Mad-x: An adapter-based framework for multi-task cross-lingual transfer." In Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp. 7654-7673. 2020.

[16] Doddapaneni, Sumanth, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. "Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages." In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 12402-12426. 2023.

[17] Agrawal, Shipra, and Nikhil Devanur. "Linear contextual bandits with knapsacks." Advances in neural information processing systems 29 (2016). 3450–3458.

[18] Angelopoulos, Anastasios, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. "Conformal risk control." In International conference on learning representations, vol. 2024, pp. 55198-55218. 2024.

[19] Kuhn, Lorenz, Yarin Gal, and Sebastian Farquhar. "Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation." arXiv preprint arXiv:2302.09664 (2023).

[20] Rei, Ricardo, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C. Farinha, Christine Maroti, José GC De Souza et al. "CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task." In Proceedings of the Seventh Conference on Machine Translation (WMT), pp. 634-645. 2022.

[21] Clark, Jonathan H., Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. ”Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages.” Transactions of the Association for Computational Linguistics 8 (2020): 454-470.

[22] Artetxe, Mikel, Sebastian Ruder, and Dani Yogatama. ”On the cross-lingual transferability of monolingual representations.” In Proceedings of the 58th annual meeting of the association for computational linguistics, pp. 4623-4637. 2020.

[23] Lewis, Patrick, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. ”MLQA: Evaluating cross-lingual extractive question answering.” In Proceedings of the 58th annual meeting of the association for computational linguistics, pp. 7315-7330. 2020.

[24] Li, Lihong, Wei Chu, John Langford, and Robert E. Schapire. ”A contextual-bandit approach to personalized news article recommendation.” In Proceedings of the 19th international conference on World wide web, pp. 661-670. 2010.

[25] Zhou, Dongruo, Lihong Li, and Quanquan Gu. ”Neural contextual bandits with ucb-based exploration.” In International conference on machine learning, pp. 11492-11502. PMLR, 2020.

[26] Zhang, Weitong, Dongruo Zhou, Lihong Li, and Quanquan Gu. ”Neural thompson sampling.” arXiv preprint arXiv:2010.00827 (2020).

[27] Riquelme, Carlos, George Tucker, and Jasper Snoek. ”Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling.” arXiv preprint arXiv:1802.09127 (2018).

[28] Badanidiyuru, Ashwinkumar, Robert Kleinberg, and Aleksandrs Slivkins. ”Bandits with knapsacks.” Journal of the ACM (JACM) 65, no. 3 (2018): 1-55.

[29] Ferreira, Kris Johnson, David Simchi-Levi, and He Wang. ”Online network revenue management using thompson sampling.” Operations research 66, no. 6 (2018): 1586-1602.

[30] Panda, Pranoy, Raghav Magazine, Chaitanya Devaguptapu, Sho Takemori, and Vishal Sharma. ”Adaptive LLM Routing under Budget Constraints.” In EMNLP (Findings), pp. 23934-23949. 2025.

[31] Lee, Nayeon, Jiwoo Song, and Byeongcheol Kang. ”CORAL: Adaptive Retrieval Loop for Culturally-Aligned Multilingual RAG.” In Findings of the Association for Computational Linguistics: ACL 2026, pp. 27215-27237. 2026.

[32] Campos, Margarida, António Farinhas, Chrysoula Zerva, Mário AT Figueiredo, and André FT Martins. ”Conformal prediction for natural language processing: A survey.” Transactions of the Association for Computational Linguistics 12 (2024): 1497-1516.

[33] Kumar, Bhawesh, 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).

[34] Mohri, C., and T. Hashimoto. “Language models with conformal factuality guarantees.” In Proceedings of the 41st International Conference on Machine Learning, vol. 235 of Proceedings of Machine Learning Research. PMLR, 2024.

[35] Quach, Victor, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi Jaakkola, and Regina Barzilay. "Conformal language modeling." In International Conference on Learning Representations, vol. 2024, pp. 11654-11681. 2024.

[36] Li, Shuo, Sangdon Park, Insup Lee, and Osbert Bastani. "TRAQ: Trustworthy retrieval augmented question answering via conformal prediction." In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 3799-3821. 2024.

[37] Fisch, A., T. Schuster, T. Jaakkola, and R. Barzilay. "Efficient conformal prediction via cascaded inference with expanded admission." In Proceedings of the Ninth International Conference on Learning Representations (ICLR 2021). 2021.

[38] Vovk, V., A. Gammerman, and G. Shafer. Algorithmic learning in a random world. Springer, 2005.

[39] Chen, Jianlyu, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. "M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation." In Findings of the association for computational linguistics: ACL 2024, pp. 2318-2335. 2024.

[40] Gala, J., P. A. Chitale, R. AK, V. Gumma, S. Doddapaneni, A. Kumar, J. Nawale, A. Sujatha, R. Puduppully, V. Raghavan, P. Kumar, M. M. Khapra, R. Dabre, and A. Kunchukuttan. "IndicTrans2: Towards high-quality and accessible machine translation models for all 22 scheduled Indian languages." Transactions on Machine Learning Research (2023).

[41] Madhani, Y., S. Parthan, P. Bedekar, G. NC, R. Khapra, A. Kunchukuttan, P. Kumar, and M. M. Khapra. "Aksharantar: Open Indic-language transliteration datasets and models for the next billion users." In Findings of the Association for Computational Linguistics: EMNLP 2023, 40–57. Association for Computational Linguistics, 2023.

[42] Hermjakob, U., J. May, and K. Knight. "Out-of-the-box universal romanization tool uroman." In Proceedings of ACL 2018, System Demonstrations, 13–18. Association for Computational Linguistics, 2018.

[43] Wilson, E. B. "Probable inference, the law of succession, and statistical inference." Journal of the American Statistical Association 22, no. 158 (1927): 209–212.

[44] Angelopoulos, A. N., S. Bates, E. J. Candès, M. I. Jordan, and L. Lei. "Learn then test: Calibrating predictive algorithms to achieve risk control."arXiv preprint arXiv:2110.01052 (2021).

[45] Bates, S., A. Angelopoulos, L. Lei, J. Malik, and M. I. Jordan. "Distribution-free, risk-controlling prediction sets."Journal of the ACM 68, no. 6 (2021): 1–34.

[46] Roark, B., L. Wolf-Sonkin, C. Kirov, S. J. Mielke, C. Johny, I. Demirsahin, and K. Hall. "Processing South Asian languages written in the Latin script: The Dakshina dataset." In of the 12th Language Resources and Evaluation Conference, 2413–2423. European Language Resources Association, 2020.

Appendix:

Appendix A. Proof of the Finite-Sample Guarantee

Fix a language ℓ; probabilities are over the random calibration–test split of ℓ and the sampling noise of the generative reader. For every question m, let $Z_m = \left(\tilde{\eta}_m^{(1)}, H_m^{(1)}, \tilde{\eta}_m^{(2)}, H_m^{(2)}\right)$, where the second pair belongs to the escalation target of the first-stage action ($\tilde{\eta}_m^{(2)} = +\infty$, $H_m^{(2)} = 0$ if none exists); $m = 1, \ldots, n$ are calibration questions and $m = n+1$ the test question.

Exchangeability. The policy, escalation rule, QE gate, calibrator of Eq. (5) and all readers are fixed before calibration on data disjoint from the evaluation sets (Table 2), and α is selected on the validation split. Hence $Z_m = f(x_m, y_m, U_m)$ for one fixed function f, where U_m is the sampling noise of question m, drawn independently. The calibration and test portions come from a uniformly random permutation of the evaluation set of ℓ, so $(Z_1, \ldots, Z_{n+1})$ are exchangeable; no independence between the two stages is required.

Lemma 1 (Conformal Risk Control for One Stage; cf. [18]). Let $(Z_1, \ldots, Z_{n+1})$ be exchangeable and $L_m(\tau) = H_m \mathbf{1}[\tilde{\eta}_m \le \tau] \in [0,1]$, non-decreasing in τ. Let $\hat{\tau}$ be defined by Eq. (8) at level α_j, with $\hat{\tau} = -\infty$ if no candidate qualifies. Then $\mathbb{E}\left[L_{n+1}(\hat{\tau})\right] \le \alpha_j$.

Proof. Define the calibration bound $G(\tau) = \frac{1}{n+1}\left(\sum_{m=1}^{n} L_m(\tau) + 1\right)$ and the full-sample risk $F(\tau) = \frac{1}{n+1}\sum_{m=1}^{n+1} L_m(\tau)$. Since $L_{n+1}(\tau) \le 1$, $F(\tau) \le G(\tau)$. Let $\Lambda^* = \{-\infty\} \cup \{\tilde{\eta}_m\}_{m=1}^{n+1}$ and $\tau^* = \max\{\tau \in \Lambda^* : F(\tau) \le \alpha_j\}$, which exists because $F(-\infty) = 0$. If $\hat{\tau} = -\infty$, then $L_{n+1}(\hat{\tau}) = 0$. Otherwise $\hat{\tau} \in \Lambda_j \subset \Lambda^*$ and $F(\hat{\tau}) \le G(\hat{\tau}) \le \alpha_j$, so $\hat{\tau} \le \tau^*$, and by monotonicity $L_{n+1}(\hat{\tau}) \le L_{n+1}(\tau^*)$ in both cases. The threshold τ^* is a symmetric function of $(Z_1, \ldots, Z_{n+1})$, so the losses $L_1(\tau^*), \ldots, L_{n+1}(\tau^*)$ are exchangeable and

$$\mathbb{E}\left[L_{n+1}(\hat{\tau})\right] \le \mathbb{E}\left[L_{n+1}(\tau^*)\right] = \mathbb{E}\left[\tfrac{1}{n+1}\textstyle\sum_{m=1}^{n+1} L_m(\tau^*)\right] = \mathbb{E}\left[F(\tau^*)\right] \le \alpha_j.$$

Proof of Proposition 1. For the test question, the cascade emits the first-stage answer if $\tilde{\eta}^{(1)} \le \hat{\tau}_1$ and the escalated answer if $\tilde{\eta}^{(1)} > \hat{\tau}_1$ and $\tilde{\eta}^{(2)} \le \hat{\tau}_2$; otherwise it abstains, which is never an emitted error. The two emission events are disjoint, so the emitted-error indicator is

$$L = H^{(1)}\mathbf{1}\left[\tilde{\eta}^{(1)} \le \hat{\tau}_1\right] + H^{(2)}\mathbf{1}\left[\tilde{\eta}^{(1)} > \hat{\tau}_1\right]\mathbf{1}\left[\tilde{\eta}^{(2)} \le \hat{\tau}_2\right] \le L^{(1)}(\hat{\tau}_1) + L^{(2)}(\hat{\tau}_2),$$

where the inequality drops the factor $\mathbf{1}\left[\tilde{\eta}^{(1)} > \hat{\tau}_1\right] \le 1$. Both thresholds are computed from the same n calibration questions by Eq. (8), applied to $L^{(1)}$ at level α_1 and to $L^{(2)}$ at level α_2. Lemma 1 applies to each stage separately, since each $L_m^{(j)}$ depends only on Z_m, which gives $\mathbb{E}\left[L^{(j)}(\hat{\tau}_j)\right] \le \alpha_j$. By linearity of expectation, which needs no independence between $\hat{\tau}_1$ and $\hat{\tau}_2$, $\Pr(\text{emit} \wedge \text{incorrect} \mid l) = \mathbb{E}[L] \le \alpha_1 + \alpha_2 = \alpha$. As only the calibration questions of ℓ are used, the bound holds within every language (Mondrian calibration).

Remark 1 (Calibrating stage 2 on all questions). Calibrating $\hat{\tau}_2$ only on questions with $\tilde{\eta}^{(1)} > \hat{\tau}_1$ would use a subset selected by a data-dependent threshold, breaking exchangeability with an escalated test question. Calibrating $L^{(2)}$ over all n questions keeps it; since $L^{(2)}$ upper-bounds the stage-2 contribution to L, part of α_2 is spent on questions that never reach stage 2, explaining the low realized stage-2 risk (0.039 against 0.10; Table 8).

Remark 2 (Tightness, ties and training-conditional validity).The per-stage bound is nearly tight for large n with distinct scores [18]; ties make it conservative, never invalid. A guarantee with probability $1-\delta$ over the calibration draw follows from risk-controlling prediction sets [45] by replacing the left-hand side of Eq. (8) with an upper confidence bound (Section 5.3). α and (α_1, α_2) are fixed on the validation

split, hence constants with respect to $(Z_1, \ldots, Z_{n+1})$; choosing them on calibration or test data would invalidate the guarantee.

Appendix B. Reproducibility Details

Preprocessing and Indices. Text is normalized to Unicode NFC. Native Wikipedia dumps of 1 March 2026 and the English DPR split of December 2018 (21,015,324 passages) are split into 100-word passages and indexed with BM25 and 1,024-dimensional BGE-M3 embeddings; the same indices are shared by all systems. Evaluation uses the MLQA answer normalization. Table A1 lists all hyper-parameters.

Table A1. Hyper-parameters of XL-CARR.

Component	**Setting**
Retrieval	100-word passages; BM25 $k_1 = 0.9$, $b = 0.4$; BGE-M3 dense retrieval; RRF with $k_{\text{RRF}} = 60$; top-5 passages read
Translation	NLLB-200-distilled-1.3B (non-Indic) and IndicTrans2 (Indic), beam 4; CometKiwi-22 QE gate, $\tau_{\text{QE}} \in \{0.5, 0.6, 0.7\}$ selected offline; entity-preserving tag masking for back-translation
Transliteration	IndicXlit, top-3 variants for query expansion
Readers	mBERT-base and XLM-R-Large (maximum length 384, stride 128), mT5-Large (at most 32 output tokens), fine-tuned on SQuAD and then on each language's TyDiQA training split; FLAN-T5-XL (t_1) fine-tuned on SQuAD v1.1, Natural Questions and TriviaQA
Controller encoder	XLM-R-base with LoRA (rank 8, scaling 16, query and value projections, learning rate 5×10^{-5}), one adapter per cluster (k-means, $k = 4$, on URIEL vectors); mean pooling, $h(x) \in \mathbb{R}^{768}$
Context	$u_l \in \mathbb{R}^{32}$ (PCA of URIEL syntax, phonology and inventory features over 24 languages); retrievability indicators of Section 3.2; $\log \lvert\mathcal{C}_l\rvert$
Network and heads	Action embedding size 64; two-layer GELU network (1,024, 128), dropout 0.1; three Bayesian linear heads with one shared precision matrix; prior precision 1.0; posterior scale $\in \{0.1, 0.25, 0.5, 1\}$ selected offline
Learning	$M = 10$; $\rho = M/\sqrt{T} \approx 0.056$ with $T = 31,563$; $\mu_C = \mu_H = 0$ at initialization; refresh every $T_r = 2,000$ rounds on a buffer of 5×10^4 for 3 epochs; multipliers frozen at their mean over the last 5,000 rounds; $N_p = 500$ probes per language; 5×10^3 uniform-exploration rounds; doubly-robust offline selection
Budgets	$\beta_C^{\text{dep}} = 0.59$, $\hat{c}_{\text{ver}} = 0.27$, $\beta_C = 0.32$; $\beta_H = 0.50$
Error score	Five candidates; temperature 0.7 for mT5; entailment threshold 0.5; XLM-R-Large fine-tuned on XNLI; logistic calibrator fitted on the probe set; entropy gate with bounds $\kappa_{\text{lo}}, \kappa_{\text{hi}}$
Cascade	$J = 2$ stages; $\alpha = 0.30$, $(\alpha_1, \alpha_2) = (0.20, 0.10)$; per-language calibration on 20% of each evaluation set; 100 splits
Evaluation	Error if F1 < 0.5; five seeds; paired bootstrap (10,000 resamples) with Holm–Bonferroni correction; latency on one A100, batch size 1