Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.088561

ARTICLE

Privacy-Preserving Edge Intelligence for Speaker Verification via Uncertainty-Aware Adaptive Feature Offloading

Yongfeng Zhang[1,*] **and Jie Chen**[2,*]

[1]The School of Software Engineering, East China Normal University, Shanghai, China
[2]The College of Electronic Engineering, National University of Defense Technology, Hefei, China

*Corresponding Authors: Yongfeng Zhang. Email: 71275902023@stu.ecnu.edu.cn; Jie Chen. Email: jchen202209@163.com

Received: 06 July 2026; Accepted: 18 August 2026; Published: 15 September 2026

ABSTRACT: Speaker verification on phones, wearables, and voice-enabled Internet-of-Things gateways must balance local privacy with reliable decisions under adverse audio. Existing privacy-preserving verification schemes generally protect a fixed representation, whereas edge-offloading policies usually adapt computation without attaching an explicit feature-disclosure budget; neither line alone coordinates trial uncertainty, communication state, and privacy expenditure. This paper presents privacy-preserving uncertainty-aware adaptive feature offloading (P-UAFO), an edge-intelligence framework that keeps raw audio local and transmits only clipped, projected, quantized, and Gaussian-perturbed intermediate features when their expected benefit justifies resource cost. Its online pipeline first estimates decision uncertainty and resource state, filters infeasible actions, selects local, low-tier, or high-tier processing, and fuses the returned score only when cloud evidence is sufficiently reliable. We establish a feature-level differential-privacy guarantee, an uncertainty threshold for offloading, a variance-optimal fusion rule, and a bounded-regret result under prediction error. The evaluation uses reproducible controlled events covering clean/noisy conditions, 0.4–12 Mb/s uplinks, 20–90 ms round-trip delay, and perturbed resource predictions. The results validate the controller logic and its practical ability to avoid unnecessary feature transmission; they do not constitute a public-corpus or mobile-device benchmark, which remains necessary before real-world deployment claims can be made.

KEYWORDS: Speaker verification; edge intelligence; cloud–edge collaboration; privacy-preserving inference; adaptive offloading

1 Introduction

Speaker verification determines whether enrollment and probe utterances belong to the same speaker. It supports mobile authentication, intelligent access control, call-center protection, and voice-enabled Internet-of-Things services. Classical factor-analysis systems and neural embeddings have improved discrimination substantially [1–3], but an edge endpoint must still manage short utterances, channel mismatch, limited energy, and variable connectivity.

Cloud assistance can improve difficult trials, whereas local-only inference avoids communication. A fixed split is often inefficient because it consumes network and privacy resources on easy inputs. Collaborative edge–cloud inference and recent offloading studies instead motivate an input- and state-dependent placement decision [4–7]. Privacy is equally central: intermediate features can retain identity-related structure, so avoiding waveform transfer alone does not establish a privacy guarantee [8]. A specific gap therefore remains: privacy-preserving speaker-verification work commonly assumes a predetermined representation

or processing path, while context-aware offloading work optimizes latency and energy without a formal, action-dependent privacy budget. Existing approaches also seldom couple calibrated verification uncertainty with tier-specific disclosure and confidence-aware score fusion.

We propose privacy-preserving uncertainty-aware adaptive feature offloading (P-UAFO). The device keeps raw audio, acoustic representations, and full embeddings local. It sends a clipped, projected, quantized, and Gaussian-perturbed feature only when the predicted risk reduction outweighs latency, energy, and privacy expenditure. Self-supervised encoders provide strong local representations [9,10], while compact speaker models and speech hardware make an edge front end increasingly practical [11,12].

The principal scientific contributions are threefold. First, P-UAFO casts each verification request as a constrained three-action decision that jointly prices error risk, delay, edge energy, and feature-level privacy. Second, it introduces a protected tiered interface whose disclosure level is selected from calibrated uncertainty and current resource feasibility rather than from uncertainty alone. Third, it derives privacy, threshold, fusion, and prediction-error regret properties and evaluates the resulting controller against local-only, fixed-split, always-cloud, and uncertainty-only policies in a reproducible controlled study. At run time, the processing sequence is: local scoring and uncertainty estimation, resource prediction, feasibility screening, utility-based tier selection, protected packet construction when needed, and confidence-weighted score fusion. The framework is intended for deadline-constrained inference and is compatible with adaptive service placement mechanisms [13]; empirical claims about public-corpus accuracy and mobile deployment are deliberately excluded from the present scope.

2 System Formulation

2.1 Verification State and Uncertainty

For enrollment x^e and probe x^p, the label is $y = 1$ for a target trial and $y = 0$ otherwise. An edge encoder $f_e(\cdot;\theta_e)$ produces an intermediate sequence H and a normalized embedding

$$z_e(x) = \frac{g_e(H)}{\|g_e(H)\|_2}, \qquad H = f_e(x;\theta_e), \tag{1}$$

where g_e denotes temporal pooling and projection. The local score is

$$s_e = \langle z_e(x^e), z_e(x^p) \rangle . \tag{2}$$

The edge also estimates a calibrated uncertainty

$$u = \mathrm{clip}_{[0,1]}\left[\alpha H_2(p_e) + \beta \frac{v_e}{v_e + \rho} + \chi d_{\mathrm{ood}}(H)\right], \tag{3}$$

from posterior ambiguity, score variance, and audio quality, where $\alpha + \beta + \chi = 1$. Here $H_2(p) = -p\log p - (1-p)\log(1-p)$, p_e is a calibrated local correctness probability, and v_e can be estimated from lightweight stochastic heads or dropout samples. The coefficients are fitted on a held-out calibration set with a proper scoring rule. Calibration is required because a raw score is not a reliable error estimate [14,15].

For every action, a conditional predictor $\hat{R}_a(\zeta)$ estimates the probability of an incorrect final decision. It uses the uncertainty vector, local score margin, duration and quality statistics, and the current resource state. The predictor is deliberately separate from the privacy mechanism: uncertainty describes expected decision quality, whereas feature protection limits what an offloaded representation reveals. The controller state is $\zeta = (u, B, r, p_b, q_c)$, containing uncertainty, predicted throughput, round-trip delay, edge energy state, and cloud load.

2.2 Protected Feature Interface

Fig. 1 summarizes the architecture. The microphone signal, log-Mel representation, and full edge embedding remain inside the edge trust boundary. The cloud receives a protected packet and returns a refined score with a confidence proxy.

For a feature $h \in \mathbb{R}^d$, the device clips

$$\bar{h} = \mathrm{clip}_C(h) = h \min\left(1, \frac{C}{\|h\|_2}\right), \tag{4}$$

then forms a tier-dependent packet

$$\tilde{h}_a = Q_{b_a}\left(P_a \bar{h} + n_a\right), \qquad n_a \sim \mathcal{N}(0, \sigma_a^2 I), \quad a \in \{1, 2\}. \tag{5}$$

Tier 1 uses fewer dimensions and bits than Tier 2. Projection and quantization reduce packet size; clipping and Gaussian noise provide the declared feature-level privacy mechanism. This follows the principle of constraining the representation exposed outside the device [16,17]. For target (ϵ_a, δ),

$$\sigma_a \geq \frac{2C\sqrt{2\log(1.25/\delta)}}{\epsilon_a}. \tag{6}$$

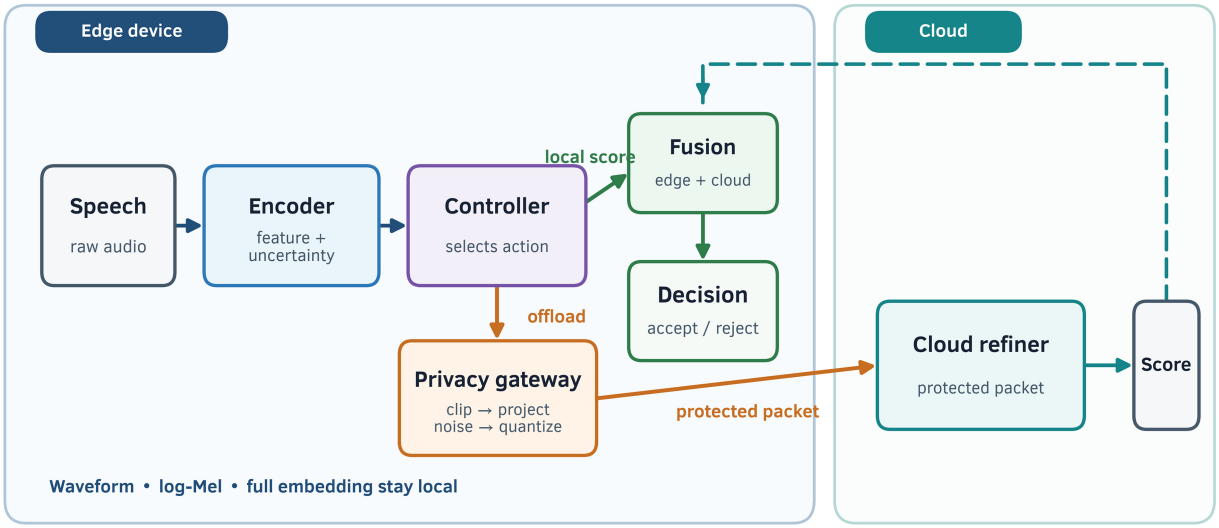

Figure 1: P-UAFO workflow. Raw speech and full embeddings remain at the edge; the cloud receives a protected feature packet only when offloading is selected.

Fig. 2 details the layered feature transformation. The guarantee is scoped to adjacent clipped features and does not replace authenticated transport, metadata minimization, or attack-based evaluation.

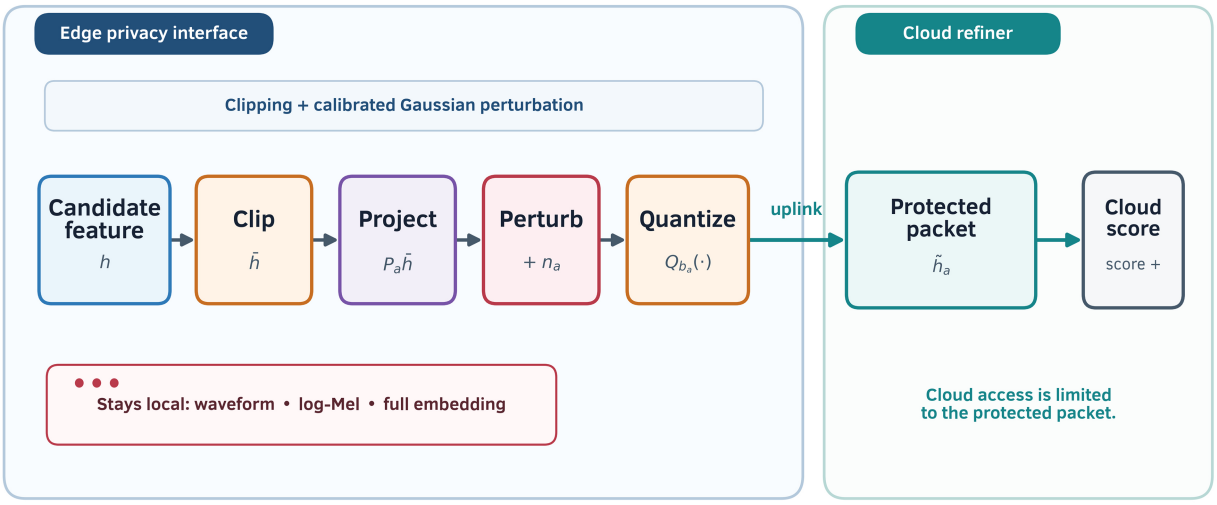

Figure 2: Layered privacy transformation before cloud refinement. Clipping and calibrated Gaussian perturbation provide the formal feature-level privacy control, while projection and quantization reduce packet dimensionality and communication load.

2.3 Resource-Aware Tier Selection

For action $a \in \{0, 1, 2\}$, local inference is $a = 0$. Approximate latency and energy are

$$T_a = T_a^e + \mathbb{I}(a > 0)\left(r + \frac{d_a b_a}{B} + T_a^c\right), \qquad E_a = E_a^e + \mathbb{I}(a > 0)\left(P_{\text{tx}}\frac{d_a b_a}{B} + E_a^{\text{crypto}}\right). \tag{7}$$

The controller predicts risk $\hat{R}_a(\zeta)$ and maximizes

$$\mathcal{U}_a = -\hat{R}_a - \lambda_T T_a - \lambda_E E_a - \lambda_P\, \epsilon_a + \lambda_G \hat{G}_a, \qquad a^* = \arg\max_{a \in A} \mathcal{U}_a, \tag{8}$$

over feasible actions. Feasibility requires $T_a \le T_{\max}$, $E_a \le E_{\max}$, and $\epsilon_a \le \epsilon_{\max}$, with $\epsilon_0 = 0$. Thus high uncertainty is not a command to upload more information: constrained bandwidth or a strict privacy budget may still favor local inference.

The controller uses a small finite action set instead of optimizing a continuous split point. This makes projection validation, transmission accounting, and privacy-budget bookkeeping practical on constrained devices. The low-tier packet is intended for moderate uncertainty, while Tier 2 is selected only when its additional score value offsets the larger latency and feature expenditure.

Fig. 3 summarizes the uncertainty-aware inference tiers and the resource-aware feasibility gate used to select among local, Tier 1, and Tier 2 processing.

For offloaded trials, the final score is

$$s_f = (1 - w_a)s_e + w_a s_c^{(a)}, \qquad w_a = \text{sigmoid}\left(\kappa[u - u_a^0] - \eta\, \epsilon_a\right), \tag{9}$$

or the inverse-variance weight derived below. The local score is therefore retained when the cloud estimate is not sufficiently reliable. The interface can be paired with efficient temporal and open-set speaker architectures without changing the controller logic [18,19].

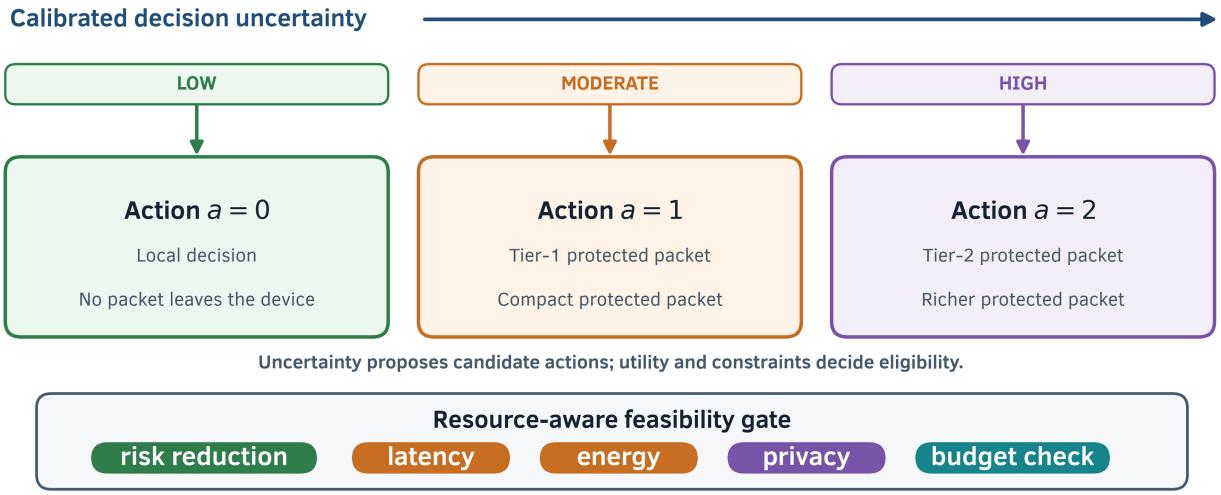

Figure 3: Uncertainty-aware inference tiers. Feasibility and utility jointly determine local, low-tier, or high-tier processing.

3 Training and Online Procedure

The edge and cloud branches are trained for discriminative speaker verification. In addition to the backbone loss ℓ_{aam}, we use score consistency and feature distillation:

$$\ell_{\text{rep}} = \ell_{\text{aam}} + \mu_1 \mathbb{E}[(\hat{p}_e - \hat{p}_c)^2] + \mu_2 \mathbb{E}[1 - \langle z_c(\tilde{h}_a), z_c(h) \rangle] + \mu_3 \mathbb{E}[\max(0, \|h\|_2 - C)]. \tag{10}$$

The controller is fitted on held-out trials paired with resource states. It minimizes the predicted cost of the selected feasible action,

$$\ell_{\text{ctl}} = \mathbb{E}\left[\hat{R}_{a^*} + \lambda_T T_{a^*} + \lambda_E E_{a^*} + \lambda_P \, \epsilon_{a^*}\right]. \tag{11}$$

This separates encoder learning from policy fitting. Margin-based objectives and disentangled speaker representations help retain useful local discrimination while enabling cloud refinement [20,21].

To avoid optimistic resource predictions, the feasible set is

$$\mathcal{A}_{\text{rob}} = \left\{a : \widehat{T}_a + b_T \le T_{\max},\ \widehat{E}_a + b_E \le E_{\max},\ \epsilon_a \le \epsilon_{\max}\right\}. \tag{12}$$

Online, the device computes local embeddings, score, and uncertainty; predicts each feasible action's risk and resource cost; selects the highest-utility action; sends a protected packet only when offloading; fuses the returned score; and records aggregate audit statistics for recalibration. The finite three-action design simplifies packet validation and privacy accounting. Table 1 records the complete sequence.

Table 1: Pseudo-code of P-UAFO.

Step	Procedure
Input	Enrollment x^e, probe x^p, device and network state ζ, privacy budget $\epsilon_{\max}$, deadline $T_{\max}$.
1	Compute edge features H^e, H^p, normalized local embeddings, local score s_e, local margin, and calibrated uncertainty u using (3).
2	For each $a \in \{0,1,2\}$, predict $\hat{R}_a$, $\widehat{T}_a$, $\widehat{E}_a$, and set $\epsilon_0 = 0$. Construct $\mathcal{A}_{\text{rob}}$ using (12).

(Continued)

Table 1 (continued)

Step	Procedure
3	For each $a \in \mathcal{A}_{\text{rob}}$, calculate $\mathcal{U}_a$ by (8). Select $a^* = \arg\max_{a \in \mathcal{A}_{\text{rob}}} \mathcal{U}_a$.
4	If $a^* = 0$, output $s_f = s_e$ and make the local threshold decision.
5	Otherwise, clip h, form $\tilde{h}_{a^*} = Q_{b_{a^*}}(P_{a^*}\bar{h} + n_{a^*})$, and transmit only the authenticated packet and tier identifier.
6	Receive $(s_c^{(a^*)}, v_c)$ from the cloud. Compute w_{a^*} from (9) or the inverse-variance estimate. Set $s_f = (1 - w_{a^*})s_e + w_{a^*}s_c^{(a^*)}$.
7	Record action, estimated and observed latency, packet size, privacy expenditure, and final decision for online recalibration.
Output	Final score s_f, binary decision $\hat{y}$, selected tier a^*, and audit metadata.

3.1 Implementation Considerations

Let C_e denote edge-encoder cost, $C_c^{(a)}$ cloud-refiner cost, and d_a packet dimension. The local action costs $O(C_e)$ and has no communication. An offloaded action adds a projection cost of $O(d_a d)$ in the dense case, although a structured projection or learned bottleneck can reduce the practical cost. Its payload is $d_a b_a$ bits plus a small authenticated header, and evaluating three utility values is negligible beside feature extraction.

The cloud is assumed to be a service component trusted for the declared refinement purpose, not an unrestricted recipient. It observes protected activations and minimal control metadata. A deployment should encrypt transport, prevent replay, rate-limit requests, avoid logging payloads, and maintain an explicit retention policy. These measures are complementary to the feature-level mechanism rather than substitutes for it.

To separate controller overhead from encoder and network delay, we also provide a reference host-side microbenchmark for the protected-feature operations. On an Intel Xeon Platinum 8573C CPU, the measured median times were 0.0022 ms for clipping, 0.0023/0.0038 ms for Tier 1/Tier 2 dense projection, 0.0043/0.0064 ms for projection plus Gaussian perturbation, 0.0072/0.0087 ms for quantization and serialization, and approximately 0.0005 ms for AES-GCM protection. The script performs 500 warm-up calls and 10,000 timed repetitions and reports both median and 95th percentile values. These numbers are implementation references only: they exclude edge encoder inference, radio transmission, and cloud processing and must not be interpreted as measurements on a phone, wearable, or embedded board.

4 Analytical Properties

Theorem 1 (Feature-level privacy): *For $\delta \in (0,1)$ and $a > 0$, if (6) holds, then $\mathcal{M}_a(\bar{h}) = P_a\bar{h} + n_a$ is (ϵ_a, δ)-differentially private with respect to adjacent clipped features. Quantization and cloud scoring preserve this guarantee.*

Proof: For adjacent clipped features $\bar{h}$ and $\bar{h}'$, the sensitivity of the projected feature is

$$\Delta_2 = \left\| P_a\bar{h} - P_a\bar{h}' \right\|_2 \le \|P_a\|_2 \left\| \bar{h} - \bar{h}' \right\|_2 \le 2C. \tag{13}$$

With the noise scale in (6), the Gaussian mechanism is (ϵ_a, δ)-differentially private. The stated scale is obtained by substituting the bound $\Delta_2 \le 2C$. Quantization, cloud refinement, score calibration, and thresholding are post-processing operations. The result is limited to the stated feature-adjacency relation and does not protect identifiers or metadata deliberately released outside the mechanism. □

Proposition 1 (Offloading threshold): *For local and one feasible offloaded action, let $D(u) = R_0(u) - R_1(u)$ be continuous and nondecreasing and let $K = \lambda_T(T_1 - T_0) + \lambda_E(E_1 - E_0) + \lambda_P\,\epsilon_1$. The controller offloads exactly when $D(u) \ge K$. If D is strictly increasing, the boundary is the unique threshold $u^* = D^{-1}(K)$ whenever K lies in the range of D.*

Proof: The utility difference is $\mathcal{U}_1 - \mathcal{U}_0 = D(u) - K$, so monotonicity directly gives the threshold rule. □

Fig. 4 visualizes the resulting utility regions under nominal resources. Changing bandwidth or the privacy weight shifts the corresponding threshold.

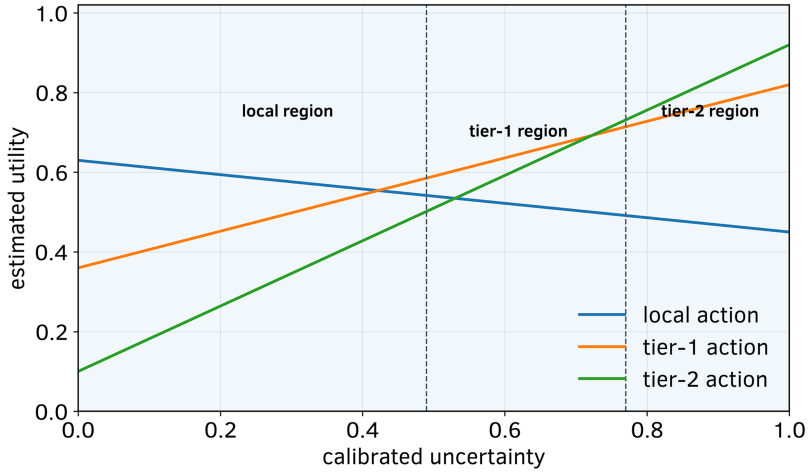

Figure 4: Controller utility regions under nominal resources.

Theorem 2 (Confidence-weighted fusion): *Suppose calibrated margins satisfy $\hat{m}_e = m + \xi_e$ and $\hat{m}_c = m + \xi_c$, where independent zero-mean errors have conditional variance proxies $v_e^2(u)$ and $v_c^2(u)$. Then*

$$w^*(u) = \frac{v_e^2(u)}{v_e^2(u) + v_c^2(u)}, \qquad \Pr(\hat{m}_{w^*} \le 0 \mid u) \le \exp\left(-\frac{m^2}{2v_*^2}\right), \tag{14}$$

where $v_*^2 = v_e^2 v_c^2/(v_e^2 + v_c^2)$.

Proof: The fused variance proxy is

$$v_w^2 = (1-w)^2 v_e^2 + w^2 v_c^2. \tag{15}$$

Its derivative is $-2(1-w)v_e^2 + 2wv_c^2$, which is zero at w^*; strict convexity proves optimality. Substitution gives $v_*^2 = v_e^2 v_c^2/(v_e^2 + v_c^2)$. Since the fused estimation error is sub-Gaussian with proxy v_*^2, the lower-tail inequality gives (14). □

Theorem 3 (Bounded selection regret): *Let U_a and $\widehat{U}_a$ be the true and predicted utilities of feasible actions. If $|\widehat{U}_a - U_a| \le \eta$ for every action, then the action $\hat{a} = \arg\max_a \widehat{U}_a$ satisfies*

$$U_{a^\star} - U_{\hat{a}} \le 2\eta, \qquad a^\star = \arg\max_a U_a \tag{16}$$

Proof: Because $\widehat{U}_{\hat{a}} \geq \widehat{U}_{a^\star}$, $U_{a^\star} \leq \widehat{U}_{a^\star} + \eta \leq \widehat{U}_{\hat{a}} + \eta \leq U_{\hat{a}} + 2\eta$. □

These results separate feature privacy, tier selection, score fusion, and imperfect resource prediction rather than treating them as one undifferentiated objective.

5 Controlled Simulation and Discussion

5.1 Protocol and Main Results

We use a reproducible Monte Carlo study to examine controller logic; it is not a corpus-based speaker-verification benchmark. One simulation event represents one enrollment–probe decision request. Its control environment contains an audio condition (clean or noisy), a bandwidth state, round-trip delay, edge-energy state, and cloud-load state. The action predictor observes noisy estimates of these variables rather than their realized values, which permits a controlled study of resource-prediction error. The simulation contains 50,000 target and 50,000 non-target trials, uncertainty-dependent score separation, 20–90 ms round-trip delay, and 0.4–12 Mb/s uplink throughput. Tier 1 uses a 48-dimensional 6-bit packet with $\epsilon_1 = 1.25$, Tier 2 uses a 96-dimensional 8-bit packet with $\epsilon_2 = 2.10$, the latency deadline is 180 ms, and $\delta = 10^{-5}$. "Synthetic EER" denotes equal-error rate computed from generated score distributions; it must not be read as VoxCeleb, CN-Celeb, or another corpus result. Latency and energy are generated from the declared event profiles, while mean ϵ is the action-weighted feature-level privacy expenditure. The accompanying scripts regenerate the illustrative profiles. Table 2 lists the principal simulation settings.

Table 2: Simulation settings.

Parameter	Value
Target and non-target trials	50,000 each
Random seed	20260702
Edge-only compute time	42 ms
Cloud refinement time	58–78 ms
RTT range	20–90 ms
Uplink throughput	0.4–12 Mb/s
Low-tier packet	48 dimensions, 6 bits, $\epsilon_1 = 1.25$
High-tier packet	96 dimensions, 8 bits, $\epsilon_2 = 2.10$
Privacy failure probability	$\delta = 10^{-5}$
Latency deadline	180 ms
Utility weights	$\lambda_T = 0.0025$, $\lambda_E = 0.015$, $\lambda_P = 0.045$

Table 3 compares nominal policies. Local-only never communicates; fixed split always sends the high-tier packet; always-cloud represents unconstrained remote processing; uncertainty-only ignores resource and privacy prices; and P-UAFO applies the full feasibility and utility rule. These aligned baselines support the quantitative comparison. Cross-paper numbers are not directly comparable because datasets, encoders, hardware, and threat models differ. Always-cloud gives the lowest synthetic EER but incurs the largest delay and privacy expenditure. Local-only avoids communication but has the weakest synthetic EER. P-UAFO selects the high tier only when its predicted benefit exceeds the added cost.

Table 3: Synthetic main results.

Method	Synthetic EER (%)	Latency (ms)	Edge Energy (mJ)	Mean ϵ	Offload Rate (%)
Local-only	5.62	42	18.1	0.00	0.0
Always-cloud	3.22	244	46.7	2.90	100.0
Fixed split	3.77	151	35.4	2.10	100.0
Uncertainty-only	3.61	113	31.2	1.90	61.8
P-UAFO	3.31	119	32.4	1.35	57.2

Relative to local-only, P-UAFO reduces synthetic EER by 2.31 percentage points. Relative to fixed split, it lowers mean privacy expenditure by 35.7% ([2.10–1.35]/2.10), reduces the offload rate by 42.8 percentage points, and decreases mean latency by 21.2%. Always-cloud is 0.09 percentage points lower in synthetic EER, but it more than doubles latency and raises privacy expenditure. All percentage statements in the tables and discussion were recalculated from the reported values. Fig. 5 displays this operating region.

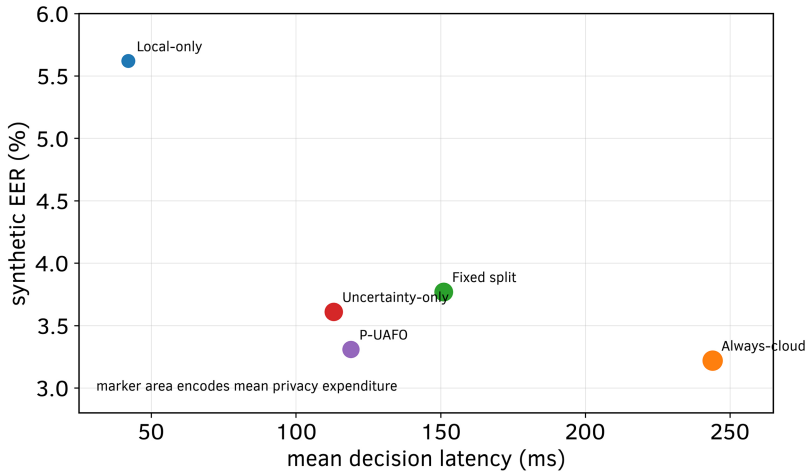

Figure 5: Synthetic accuracy–latency–privacy trade-off. Marker area encodes mean privacy expenditure.

The comparison is intentionally not framed as a claim that adaptive offloading surpasses unconstrained cloud inference on accuracy. Rather, it identifies a controllable operating point: services with a strict delay or exposure budget can accept a small numerical gap to all-cloud processing in exchange for selective transmission. Conversely, a privacy-sensitive application can penalize feature disclosure more strongly by increasing λ_P, whereas λ_T and λ_E separately control latency and energy penalties. Because $\hat{R}_a$ enters (8) with unit coefficient, its relative influence can be increased by reducing the competing cost weights, without changing the feature interface.

Table 4 summarizes the component ablations.

Ablations support the intended role of each component. Replacing calibrated uncertainty with an uncalibrated proxy increases synthetic EER from 3.31% to 3.58% and raises the offload rate from 57.2% to 64.1%. Removing the privacy price yields a slightly lower synthetic EER (3.24%) but increases mean ϵ from 1.35 to 2.03. Restricting the policy to one cloud tier increases latency from 119 to 132 ms. At low throughput, the controller selects local or Tier 1 actions more often; the associated scripts retain the full sensitivity profiles.

Table 4: Synthetic ablation results.

Variant	Synthetic EER (%)	Latency (ms)	Mean ϵ	Offload Rate (%)	Utility
P-UAFO	3.31	119	1.35	57.2	−0.512
No calibration	3.58	128	1.53	64.1	−0.563
No privacy price	3.24	137	2.03	79.5	−0.548
No tiering	3.38	132	1.82	61.9	−0.545
No fusion	3.47	119	1.35	57.2	−0.534

We further perturb the throughput, round-trip-delay, and cloud-load estimates before action selection. Table 5 shows that the robust feasibility margin gradually reduces offloading as mean absolute prediction error grows. The resulting synthetic EER degrades smoothly rather than abruptly, and the deadline-miss rate remains below 5% at 30% prediction error. This experiment addresses policy sensitivity to imperfect network-state collection; it does not substitute for traces from an operational network.

Table 5: Synthetic sensitivity to resource-prediction error.

Prediction Error (%)	Synthetic EER (%)	Offload Rate (%)	Deadline Misses (%)	Mean ϵ
0	3.31	57.2	2.1	1.35
10	3.36	55.8	2.7	1.31
20	3.44	53.1	3.6	1.26
30	3.57	49.4	4.8	1.18

5.2 Bandwidth and Privacy Sensitivity

Fig. 6 evaluates the effect of uplink throughput. Cloud-assisted policies improve when bandwidth increases, but P-UAFO retains more trials locally or selects Tier 1 when the predicted response cannot meet the deadline. The local-only baseline is independent of communication but preserves its larger synthetic EER.

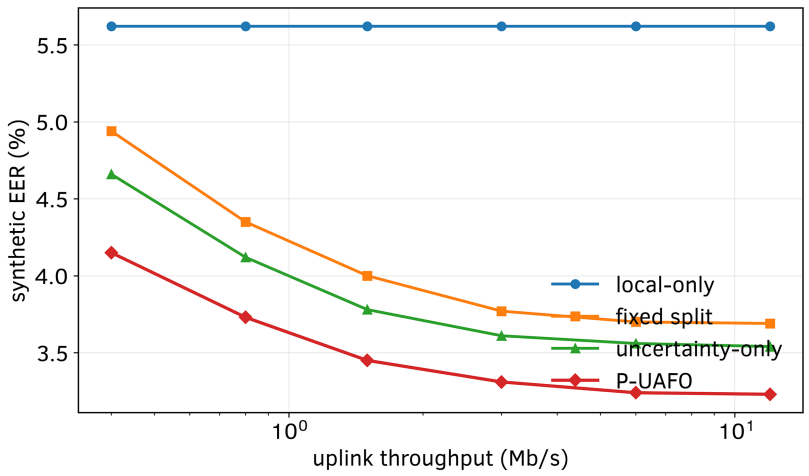

Figure 6: Synthetic accuracy under bandwidth variation.

Fig. 7 compares the mean feature-level privacy expenditure across clean/noisy and high/low bandwidth profiles. Fixed split uses the same rich packet regardless of context, whereas P-UAFO can reduce its average budget under constrained conditions while retaining the option of selected cloud refinement.

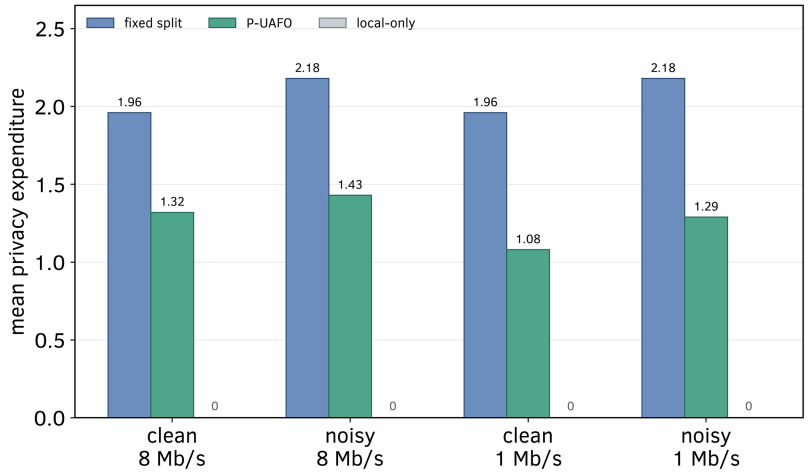

Figure 7: Colored comparison of mean privacy expenditure across operating profiles. The adaptive P-UAFO policy reduces feature-level privacy expenditure relative to fixed split while preserving the option of selective cloud refinement.

Table 6 gives the corresponding packet statistics. Noisy audio raises the Tier 2 rate because extra evidence has more predicted value; low bandwidth increases the use of local and Tier 1 actions. The inversion-difficulty indicator is a diagnostic of the synthetic feature model, not an empirical privacy-attack result.

Table 6: Synthetic privacy and resource results.

Profile	Tier-1 Rate (%)	Tier-2 Rate (%)	Packet Size (bytes)	Mean ϵ	Inversion Difficulty
Clean, 8 Mb/s	29.8	22.4	62.1	1.32	0.71
Noisy, 8 Mb/s	31.4	29.7	71.6	1.43	0.67
Clean, 1 Mb/s	24.6	18.1	53.8	1.08	0.76
Noisy, 1 Mb/s	29.2	23.6	64.3	1.29	0.70

In the clean 1 Mb/s profile, P-UAFO uses Tier 2 on only 18.1% of trials and keeps the mean packet size at 53.8 bytes. Under noisy 1 Mb/s conditions, the Tier 2 rate increases to 23.6%, yet the mean privacy expenditure remains lower than that of a fixed high-tier split. This asymmetric response is desirable: difficult audio should not automatically trigger maximal disclosure, while a modest protected packet can still be justified when it materially reduces expected decision risk.

5.3 Limitations

The current evidence is limited to generated score and resource events plus a reference host-side operation benchmark. It does not include speaker embeddings from a public corpus, short-utterance or cross-device trials, measurements on a phone or embedded board, or traces from a real wireless network. Consequently, the manuscript does not claim corpus-level accuracy, real-time operation, or demonstrated deployability. A full evaluation should report standard speaker-verification protocols, EER and minDCF, calibration error, device-specific energy and latency, network-state prediction error on measured traces, and

attack-based privacy tests. It should also consider spoofing, over-the-air manipulation, calibration drift, and speech enhancement under noisy conditions [22–24].

Feature-level differential privacy does not by itself provide complete biometric-template protection. It must be combined with an application-specific threat model, secure transport, server governance, and lifecycle controls for device and cloud models. In addition, repeated authentications require cumulative privacy accounting rather than a one-shot budget. The next empirical stage will compare local-only, fixed low/high split, always-cloud, uncertainty-only, and P-UAFO under identical public-corpus trials and measured mobile/network traces. These limitations motivate a corpus- and hardware-based follow-up study rather than stronger claims from the present synthetic profiles.

6 Conclusion

P-UAFO keeps raw speech local and selectively offloads protected intermediate features when uncertainty, resource state, and privacy budget justify cloud computation. The approach combines calibrated risk prediction, tiered feature protection, resource-aware selection, and confidence-weighted fusion. Its formal properties and reproducible simulations show why selective assistance can balance accuracy, latency, and privacy more effectively than local-only or fixed-split inference.

The present contribution is a formally analyzed and reproducible control framework, not a completed deployment study. Future work will evaluate real corpora and device traces, report EER, minDCF, and calibration metrics under matched baselines, include reconstruction, linkage, membership, and voice-cloning-oriented attacks, and extend the controller to repeated multi-edge requests with online calibration and cumulative privacy accounting.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Yongfeng Zhang and Jie Chen; methodology, Yongfeng Zhang; software, Yongfeng Zhang; validation, Yongfeng Zhang and Jie Chen; formal analysis, Yongfeng Zhang and Jie Chen; investigation, Yongfeng Zhang; resources, Yongfeng Zhang and Jie Chen; data curation, Yongfeng Zhang; writing—original draft preparation, Yongfeng Zhang; writing—review and editing, Yongfeng Zhang and Jie Chen; visualization, Yongfeng Zhang; supervision, Jie Chen; project administration, Jie Chen. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: Given his role as Youth Editorial Board Member of this journal, Jie Chen had no involvement in the peer review of this article and had no access to information regarding its peer review. Full responsibility for the editorial process for this article was delegated to another journal editor. The authors declare no conflicts of interest.

Nomenclature

Symbol	Definition
x	Input utterance held locally at the edge device.
(x^e, x^p)	Enrollment and probe utterances in one verification trial.
h	Edge intermediate feature before privacy processing.
z_e, z_c	Local and cloud-refined speaker embeddings.
s_e, s_c	Local and cloud verification scores.

u	Calibrated trial uncertainty in $[0, 1]$.
a	Inference action: 0 (local), 1 (low-tier), or 2 (high-tier).
P_a	Low-rank projection matrix for feature tier a.
Q_b	b-bit uniform quantizer.
$\tilde{h}_a$	Protected feature packet transmitted for action a.
C	Clipping radius of an edge feature.
σ_a	Standard deviation of Gaussian perturbation at tier a.
ϵ_a, δ	Differential-privacy parameters of a transmitted packet.
B	Predicted uplink throughput.
T_a, E_a	Expected latency and edge energy under action a.
R_a	Predicted verification risk under action a.
$\mathcal{U}_a$	Utility of action a.
w	Cloud-score fusion weight.
m	Signed verification margin.
$\lambda_T, \lambda_E, \lambda_P$	Weights for latency, energy, and privacy expenditure.
$\mathcal{A}$	Feasible action set after resource constraints.

References

1. Dehak N, Kenny PJ, Dehak R, Dumouchel P, Ouellet P. Front-end factor analysis for speaker verification. IEEE Trans Audio Speech Lang Process. 2011;19(4):788–98. doi:10.1109/tasl.2010.2064307.
2. Snyder D, Garcia-Romero D, Sell G, Povey D, Khudanpur S. X-vectors: robust DNN embeddings for speaker recognition. In: Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2018 Apr 15–20; Calgary, AB, USA. p. 5329–33. doi:10.1109/icassp.2018.8461375.
3. Chung JS, Nagrani A, Zisserman A. VoxCeleb2: deep speaker recognition. In: Proceedings of the Interspeech 2018; 2018 Sep 2–6; Hyderabad, India. p. 1086–90. doi:10.21437/interspeech.2018-1929.
4. Kang Y, Hauswald J, Gao C, Rovinski A, Mudge T, Mars J, et al. Neurosurgeon: collaborative intelligence between the cloud and mobile edge. In: Proceedings of the 22nd ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'17); 2017 Apr 8–12; Xi'an, China. p. 615–29. doi:10.1145/3037697.3037698.
5. Malik UM, Javed MA, Alvi AN, Alkhathami M. Reliable task offloading for 6G-based IoT applications. Comput Mater Contin. 2025;82(2):2255–74. doi:10.32604/cmc.2025.061254.
6. Quasim MT, Nisa KU, Husain MS, Aadam AIA, Sheraz MW, Khan MZ. Intelligent management of resources for smart edge computing in 5G heterogeneous networks using blockchain and deep learning. Comput Mater Contin. 2025;84(1):1169–87. doi:10.32604/cmc.2025.062989.
7. Mohapatra H. Task offloading and edge computing in IoT—gaps, challenges and future directions. Comput Mater Contin. 2026;87(3):1–10. doi:10.32604/cmc.2026.076726.
8. Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography. Berlin/Heidelberg, Germany: Springer; 2006. p. 265–84. doi:10.1007/11681878_14.
9. Hsu WN, Bolte B, Tsai YH, Lakhotia K, Salakhutdinov R, Mohamed A. HuBERT: self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans Audio Speech Lang Process. 2021;29:3451–60. doi:10.1109/taslp.2021.3122291.
10. Chen S, Wang C, Chen Z, Wu Y, Liu S, Chen Z, et al. WavLM: large-scale self-supervised pre-training for full stack speech processing. IEEE J Sel Top Signal Process. 2022;16(6):1505–18.
11. Liu B, Wang H, Qian Y. Towards lightweight speaker verification via adaptive neural network quantization. IEEE/ACM Trans Audio Speech Lang Process. 2024;32:3771–84. doi:10.1109/taslp.2024.3437237.
12. Tan F, Yu WH, Lin J, Un KF, Martins RP, Mak PI. A 1.8% FAR, 2 ms decision latency, 1.73 nJ/decision keywords-spotting (KWS) chip incorporating transfer-computing speaker verification, hybrid-IF-domain computing and scalable 5T-SRAM. IEEE J Solid State Circuits. 2025;60(3):1103–12. doi:10.1109/jssc.2024.3440506.

13. Huang W, Zhang Q, Liu T, Xu Y, Zhang D. Service function chain deployment algorithm based on multi-agent deep reinforcement learning. Comput Mater Contin. 2024;80(3):4875–93. doi:10.32604/cmc.2024.055622.
14. Angelopoulos AN, Bates S. Conformal prediction: a gentle introduction. Found Trends Mach Learn. 2023;16(4):494–591. doi:10.1561/2200000101.
15. Minderer M, Djolonga J, Romijnders R, Hubis F, Zhai X, Houlsby N, et al. Revisiting the calibration of modern neural networks. Adv Neural Inf Process Syst. 2021;34:15682–94. doi:10.48550/arxiv.2106.07998.
16. Altaibek M, Zulkhazhav A, Yergesh B, Bekmanova G, Omarbekova A, Sharipbay A. Privacy-preserving speaker verification in end-to-end encrypted chats: a feasibility study. IEEE Access. 2025;13:217556–72. doi:10.1109/access.2025.3648046.
17. Hu C, Hao Y, Zhang F, Luo X, Shen Y, Gao Y, et al. Privacy-preserving speaker verification via end-to-end secure representation learning. In: Proceedings of the Interspeech 2025; 2025 Aug 17–21. Rotterdam, The Netherlands. p. 1508–12. doi:10.21437/interspeech.2025-1096.
18. Chen D, Zhou Y, Wang X, Xiang S, Liu X, Sang Y. Res2Former: integrating Res2Net and transformer for a highly efficient speaker verification system. Electronics. 2025;14(12):2489. doi:10.3390/electronics14122489.
19. Chen Z, Wu S, Li X, Ai Z, Xu S. Open-set speaker identification through efficient few-shot tuning with speaker reciprocal points and unknown samples. IEEE Trans Audio, Speech Lang Process. 2025;33:3347–62. doi:10.1109/taslpro.2025.3587591.
20. Li Z, Mak MW, Pilanci M, Meng H. Mutual information-enhanced contrastive learning with margin for maximal speaker separability. IEEE Trans Audio, Speech Lang Process. 2025;33:2961–72. doi:10.1109/taslpro.2025.3583485.
21. Li Z, Mak MW, Chien JT, Pilanci M, Jin Z, Meng H. Disentangling speech representations learning with latent diffusion for speaker verification. IEEE Trans Audio, Speech Lang Process. 2025;33:3896–907. doi:10.1109/taslpro.2025.3610023.
22. Han S, Ahn Y, Shin JW. Noise-robust speaker verification with attenuated speech restoration and consistency training. IEEE Trans Audio, Speech Lang Process. 2025;33:1987–97. doi:10.1109/taslpro.2025.3567758.
23. Liu T, Truong DT, Kumar Das R, Aik Lee K, Li H. Nes2Net: a lightweight nested architecture for foundation model driven speech anti-spoofing. IEEE Trans Inform Forensic Secur. 2025;20:12005–18. doi:10.1109/tifs.2025.3626963.
24. Wang L, Lei X, He H, Wang L, Shi J, Wu Z. Over-the-air adversarial attacks and detection for automatic speaker verification. IEEE Trans Audio Speech Lang Process. 2026;34(11):1272–85. doi:10.1109/taslpro.2026.3651121.