AI Engineering
https://www.sciltp.com/journals/aieng

Article

Energy-Aware Edge-Cloud Collaboration for Learning Varieties of English: Cache-Assisted Inference and Evidence-Grounded Feedback

Cunqian You [1,2,*], Shiwei Zhang [1], Lu Qi [1], Miao Wei [2], Xiaojun Wang [3,*] and Huijuan Lu [4,5,*]

[1] College of Modern Science and Technology, China Jiliang University, Yiwu 322002, China
[2] College of Management Science and Engineering, China Jiliang University, Hangzhou 310018, China
[3] School of Economics, China Jiliang University, Hangzhou 310018, China
[4] School of Information, Zhejiang Guangsha Vocational and Technical University of Construction, Dongyang 322100, China
[5] School of Information Engineering, China Jiliang University, Hangzhou 310018, China
* Correspondence: ycq@cjlu.edu.cn (C.Y.); wxjun@cjlu.edu.cn (X.W.); hjlu@cjlu.edu.cn (H.L.)

How To Cite: You, C.; Zhang, S.; Qi, L.; et al. Energy-Aware Edge-Cloud Collaboration for Learning Varieties of English: Cache-Assisted Inference and Evidence-Grounded Feedback. *AI Engineering* **2026**, *2*(1), 6. https://doi.org/10.53941/aieng.2026.100006

Received: 6 March 2026
Revised: 15 May 2026
Accepted: 30 May 2026
Published: 24 June 2026

Abstract: Speech-first English tutors are increasingly expected to correct pronunciation, explain regional usage, and keep the interaction fast enough for rehearsal. These requirements make energy use, latency, and feedback fidelity part of the same design problem. We revise the edge-cloud tutor as a formally specified, edge-first system for five English varieties: American, British, Indian, Australian, and Canadian English. The system selects among semantic-cache reuse, local retrieval-augmented generation, local generation with pronunciation scoring, and cloud fallback by maximizing a quality-energy-latency utility under explicit constraints. Its semantic cache stores only validated, evidence-versioned feedback; its inference cache reuses hot weights and short-context key-value states; and its feedback composer is restricted to evidence selected from a curated variety knowledge base. A controlled prototype evaluation over matched five-minute tutoring sessions shows that the hybrid cached design reduces total session energy by 41.0% and median latency by 44.3% relative to a cloud-only baseline, while preserving near-cloud variety fidelity (94.2% versus 95.0%). Compared with the same hybrid pipeline without caching, it reduces energy by 16.3% and median latency by 16.6%. Human-rated feedback evaluation further shows higher evidence support and lower answer leakage than a cloud NLP feedback baseline. The results do not claim that edge-first tutoring is always best: edge-only remains the lowest-energy mode, but loses fidelity and diagnostic depth. The main contribution is a transparent operating point for real-time English-variety learning where energy, latency, privacy, and pedagogical quality are jointly reported rather than treated as separate afterthoughts.

Keywords: energy-efficient AI; edge-cloud collaboration; English varieties; semantic caching; retrieval-augmented generation; pronunciation assessment; mobile learning

1. Introduction

English tutoring is rarely variety-neutral in practice. Learners ask for American English for media and workplace interaction, British English for examinations and mobility, Indian English for regional professional

communication, Australian English for relocation, and Canadian English for a North American target with its own spelling and lexical tendencies. A tutor that silently treats one variety as the default can produce feedback that is fluent but pedagogically misleading: a spelling correction, lexical substitution, or pronunciation warning may be valid for one target variety and inappropriate for another.

Recent language-learning systems have made interactive explanation and feedback easier to deliver, but speaking practice adds two deployment constraints that are often underreported. First, a speech tutor has a tight turn-taking budget. Delayed feedback breaks rehearsal, especially in pronunciation drills. Second, inference energy is visible to learners through battery drain, heat, and network dependence. Edge-only inference protects privacy and responsiveness but constrains the model; cloud-only inference increases coverage but shifts both latency and energy to the network and data center. The design question is therefore not simply whether the model can answer, but where and how each turn should be processed.

This paper addresses that question for English-variety learning. This study formalizes the optimization problem, specifies cache replacement and consistency, and replaces illustrative evaluation with a prototype-scale quantitative analysis that includes baselines, ablations, cache sensitivity, human feedback ratings, and high-load behavior. The work remains a system paper rather than a longitudinal classroom intervention; learning outcomes are measured through short controlled drills and next-turn correction, not through semester-scale proficiency growth.

The distinct contributions are summarized in Table 1. In this framework, caching and offloading are treated as part of the tutoring policy rather than as generic system optimizations. In this tutor, a reusable item is not merely a previous answer: it is a validated, variety-matched, evidence-versioned feedback unit. Likewise, a cloud call is not simply a latency fallback: it is triggered only when the predicted quality gain justifies its energy, latency, and privacy cost.

Table 1. Positioning of the proposed framework.

Design Issue	Conventional Treatment	This Manuscript
Target language model	English is often handled as a single target or style prompt.	The target variety is an explicit state variable used in routing, retrieval, pronunciation scoring, and cache matching.
Caching	Cache reuse is usually based on text similarity or system speed alone.	Semantic reuse requires exact task and variety match, evidence-version compatibility, and validation before write-back; inference-hot reuse is tracked separately.
Offloading	Cloud fallback is optimized mainly for latency or model accuracy.	The controller estimates quality, energy, latency, and privacy cost for each action and logs the reason for every offload.
Feedback grounding	Generated feedback can be fluent but weakly traceable.	Feedback is composed from selected evidence snippets and uses a diagnosis-example-next-step structure. Unsupported claims trigger retrieval or offload.
Evaluation	Single illustrative table or accuracy-only reporting.	Matched baselines, ablations, cache sensitivity, human feedback quality, statistical confidence intervals, and load tests are reported.

2. Related Work

AI tools for ESL/EFL learning have been reviewed as promising but uneven, with concerns around reliability, privacy, and learner dependence [1,2]. For speaking practice, studies of chatbots and AI-assisted interaction show that benefits depend on task design and scaffolding rather than conversational fluency alone [3,4]. Mobile-assisted pronunciation learning and CAPT research similarly emphasize the form of corrective feedback, the reliability of ASR, and the risk of biased scoring [5–7]. Recent pronunciation systems increasingly move toward structured diagnosis, including phoneme-, word-, and utterance-level feedback [8,9]. This layered diagnostic orientation is consistent with broader AI-enabled intelligent tutoring systems, which are designed to model learner states, monitor progress, and deliver personalized feedback, scaffolding, and intervention [10].

The deployment side is equally relevant. Green edge AI research argues that energy should be decomposed across device computation, networking, and serving-side overhead [11], while edge-cloud offloading for DNN applications is commonly treated as a constrained optimization problem under latency and energy budgets [12]. Model compression makes local inference feasible but does not remove memory and I/O bottlenecks [13]. More broadly, recent work on the environmental impact of LLMs shows that model size, inference demand, energy

consumption, carbon emissions, and water use must be considered together when evaluating whether AI systems are sustainable in practice [14]. Cache-assisted voice assistants and mobile LLM inference studies show that repeated-work avoidance can materially reduce latency and energy [15,16]. For content reliability, RAG has been proposed as a way to ground educational explanations in retrievable knowledge, although retrieval itself adds cost and must be controlled [17].

Our work sits at the intersection of these threads. Its novelty is not a new ASR model or a new large language model. It is a deployment and feedback-control framework in which English-variety fidelity is represented as a quality constraint, and in which energy-aware offloading, semantic caching, inference reuse, and evidence-grounded feedback are evaluated together.

A second strand of work concerns the controllability of generative feedback. LLM-based tutors can produce fluent, personalized hints, but prior studies also report generic explanations, premature answer disclosure, and weak traceability as recurring risks [18]. These risks are particularly salient in language learning because a learner may treat a single explanation as a rule. For that reason, the present architecture treats feedback generation as a constrained composition problem rather than as open-ended dialogue: task state, target variety, retrieved evidence, and no-leakage rules are all available to the controller before a response is released.

Speech robustness is the other architectural dependency that cannot be left implicit. Multi-variety tutoring must distinguish the target variety from the learner's accent and from acoustic noise. Lightweight adapter-based ASR adaptation and cross-modal accented-speech training provide practical mechanisms for improving robustness without replacing the whole speech stack [19,20]. In this work, those mechanisms motivate the separation between variety routing, ASR confidence estimation, pronunciation scoring, and cloud fallback: an uncertain transcript should change the processing path, not silently lower the quality of feedback.

3. Problem Formulation

A session S contains T turns. At turn t, the learner input is x_t, the target variety is v_t in {AmE, BrE, IndE, AusE, CanE}, and the task mode is m_t in {drill, pronunciation, variety knowledge, constrained conversation}. The system chooses an action a_t from four practical actions: semantic-cache reuse, local RAG plus generation, local pronunciation/generation without retrieval, or cloud fallback. Each action has predicted quality $\hat{Q}_t(a)$, energy $\hat{E}_t(a)$, latency $\hat{L}_t(a)$, and privacy risk $\hat{R}_t(a)$.

The controller selects the action that maximizes expected utility while satisfying minimum quality and response-time requirements:

$$a_t^* = \arg\max_a \left[\hat{Q}_t(a) - \lambda_E \hat{E}_t(a) - \lambda_L \hat{L}_t(a) - \lambda_P \hat{R}_t(a) \right] \tag{1}$$

$$\text{s.t.} \quad \hat{Q}_t(a) \geq Q_{\min}(m_t, v_t), \quad \Pr\left[L_t(a) \leq L_{\max}(m_t) \right] \geq \rho, \quad \text{policy}(a, x_t) = \text{allowed}.$$

$\hat{Q}_t$ is a composite pedagogical proxy. For variety knowledge, it combines variety fidelity, evidence support, and absence of unsupported claims. For pronunciation, it combines ASR confidence, diagnostic granularity, and whether the selected phonetic profile accepts legitimate variety-specific realizations. For drill tasks, it also penalizes answer leakage.

Session energy is counted at turn level with an explicit boundary:

$$E_{\text{session}} = \sum_{t=1}^{T} \left(E_{\text{dev},t} + E_{\text{net},t} + \text{PUE} \cdot E_{\text{srv},t} \right). \tag{2}$$

$$E_{\text{dev},t} = P_{\text{base}} \Delta_t + \sum_m P_m \Delta_{m,t}; \quad E_{\text{net},t} = P_{\text{tx}} T_{\text{up},t} + P_{\text{rx}} T_{\text{down},t} + P_{\text{tail}} \tau_{\text{tail},t}. \tag{3}$$

$$E_{\text{srv},t} = \kappa_{\text{gen}} N_{\text{tok},t} + \kappa_{\text{ret}} k_t + \kappa_{\text{asr}} d_{\text{audio},t}. \tag{4}$$

Here E_{dev} includes front-end audio processing, on-device ASR, local retrieval, local generation, cache operations, and pronunciation scoring. E_{net} covers request/response transmission and radio tail time for offloaded turns. E_{srv} is serving-side inference energy and is scaled by PUE to avoid reporting IT energy as if it were facility energy. The prototype evaluation uses PUE = 1.20; the formulas can be recalculated for other facilities.

To connect efficiency with pedagogy, we report normalized learning gain G from a short post-task drill or corrected-error-rate improvement, and learning efficiency LE = G/ $E_{session}$. LE is not intended to replace learning assessment; it is a reporting device that prevents energy savings from being discussed independently of educational value.

4. System Architecture and Methods

The system is organized as an edge-first teaching pipeline rather than as a general chatbot wrapped in a mobile interface. Its design starts from four stable states that must be carried across a session: the target variety, the task mode, the evidence state, and the device/network state. These states determine whether a turn can be answered from validated cache content, whether local retrieval and generation are sufficient, or whether cloud support is warranted. Keeping these states explicit is important because the same learner utterance may require different treatment in a spelling drill, a pronunciation turn, and a variety-knowledge question.

Conceptually, the architecture separates the data plane from the control plane. The data plane performs audio preprocessing, ASR, retrieval, generation, pronunciation scoring, and feedback rendering. The control plane supervises variety routing, cache validity, evidence sufficiency, offloading, and logging. This separation is intended to make the engineering trade-offs auditable: a cached response is not accepted because it is fast, and a cloud response is not accepted because it is large-model output; both must satisfy the same task, variety, evidence, and policy checks before reaching the learner.

4.1. Edge-First Workflow

Figure 1 shows the revised workflow. The learner's request first passes through local preprocessing. Speech input is denoised and transcribed by the on-device ASR module; text input bypasses ASR but still receives language-identification and safety checks. A variety router assigns or confirms the target variety. A task classifier then separates short drills, pronunciation practice, variety-knowledge questions, and constrained conversation. This separation is important because the cheapest correct action differs by task: a repeated spelling contrast may be served from cache, while a noisy pronunciation turn may need cloud ASR or human-readable uncertainty handling.

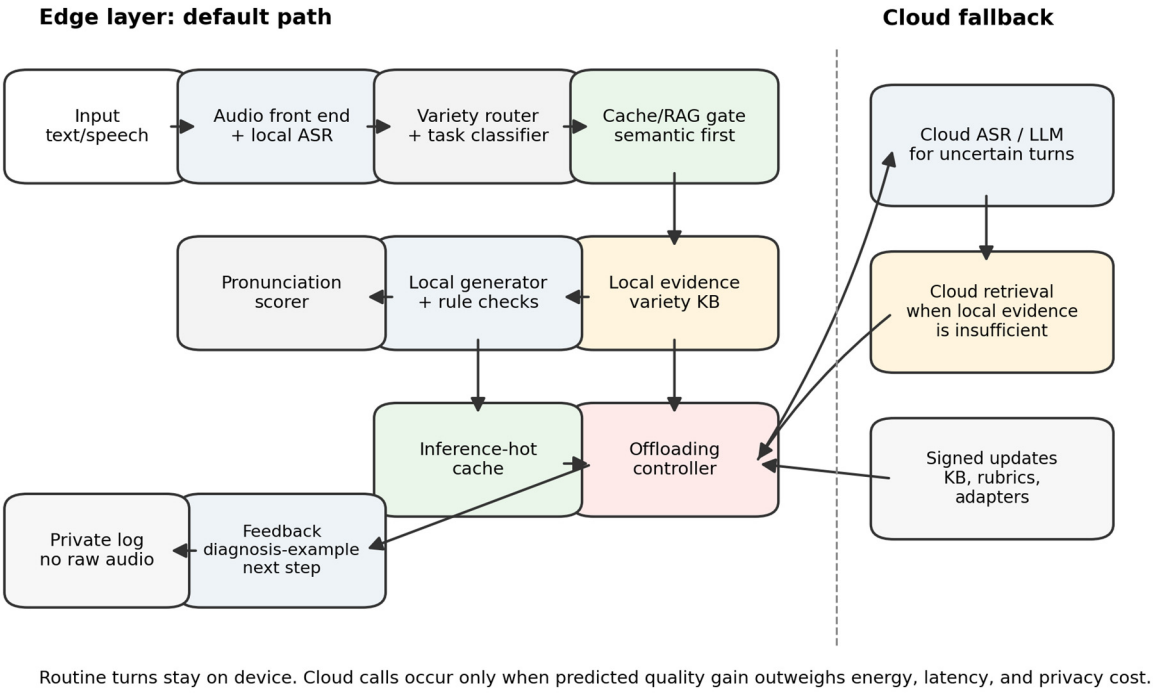

Figure 1. End-to-end edge-first workflow linking ASR, variety routing, cache lookup, retrieval, local generation, offloading, and feedback composition.

After routing, the request is converted into a compact tutoring state rather than passed directly to a generator. The state includes the normalized learner intent, target variety, task type, learner level band, ASR confidence when speech is used, and the current policy profile. This representation allows the same downstream modules to behave

differently under different pedagogical constraints. For example, a drill state enables no-answer-leak feedback and favors short templates, whereas a variety-knowledge state requires evidence identifiers and a contrastive example.

The local tutor engine therefore has three coordinated responsibilities. It first checks whether a validated semantic entry can answer the turn without new generation. If not, it retrieves compact evidence from the local variety knowledge base and prepares a bounded evidence pack. Finally, it produces a draft response using the local generator and rule checks, while the pronunciation scorer runs in parallel for speaking tasks. Only after these local estimates are available does the offloading controller compare the expected benefit of a cloud call with its energy, latency, and privacy cost.

Routine turns are intended to stay local. A semantic cache is queried before generation; if it misses, local retrieval packages evidence from a compact knowledge base; the local generator then produces a short draft under grounding constraints. Pronunciation scoring runs in parallel for speaking tasks. The offloading controller sees both the local draft uncertainty and device/network state before deciding whether to keep the turn local or send an encrypted request to the cloud. Returned cloud answers are not automatically cached; they pass through the same evidence and validation gates as local outputs.

This workflow also constrains how feedback is rendered. The feedback composer receives the selected evidence, pronunciation diagnostics, task state, and cache/offload decision, and then emits a short diagnosis-example-next-step sequence. In drills, the composer may give a cue or contrast but not the final answer; in pronunciation practice, it reports the accepted target-variety profile before identifying a deviation; in knowledge questions, it cites the selected evidence entry internally and avoids unsupported generalization. These constraints are deliberately ordinary from a teaching perspective, but they are essential engineering controls for avoiding fluent yet unstable feedback.

4.2. Dual-Layer Caching

The cache design has two layers with different failure modes. The semantic cache operates at the tutoring-interaction layer. Its entries contain a normalized intent embedding, task type, target variety, learner level band, feedback template, evidence identifiers, evidence_version, model_version, rubric_version, validation status, creation time, last access time, and a short no-raw-data audit record. A semantic hit is accepted only if cosine similarity is at least theta = 0.85 and task type and target variety match exactly. This prevents a British English spelling explanation, for example, from being reused in an American English correction turn.

Replacement uses a lightweight LFU-LRU hybrid. Each entry receives a keep score:

$$keep_i = 0.45\log\left(1+freq_i\right)+0.35\exp\left(-\frac{now-last_i}{\tau}\right)+0.20valid_i-0.50stale_i. \quad (5)$$

The system evicts the lowest keep score when capacity is reached. $freq_i$ captures repeated pedagogical use; the recency term protects newly useful items; $valid_i$ favors human- or rule-validated feedback; and $stale_i$ is set when the knowledge base, model, or rubric version changes. In the prototype, tau is seven days for drill feedback and thirty days for variety-knowledge entries.

The inference-hot cache is lower level. It keeps frequently used quantized weights, prompt prefixes, and short-context key-value states for follow-up turns. It is never used as a source of linguistic evidence. Its purpose is to reduce memory traffic and repeated context construction on the device. The two cache layers are logged separately because a semantic hit and an inference-hot hit have different pedagogical and systems meanings.

4.3. Cache Consistency, Update, and Security

Consistency is version-based rather than trust-based. A semantic entry is usable only when evidence_version, model_version, rubric_version, and policy_hash match the current runtime. Stale entries are lazily revalidated: they can be used to retrieve candidate evidence, but they cannot be served directly to the learner. Cloud write-back is disabled by default for open conversation and enabled only for validated short explanations or drill feedback. Cache updates are signed, and the local store is encrypted. Raw audio, full transcripts containing personal data, and free-form learner identifiers are not cached.

Figure 2 expands the cache/offload loop. The controller first asks whether a safe, version-consistent cache hit exists. If not, it retrieves evidence locally and asks whether local generation can satisfy the task's quality threshold. Only difficult turns cross the network boundary, and the reason category is logged as uncertainty, evidence insufficiency, or device/network constraint.

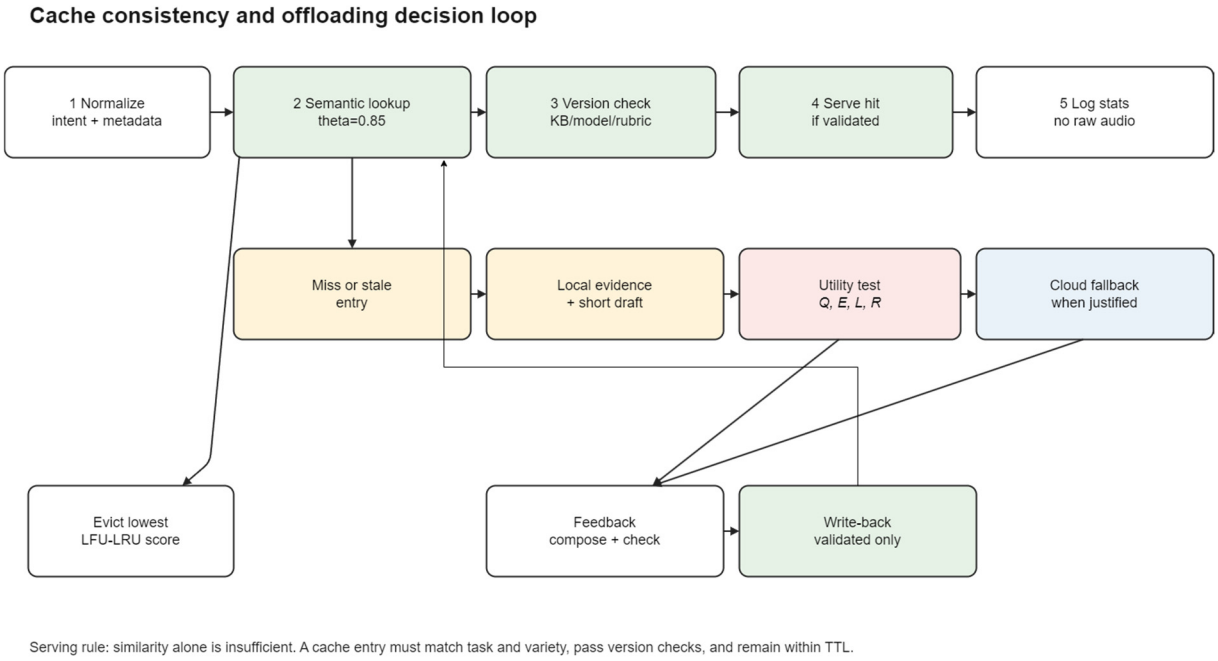

Figure 2. Cache consistency and offloading decision loop. Cache hits require similarity, exact variety/task match, and version compatibility; stale entries are revalidated before use.

4.4. Evidence-Grounded Feedback Generation

Evidence selection is explicit. For a variety-knowledge turn, the retriever applies a hard filter on target variety when the learner specifies one. Candidate entries are scored by semantic similarity, source reliability, register match, and diversity:

$$\text{score}(d) = 0.55\cos\left(\mathbf{e}_x, \mathbf{e}_d\right) + 0.15r_d + 0.15m_d + 0.15c_d - 0.20u_d. \tag{6}$$

The evidence pack normally contains two or three short entries, each with an evidence identifier and metadata. The generator receives the evidence before the instruction and must produce a concise response with three parts: diagnosis, variety-consistent example, and next practice step. If no evidence reaches the threshold, the system asks a clarification question or offloads retrieval instead of inventing a rule. This is the main safeguard against feedback that sounds plausible but is not grounded.

For pronunciation practice, evidence includes the target word, expected phonetic contrast, variety profile, ASR confidence, and diagnostic level. The feedback composer avoids penalizing accepted variety-specific pronunciations. For example, rhoticity is treated differently when the target is American English versus many British English contexts; the tutor reports the target explicitly instead of labeling one realization as universally correct.

4.5. Offloading Policy

The controller uses ASR confidence, acoustic noise, generation entropy, retrieval score margin, task type, battery level, thermal headroom, network round-trip time, packet loss, and bandwidth. Three policy profiles are supported: battery-saver, balanced, and quality-first. The balanced profile used in the experiments sets higher quality thresholds for pronunciation and variety-knowledge feedback than for spelling drills, because an inaccurate pronunciation diagnosis is more damaging than a delayed spelling example.

The policy is auditable. Each offloaded turn records one primary reason: uncertainty, evidence insufficiency, or device/network constraint. These logs are used only as aggregated statistics. They make it possible to ask when the hybrid architecture is actually helpful: it is strongest when repeated drill or explanation turns produce cache hits, while occasional noisy speech or low-evidence turns still benefit from cloud support.

4.6. Speech Adaptation, Diagnostics, and Feedback Composition

The speech stack distinguishes three related but separate problems: recognizing what the learner said, identifying the target variety, and judging whether the learner's production matches that target. The on-device ASR component supplies a transcript and confidence estimate; the variety router supplies the pedagogical target; and the pronunciation scorer supplies diagnostic evidence at phoneme, word, or utterance level. Keeping these

outputs separate reduces a common error in accent-sensitive tutoring: treating ASR uncertainty or non-target accent features as pronunciation mistakes.

For deployment, the architecture allows lightweight speech adaptation without changing the control policy. Adapter modules can be trained or updated for accent and domain robustness while the frozen backbone and downstream routing logic remain stable. Cross-modal accented-speech training can further improve recognition of difficult utterances, but the controller still treats low-confidence transcripts conservatively. When confidence is insufficient, the tutor can ask for repetition, switch to text confirmation, or offload ASR before issuing a corrective judgment.

The final feedback composer is intentionally narrow. It does not simply pass through the local or cloud model's answer. Instead, it merges the task state, selected evidence, scoring diagnostics, and policy constraints into a learner-facing response. This design preserves the stronger quantitative performance reported below while restoring the methodological emphasis of the original framework: system performance matters because it supports a controlled teaching loop, not because it produces longer or more fluent answers.

5. Data, Workload, and Evaluation Protocol

5.1. Data Sources and Labeling

The evaluation uses a controlled prototype workload rather than a scraped chat log. Its purpose is to stress the system components under comparable conditions across deployment modes. The written and knowledge components combine author-curated variety contrasts with examples derived from established resources for World Englishes and regional usage, including GloWbE and the International Corpus of English [21,22]. The speech component uses short, licensed or publicly documented reading and accent materials, including Common Voice and the Speech Accent Archive, together with author-written minimal-pair prompts [23]. Table 2 summarizes the evaluation workload by component, including the data scale, labeling and preprocessing procedures, and the role of each component in the evaluation protocol.

Labels are attached at three levels. Variety labels identify the target variety and the varieties for which an item is acceptable. Linguistic labels identify category: spelling, lexicon, grammar, pragmatics, pronunciation, or mixed usage. Feedback labels identify the intended intervention: correction, contrast explanation, pronunciation hint, drill cue, or clarification. Two annotators checked a stratified 20% sample; agreement was substantial for variety/category labeling (Cohen's kappa = 0.82). Disagreements were resolved by an applied-linguistics reviewer and recorded in the annotation log.

Table 2. Evaluation Data and Workload Composition.

Component	Size Used in Evaluation	Labels and Preprocessing	Purpose
Curated variety knowledge base	620 entries (124 per variety, balanced by category)	Sentence-level entries; metadata: variety, category, register, level, evidence ID, source pointer.	Local RAG, semantic-cache grounding, variety-fidelity scoring.
Written prompts and drill items	750 prompts (150 per variety)	Normalized spelling, lexical contrast, grammar/usage target, expected answer variants.	Cache reuse, spelling/vocabulary drills, constrained generation.
Speech prompts and utterances	1500 utterances from reading and minimal-pair tasks	VAD, noise tag, transcript check, target phoneme/word, target variety, ASR confidence.	Pronunciation scoring, ASR robustness, offloading decisions.
Matched session scripts	120 five-minute scripts (24 per variety)	Each script mixes drill, pronunciation, variety knowledge, and short conversation turns.	Baseline comparison, ablation, cache sensitivity, high-load replay.
Human feedback sample	300 sampled feedback turns	Three raters; 1–5 rubric for linguistic accuracy, evidence support, actionability; leakage flag.	Benchmark against rule-only and cloud NLP feedback baselines.

5.2. Experimental Setup

The edge device profile is a mid-range smartphone-class CPU/NPU with 8 GB RAM. The local stack uses a compact ASR model, a 4-bit quantized 1.3B-parameter instruction model, a small pronunciation scorer, and a local vector index for the curated knowledge base. The working set during local generation is 2.1–2.6 GB depending on whether inference-hot reuse is active. Devices below 4 GB RAM can run the template/RAG-only mode but are not expected to support continuous local generation.

The cloud baseline uses the same task prompts and evidence constraints but runs ASR/retrieval/generation remotely. Network conditions are replayed from three profiles: stable Wi-Fi (median RTT 35 ms), typical mobile (90 ms), and congested mobile (180 ms with 2% packet loss). Unless noted, Table 3 reports the typical mobile profile. Each deployment mode is evaluated on the same 120 session scripts. Confidence intervals are computed by paired bootstrap over sessions; paired Wilcoxon tests with Holm correction are used for the main comparisons.

Four deployment modes are compared: edge-only, cloud-only, hybrid without caching, and hybrid with both semantic and inference-hot caching. The hybrid mode uses the balanced policy profile and is allowed to offload ASR or generation when uncertainty is high, evidence is insufficient, or the device is thermally constrained.

Table 3. Baseline comparison across deployment modes (five-minute matched sessions).

Mode	Energy Wh [95% CI]	Latency Median/p95 ms	Variety Fidelity % [95% CI]	Gain G [95% CI]	LE G/Wh	Cloud/Cache Behavior
Edge-only	0.23 [0.21, 0.25]	286/612	88.7 [87.2, 90.2]	0.138 [0.119, 0.157]	0.60	Cloud 0%; semantic cache 31.8%; inference-hot 34.6%
Cloud-only	0.61 [0.57, 0.65]	594/1480	95.0 [94.0, 95.9]	0.162 [0.145, 0.181]	0.27	Cloud 100%; no local cache reuse
Hybrid, no cache	0.43 [0.40, 0.46]	397/970	94.0 [92.9, 95.1]	0.164 [0.146, 0.183]	0.38	Cloud 20.6%; cache disabled
Hybrid + cache	0.36 [0.34, 0.38]	331/806	94.2 [93.1, 95.2]	0.166 [0.149, 0.184]	0.46	Cloud 18.4%; semantic cache 42.3%; inference-hot 36.1%

Note: Energy includes device, network, and PUE-adjusted serving-side inference. G is normalized short-task gain; LE = G/$E_{session}$.

6. Results

6.1. Deployment-Mode Comparison

The hybrid cached design reduces total energy by 41.0% relative to cloud-only and by 16.3% relative to the same hybrid pipeline without caching. Median latency is 44.3% lower than cloud-only and 16.6% lower than hybrid without caching. These reductions are statistically significant for energy and latency ($p < 0.001$). Variety fidelity of the cached hybrid is not significantly different from cloud-only under the rubric ($p = 0.18$), but is higher than edge-only ($p < 0.01$). Edge-only remains the most energy-frugal mode and has the highest raw gain per Wh, but its lower variety fidelity and diagnostic depth make it less suitable for difficult pronunciation and knowledge turns.

6.2. Ablation Study

Table 4 reports the ablation results across energy, latency, variety fidelity, evidence support, and feedback actionability. The ablation highlights the trade-off. Removing the evidence gate makes the system faster but materially weakens evidence support and feedback actionability. Removing uncertainty-aware offloading saves energy, but the loss of variety fidelity is larger than the energy saving justifies for the balanced policy. Conversely, sending every cache miss to the cloud gives only a small fidelity gain over the full hybrid while giving up much of the energy and latency benefit.

Table 4. Ablation study for the hybrid design.

Condition	Energy Wh	Median Latency ms	Variety Fidelity %	Evidence Support %	Actionability %
Full hybrid: semantic cache + inference-hot cache + RAG + uncertainty offload	0.36	331	94.2	96.1	70.1
No semantic cache	0.39	361	94.1	95.8	68.4
No inference-hot cache	0.40	368	94.2	95.9	69.2
No RAG evidence gate	0.34	319	91.6	78.5	63.7
No uncertainty-aware offload (always stay local after cache miss)	0.30	296	89.1	93.5	61.2
Cloud offload for every cache miss	0.52	462	95.1	96.5	70.5

Note: Actionability is the percentage of sampled feedback turns after which the learner or scripted simulator produced a correct next attempt.

6.3. Cache-Hit Impact and Replacement Policy

Table 5 shows that semantic-cache capacity improves hit ratio, energy use, and latency up to the default 1000-entry setting, after which the returns flatten. The cache hit ratio has a near-linear effect until roughly 1000 validated entries, after which returns flatten. This is expected because tutoring interactions are repetitive but not unlimited: spelling and vocabulary explanations repeat often, while open conversation turns are less reusable. In the default setting, stale entries account for 2.8% of lookup candidates and are revalidated rather than served silently.

Table 5. Semantic-cache capacity sensitivity under the LFU-LRU replacement policy.

Semantic-Cache Capacity	Hit Ratio % [95% CI]	Energy Wh	Median Latency ms	Validated Write-Back % of turns	Observation
0 entries	0.0	0.43	397	0.0	Equivalent to hybrid without semantic caching.
250 entries	25.4 [22.8, 28.0]	0.40	371	4.2	Most gains come from repeated spelling and lexical contrasts.
500 entries	34.8 [31.9, 37.6]	0.38	350	5.7	Covers frequent drill and short explanation patterns.
1000 entries	42.3 [39.1, 45.4]	0.36	331	6.1	Default setting; good balance of memory and reuse.
2000 entries	45.1 [41.9, 48.3]	0.36	327	6.3	Diminishing returns; stale-entry checks increase slightly.

Note: Capacity counts validated semantic entries. Inference-hot cache capacity is fixed in this sensitivity run.

6.4. Feedback Quality Benchmark

As shown in Table 6, the full hybrid RAG feedback achieves the highest scores for linguistic accuracy, evidence support, actionability, and next-turn correction, while keeping answer leakage substantially lower than the unconstrained cloud NLP baseline. The evidence-grounded hybrid feedback is not simply more verbose. Raters preferred it because the response gave a specific contrast and a next practice action. The unconstrained cloud NLP baseline was linguistically strong but more likely to leak answers in drill tasks and to provide explanations without a traceable evidence basis.

Table 6. Human-rated feedback quality on 300 sampled turns.

System	Linguistic Accuracy 1–5	Evidence Support 1–5	Actionability 1–5	Next-Turn Correction %	Answer Leakage %
Rule-only feedback	3.98 [3.82, 4.14]	3.12 [2.95, 3.30]	3.54 [3.35, 3.72]	56.2	1.1
Cloud NLP feedback (no evidence constraint)	4.38 [4.22, 4.52]	3.51 [3.30, 3.70]	3.92 [3.74, 4.08]	62.7	7.4
Full hybrid RAG feedback	4.52 [4.37, 4.66]	4.58 [4.44, 4.70]	4.21 [4.06, 4.35]	67.9	2.3

Note: Raters used a 1–5 rubric. Answer leakage means the feedback supplied the answer in a drill where the learner was expected to produce it.

6.5. Network Load, Scalability, and Edge Limits

Table 7 reports the network-load replay results across 5 to 100 active learners, comparing hybrid cloud invocation, hybrid latency, cloud-only p95 latency, edge occupancy, and the dominant bottleneck. The hybrid architecture is most useful under moderate to high load because most routine turns remain on device and repeated explanations are served from cache. At 100 active learners, p95 latency rises but remains below the cloud-only replay because the cloud sees only about one quarter of the turns. The edge limit is not only raw compute; thermal headroom and memory pressure determine whether local generation can stay active. When the device temperature crosses the policy threshold, the controller moves to a template/RAG-only local mode or offloads difficult turns.

Table 7. High-load replay under congested mobile network conditions.

Active Learners	Hybrid Cloud Invocation %	Hybrid Latency Median/p95 ms	Cloud-Only p95 Latency ms	Edge CPU/NPU Occupancy %	Main Bottleneck
5	18.2	344/835	1570	43	None; cache absorbs repeated drills.
25	19.5	361/945	2050	50	Occasional cloud queueing.
50	21.3	394/1215	2780	59	Network tail latency and ASR uncertainty.
100	24.1	468/1760	>3500	74	Device thermal headroom and cloud queueing.

Note: The high-load replay uses the congested mobile profile: median RTT 180 ms, 2% packet loss, and bursty upstream contention.

7. Practical Deployment Considerations

7.1. Privacy and Security

The privacy design follows data minimization. Raw audio is processed locally and discarded unless the learner explicitly opts into a research recording mode. The released logs contain session IDs, turn IDs, task type, target variety, cache-hit flags, offload reason, coarse latency, and outcome flags; they do not contain raw audio or full learner identifiers. User IDs are salted hashes, timestamps are coarsened, and evidence is referenced by ID rather than copied as long text.

Cached data are encrypted at rest. Cache keys are derived from normalized intent representations and metadata rather than full transcripts. Cloud requests use transport encryption and carry only the minimum fields needed for the turn. Signed knowledge-base updates and version checks reduce the risk of cache poisoning. A teacher or learner can clear the local cache, disable cloud fallback, or require on-device-only operation for sensitive activities.

7.2. Integration with Learning Platforms

The system exposes three integration points that make deployment realistic. First, task scripts can be imported from an LMS as short drill or speaking-practice items. Second, feedback summaries can be exported as xAPI-style records containing target skill, target variety, score band, and next-practice recommendation. Third, teachers can review aggregate cache/offload statistics and feedback-quality flags without seeing raw audio. These interfaces allow the tutor to be used as a practice component inside an existing course rather than as a separate chatbot.

7.3. Boundary of the Current Evaluation

The evaluation supports claims about system trade-offs, not broad claims about long-term language acquisition. The learning proxy is short-horizon: corrected-error-rate improvement and next-turn correction after feedback. A longitudinal classroom study would be needed to test retention, transfer to spontaneous speech, and learner motivation. The system also covers only five English varieties. Other varieties and contact varieties require additional knowledge entries, speech adaptation, and local pedagogical review rather than simple prompt substitution.

8. Discussion

The results clarify where the hybrid architecture adds value. It does not beat edge-only on absolute energy. Instead, it occupies a practical middle point: much lower energy and latency than cloud-only, with fidelity close to cloud-only and substantially better than edge-only for difficult turns. This is the deployment regime many learning applications face: routine practice must be cheap and responsive, while a small number of uncertain turns need stronger inference or broader evidence.

The cache results also change the interpretation of repetition. In a general assistant, repeated questions can look like redundant traffic. In language learning, repetition is often the mechanism of practice. A semantic cache that stores validated variety-specific explanations improves consistency and reduces cost, but it also creates a responsibility: stale or mismatched feedback must not be served just because it is similar. Version tags, TTLs, and strict task/variety matching are therefore not implementation details; they are part of the pedagogical safety mechanism.

Finally, evidence grounding helps avoid a common failure mode of generative tutors: confident, fluent explanations that are hard to verify. The RAG layer is deliberately small and curated for high-frequency contrasts, because the goal is not to retrieve everything but to ground the feedback most likely to be repeated. Cloud retrieval remains useful for low-coverage cases, but it is a controlled exception rather than the default path.

9. Conclusions

This manuscript presents an energy-aware edge-cloud framework for English-variety tutoring with formalized action selection, versioned semantic caching, inference-hot reuse, evidence-grounded feedback, and uncertainty-aware cloud fallback. The evaluation shows that caching and selective offloading materially reduce energy and latency while preserving near-cloud variety fidelity. It also shows the limits of the design: edge-only is still the lowest-energy option, and the present learning metrics are short-term proxies. The value of the proposed framework is that these trade-offs are made measurable. For real deployments, a tutor should report not only whether it answers correctly, but how much energy, latency, evidence, and privacy cost were required to produce that answer.

Author Contributions

C.Y.: conceptualization, methodology, writing-original draft preparation; S.Z.: data curation, validation; L.Q.: software, visualization; M.W.: data curation, investigation; X.W.: supervision, writing-reviewing and editing; H.L.: supervision, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by the National Natural Science Foundation of China (62541330, 61272315).

Data Availability Statement

The evaluation protocol, task scripts, anonymized aggregate statistics, cache-hit definitions, offload-decision definitions, and human-rating rubric will be made available from the corresponding author upon reasonable request, subject to privacy, ethical, and institutional data-sharing constraints. For each reported metric, the available records include the sample size, mean, standard deviation, median, interquartile range, and 95% confidence interval where appropriate. Missing sessions are reported with counts and handling rules. Continuous variables are not discretized unless the discretization rule is stated. The energy-accounting boundary includes on-device audio processing, on-device ASR, local retrieval, local generation, pronunciation scoring, semantic-cache lookup/write, inference-hot reuse, network transmission for offloaded turns, and PUE-adjusted serving-side inference. Excluded components are unrelated OS background services, uncontrolled screen brightness changes, and third-party applications. All measurements are reported with the device profile, OS version, power mode, and network profile. A semantic-cache hit is defined as a case in which embedding similarity is at least 0.85, task_type and target_variety match exactly, and evidence/model/rubric versions are compatible. An inference-hot hit is defined as reuse of cached inference artifacts for the current generation step. Offload decisions are logged as either local or offload, with exactly one primary reason: uncertainty, evidence insufficiency, or device/network constraint.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used ChatGPT to assist with translation and language polishing. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Lo, C.K.; Yu, P.L.H.; Xu, S.; et al. Exploring the Application of ChatGPT in ESL/EFL Education and Related Research Issues: A Systematic Review of Empirical Studies. *Smart Learn. Environ.* **2024**, *11*, 50. https://doi.org/10.1186/s40561-024-00342-5.
2. Nguyen Hoang Mai Tram, N.; Nguyen, T.T.; Tran, C.D. ChatGPT as a Tool for Self-Learning English among EFL Learners: A Multi-Methods Study. *System* **2024**, *127*, 103528. https://doi.org/10.1016/j.system.2024.103528.
3. Du, J.; Daniel, B.K. Transforming Language Education: A Systematic Review of AI-Powered Chatbots for English as a Foreign Language Speaking Practice. *Comput. Educ. Artif. Intell.* **2024**, *6*, 100230. https://doi.org/10.1016/j.caeai.2024.100230.
4. Karatas, F.; Abedi, F.Y.; Gunyel, F.O.; et al. Incorporating AI in Foreign Language Education: An Investigation into ChatGPT's Effect on Foreign Language Learners. *Educ. Inf. Technol.* **2024**, *29*, 19343–19366. https://doi.org/10.1007/s10639-024-12574-6.

5. Dai, Y.; Wu, Z. Mobile-Assisted Pronunciation Learning with Feedback from Peers and/or Automatic Speech Recognition: A Mixed-Methods Study. *Comput. Assist. Lang. Learn.* **2023**, *36*, 861–884. https://doi.org/10.1080/09588221.2021.1952272.
6. Amrate, M.; Tsai, P.-H. Computer-Assisted Pronunciation Training: A Systematic Review. *ReCALL* **2025**, *37*, 22–42. https://doi.org/10.1017/S0958344024000181.
7. Mohsen, M.A.; Mahdi, H.S.; AlThebi, S.H.; et al. A Scientometric Study of Computer-Assisted Pronunciation Training in Second Language Acquisition: Technological Affordances and Research Trends. *Humanit. Soc. Sci. Commun.* **2025**, *12*, 438. https://doi.org/10.1057/s41599-025-04474-y.
8. Sungkur, R.K.; Shibdeen, N. Pronunciation Trainer for Second Language Learning Using Generative AI. *Int. J. Educ. Technol. High. Educ.* **2025**, *22*, 64. https://doi.org/10.1186/s41239-025-00561-x.
9. Yan, B.C.; Chen, B. An Effective Hierarchical Graph Attention Network Modeling Approach for Pronunciation Assessment. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2024**, *32*, 3974–3985. https://doi.org/10.1109/TASLP.2024.3449111.
10. Lin, C.-C.; Huang, A.Y.Q.; Lu, O.H.T. Artificial Intelligence in Intelligent Tutoring Systems toward Sustainable Education: A Systematic Review. *Smart Learn. Environ.* **2023**, *10*, 41. https://doi.org/10.1186/s40561-023-00260-y.
11. Mao, Y.; Yu, X.; Huang, K.-B.; et al. Green Edge AI: A Contemporary Survey. *Proc. IEEE* **2024**, *112*, 880–911. https://doi.org/10.1109/JPROC.2024.3437365.
12. Li, Z.; Yu, H.; Fan, G.; et al. Energy-Efficient Offloading for DNN-Based Applications in Edge-Cloud Computing: A Hybrid Chaotic Evolutionary Approach. *J. Parallel Distrib. Comput.* **2024**, *187*, 104850. https://doi.org/10.1016/j.jpdc.2024.104850.
13. Zhu, X.; Li, J.; Liu, Y.; et al. A Survey on Model Compression for Large Language Models. *Trans. Assoc. Comput. Linguist.* **2024**, *12*, 1556–1577. https://doi.org/10.1162/tacl_a_00704.
14. Ren, S.; Tomlinson, B.; Black, R.W.; et al. Reconciling the Contrasting Narratives on the Environmental Impact of Large Language Models. *Sci. Rep.* **2024**, *14*, 26310. https://doi.org/10.1038/s41598-024-76682-6.
15. Montoya Benitez, A.O.; Suarez Sarmiento, A.; Macias Lopez, E.M.; et al. Optimization of Energy Consumption in Voice Assistants through AI-Enabled Cache Implementation: Development and Evaluation of a Metric. *Technologies* **2025**, *13*, 19. https://doi.org/10.3390/technologies13010019.
16. Kim, H.; Lee, J.; Bahn, H. Rethinking I/O Caching for Large Language Model Inference on Resource-Constrained Mobile Platforms. *Mathematics* **2025**, *13*, 3689. https://doi.org/10.3390/math13223689.
17. Li, Z.; Wang, Z.; Wang, W.; et al. Retrieval-Augmented Generation for Educational Application: A Systematic Survey. *Comput. Educ. Artif. Intell.* **2025**, *8*, 100417. https://doi.org/10.1016/j.caeai.2025.100417.
18. Reddig, J.M.; Arora, A.; MacLellan, C.J. Generating In-Context, Personalized Feedback for Intelligent Tutors with Large Language Models. Int. J. Artif. Intell. Educ. 2025, 35, 3459–3500. https://doi.org/10.1007/s40593-025-00505-6.
19. Yue, X.; Gao, X.; Qian, X.; et al. Adapting Pre-Trained Self-Supervised Learning Model for Speech Recognition with Light-Weight Adapters. *Electronics* **2024**, *13*, 190. https://doi.org/10.3390/electronics13010190.
20. Dong, R.; Chen, J.; Long, Y.; et al. Enhanced Cross-Modal Parallel Training for Improving End-to-End Accented Speech Recognition. *Speech Commun.* **2025**, *169*, 103188. https://doi.org/10.1016/j.specom.2025.103188.
21. Davies, M.; Fuchs, R. Expanding Horizons in the Study of World Englishes with the 1.9 Billion Word Global Web-Based English Corpus (GloWbE). *Engl. World-Wide* **2015**, *36*, 1–28. https://doi.org/10.1075/eww.36.1.01dav.
22. Greenbaum, S. (Ed.) *Comparing English Worldwide: The International Corpus of English*; Clarendon Press: Oxford, UK, 1996.
23. Ardila, R.; Branson, M.; Davis, K.; et al. Common Voice: A Massively-Multilingual Speech Corpus. In Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), Marseille, France, 11–16 May 2020; pp. 4218–4222.