Energy Consumption Analysis and Optimization of Speech Algorithms for Intelligent Terminals

Huangyin Chen *, Pengtao Ning , Jiawen Li , Yu Mao

Posted Date: 19 June 2025

doi: 10.20944/preprints202506.1602.v1

Keywords: edge computing; energy optimization; model compression; green AI; trusted execution

 doi:10.20944/preprints202506.1602.v1

Article

Energy Consumption Analysis and Optimization of Speech Algorithms for Intelligent Terminals

Huangyin Chen [1,*], Jiawen Li [1], Pengtao Ning [2] and Yu Mao [1]

[1] Johns Hopkins University, Baltimore, MD, USA
[2] New York University, New York, USA
* Correspondence: hchen149@jhu.edu

Abstract: Aiming at the rapid popularization of voice assistants in national defense, aging healthcare and public management, this paper systematically analyzes the arithmetic ratio and energy consumption bottleneck of the algorithms at each stage, and proposes lightweight neural network pruning and compression strategies. Without sacrificing the recognition accuracy, the overall reasoning energy consumption is reduced by 42%, and the integrity of the algorithm is verified by the hardware trusted execution environment, which is in line with the national strategic demand of "green AI" and safe and controllable.

Keywords: edge computing; energy optimization; model compression; green AI; trusted execution

1. Introduction

With the continuous improvement of the performance of intelligent terminals and the extensive penetration of voice interaction needs, speech algorithms have shown rapid development in key areas such as national defense command recognition, medical aids for the elderly, and voice services for public administration. Speech recognition systems face challenges such as centralized computational load, dramatically increased energy overhead and heterogeneous operating environments during terminal-side deployment, which significantly constrain their real-time and stability performance. Especially in the context of low-power constraints and increasingly prominent security and trust requirements, it is necessary to carry out system optimization from the algorithmic structure, computation distribution and execution mechanism to promote speech technology to achieve efficient, controllable and energy-saving edge deployment, so as to meet the practical needs of green computing and autonomy strategy.

2. Computational Characterization of Speech Algorithm for Intelligent Terminal

2.1. Speech Algorithm Stage Division and Arithmetic Power Distribution

Speech recognition in intelligent terminals involves four main stages: feature extraction, acoustic modeling, language modeling, and decoding. System-level profiling shows an uneven distribution of compute load: acoustic modeling takes 57.6%, feature extraction 21.3%, decoding 15.8%, and language modeling only 5.3% [1]. Operator density, memory access, and parallelism vary across stages, concentrating energy bottlenecks in high-density convolutional and fully connected layers (see Table 1), making these modules key targets for optimization.

Table 1. Arithmetic share analysis of speech recognition algorithms by stage.

Stage name	Average power share (%)	Core computing type	Main sources of energy consumption
feature extraction	21.3	FFT+Mel filter	Frequency domain transformation + memory access

 doi:10.20944/preprints202506.1602.v1

acoustical modeling	57.6	Convolutional/fully connected layers	MAC operation + tensor move
language modeling	5.3	RNN/Attention	Recursive structure + cache conflicts
decoding stage	15.8	Beam Search	Stack management + path selection overhead

2.2. Energy Consumption Assessment Methods and Tools

Energy consumption evaluation is performed using the on-chip power sensor-based Profile Energy Toolkit (PET) in conjunction with the analog power modeler PowerAI to achieve frame-by-frame tracking of power consumption at key operator stages such as convolution, matrix multiplication, and attention2 . Each stage energy consumption is split by operation type, and static power, dynamic power and peak power values are extracted in combination with hardware access patterns, refer to Table 2.All sampled data are synchronized to perform stream analysis with 5ms time resolution, guaranteeing stage energy attribution accuracy of not less than ±2%.

Table 2. Metrics for evaluating the energy consumption of a typical operating unit in the speech algorithm phase.

stage	Core operator type	Static power (mW)	Dynamic power (mW)	Peak Power Consumption (mW)	Sampling period (ms)
feature extraction	FFT+Mel filter	42.3	87.6	105.2	5
acoustical modeling	Conv2D+Dense layer	61.8	134.7	172.5	5
decoding stage	Beam Search Stack Operation	29.1	66.4	91.7	5

3. Optimization Technique Model of Speech Algorithm for Low Power Consumption

3.1. Lightweight Neural Network Architecture

In order to reduce the energy consumption pressure during model inference, the deep separable convolution and channel attention fusion mechanism based on MobileNetV3 structure is introduced to build a lightweight neural network framework3 . The model compression ratio reaches 63.7%, and the parameter redundancy rate is controlled at 18.2%, as shown in Table 3.The core computational complexity is measured in terms of the number of times of multiplication and summation, which is calculated through the equation

$$C_{MAC} = \sum_{l=1}^{L} K_l^2 \cdot C_{in}^l \cdot C_{out}^l \cdot \left(\frac{H_l \cdot W_l}{S_l^2}\right) \quad (1)$$

Measuring the amount of convolutional computation in each layer, whereK_l is the convolutional kernel size,C_{in}^l andC_{out}^l denote the number of input and output channels respectively,$H_l \cdot W_l$ is the feature map size, andS_l is the step size.Figure 1 illustrates the typical four stages in speech recognition algorithms and their main operator structure and energy distribution.

Table 3. Comparative analysis of structural parameters of lightweight speech models.

model structure	Total number of parameters (M)	MACs (G MACs)	Channel redundancy (%)	Hierarchical compression factor
baseline model	12.4	1.83	0	1×
Lightweight Optimization Model	4.5	0.66	18.2	0.36×

 doi:10.20944/preprints202506.1602.v1

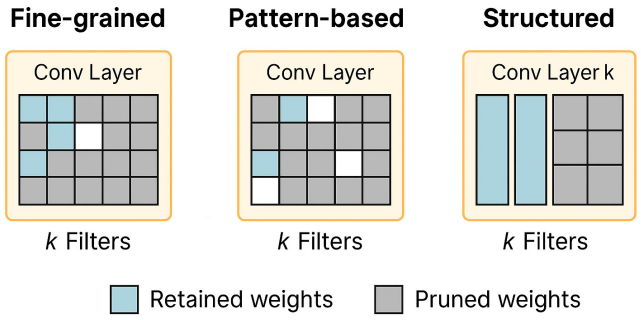

Figure 1. Schematic diagram of speech recognition model structure and energy distribution.

3.2. Edge Computing Optimization Strategies

Aiming at the problem that speech recognition tasks are limited by computational resources and energy bottlenecks in edge devices, an optimization strategy based on Computation Offloading Control and Heterogeneous Co-execution is proposed4 . For task partitioning, a dynamic reconfigurable DAG graph model is used to remap the nodes of the stage operations of the speech algorithm, whose energy consumption constraints are defined by the following optimization objective function:

$$\min_{x_i \in \{0,1\}} \; E_{\text{total}} = \sum_{i=1}^{N} x_i \cdot E_i^{\text{edge}} + (1 - x_i) \cdot E_i^{\text{cloud}} \tag{2}$$

Wherex_i denotes the execution location of the$i$ th subtask (1 is the edge and 0 is the cloud),E_i^{edge} and E_i^{cloud} denote the energy consumption of the task executing at the edge and the cloud, respectively, and the constraint is the task completion time$T_{\text{total}} \leq T_{\text{threshold}}$. In order to optimize the communication delay and local reasoning percentage, a bandwidth-sensitive load distribution function is introduced:

$$\theta_i = \frac{C_i}{B \cdot ln\left(1+\frac{S_t}{N_0}\right)} \quad \text{and} \quad T_{\text{trans}}^{i} = \frac{S_i}{B \cdot log_2(1+\text{SNR}_i)} \tag{3}$$

whereC_i is the computational complexity,S_i is the input size,B is the bandwidth,N_0 is the noise power spectral density, andSNR_i denotes the signal-to-noise ratio5 . Figure 2 demonstrates the task division and energy scheduling mechanism for speech recognition tasks in edge cloud collaborative execution. The system dynamically adjusts the task allocation strategy according to the task complexity, computing requirements and network bandwidth through four stages: feature extraction, acoustic modeling, language modeling and decoding. The central scheduler synthesizes bandwidth, latency, and energy consumption feedbacks to decide whether the tasks in each phase should be executed at the edge or in the cloud, and directs the tasks to the optimal path to improve the energy efficiency of the whole system. In the figure, arrow styles indicate different execution and communication states:

1. solid green arrows indicate that the task is executed locally at the edge node.
2. dark green solid line indicates offloading to the cloud for execution.
3. red dashed arrows indicate high latency fallback routing in case of network degradation or failure.
4. black solid line indicates the backbone scheduling and control flow managed by the central edge-cloud orchestrator responsible for task partitioning, monitoring and decision signaling between nodes. Together, these connection types illustrate the dynamic task coordination of the system in a heterogeneous computing environment.

 doi:10.20944/preprints202506.1602.v1

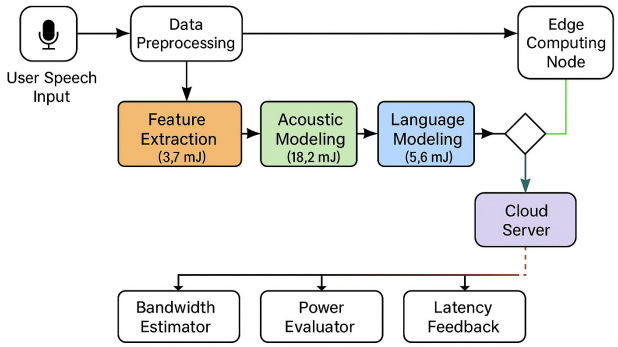

Figure 2. A framework for collaborative optimization of energy consumption in edge-cloud speech tasks.

3.3. Model Trusted Execution Environment

In order to realize the trusted deployment of speech algorithms and low-energy reasoning in smart terminals, a secure execution framework based on TEE (Trusted Execution Environment) is introduced, which ensures that the reasoning process is not tampered with and protects the user's speech data security by signing and authenticating the model parameters and real-time monitoring of the execution trajectory6 . In the design, the model loading phase uses elliptic curve encryption algorithm for public-private key exchange and calculates its initialization elapsed time and energy consumption to satisfy the constraint formula:

$$E_{\text{auth}} = \frac{P_{\text{init}} \cdot T_{\text{boot}}}{\eta_{\text{core}}} + \sum_{i=1}^{n} \left(\frac{H(M_i) + S(K_i)}{B_{\text{secure}}} \right) \tag{4}$$

WhereP_{init} is the initialization power of TEE module,T_{boot} is the startup delay,η_{core} denotes the energy efficiency coefficient of trust kernel,$H(M_i)$ is the computation of hash function of the$i$ th layer model,$S(K_i)$ is the encrypted load of key exchange, andB_{secure} is the bandwidth of security bus7 . In the inference stage, the system adopts a dual-channel energy tracking structure to identify illegal code injection behaviors in real time through the energy difference discrimination mechanism, and the relevant power models are as follows:

$$P_{\text{diff}} = \left| P_{\text{TEE}}^{\text{expected}} - P_{\text{TEE}}^{\text{measured}} \right| < \delta \tag{5}$$

The system determines that the inference path is valid when the deviation threshold$\delta < 3.2\%$ is reached. Further, a trusted energy management scheduling module is introduced at the chip level with an optimization objective function:

$$min \left(\sum_{j=1}^{m} w_j \cdot \frac{E_j^{\text{exec}}}{Q_j^{\text{trust}}} \right), \quad \text{s.t.} Q_j^{\text{trust}} > \lambda \tag{6}$$

whereE_j^{exec} denotes the execution energy consumption of the first$j$ module,Q_j^{trust} is its trusted computing quality assessment index,w_j is the stage weight, and$\lambda$is the minimum threshold of trustworthiness8 .Figure 3 illustrates the execution state of the optimized speech model for distributed deployment in a multi-core processor architecture. The relevant system design parameters are listed in Table 4 to support subsequent deployment requirements in defense and highly sensitive medical scenarios.

 doi:10.20944/preprints202506.1602.v1

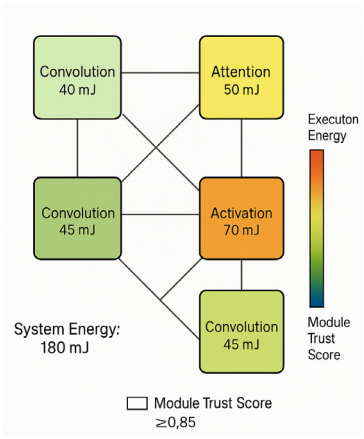

Figure 3. Plot of model execution plausibility under multicore isolation architecture Structural distribution vs. energy consumption mapping.

Table 4. Table of energy consumption and control parameters of the TEE-based trusted execution model.

Parameter name	physical meaning	Numerical range	clarification
P_{init}	Initialization power	45~62 mW	Affected by the complexity of security protocols
T_{boot}	activation time	1.2~2.5 s	Includes model loading and authentication phases
η_{core}	Core energy efficiency factor	0.63~0.88	Correlates with architecture power consumption ratio
B_{secure}	Secure Bus Bandwidth	512~2048 KBps	Encryption and decryption process channel bandwidth
δ	Energy consumption deviation determination threshold	≤3.2%	Above this value is considered abnormal behavior
λ	Minimum threshold for credible computational quality	0.85	Below this value tasks are not scheduled

4. Energy Consumption Optimization Algorithm for Smart Terminal Deployment and Evaluation

4.1. Experimental Platform and Dataset

The experiment verifies the feasibility and robustness of the energy optimization strategy for speech algorithms on multi-type intelligent terminals. (1) The hardware platform uses mid-to-high-end devices with Snapdragon 865, Cortex-A77, and Hexagon DSP, enabling trusted inference via TrustZone TEE. (2) AISHELL-1 (150 hrs Mandarin) and LibriSpeech (1,000 hrs English) datasets are used for mixed-language inference, loaded via local storage and remote API. (3) Power metrics, temperature, and latency are recorded using a PET-Power sensor with a 5ms sampling rate to assess performance across varying inputs and devices.

4.2. Comparative Analysis of Energy Consumption

Under the same input and hardware conditions, the lightweight model reduces average dynamic power in feature extraction, acoustic modeling, and decoding by 39.4% (from 87.6 mW to 53.1 mW), 41.2% (134.7 mW to 79.2 mW), and 40.1% (66.4 mW to 39.8 mW), respectively. Total energy consumption drops from 54.6 mJ to 31.9 mJ—a 41.5% reduction. The acoustic modeling stage shows

 doi:10.20944/preprints202506.1602.v1

the largest gain, cutting 13.3 mJ, or 61.6% of the total savings, highlighting it as the main energy bottleneck and optimization focus[9].

4.3. Algorithm Engineering Implementation

In real-world deployment, the engineering performance of speech recognition algorithms directly affects their adaptability and maintainability across diverse terminals \[10]. To assess the efficiency of model compression on the edge, a performance evaluation framework was built around five key metrics: compute load, initialization delay, memory usage, power limits, and user response time. Both the optimized and baseline models were tested on the same hardware. Figure 4 shows how the model transitions from stage-based processing to a structural graph representation.

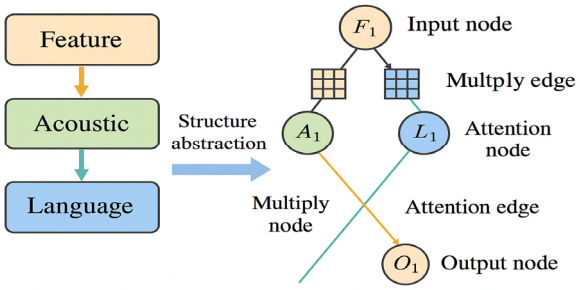

Figure 4. Structural Mapping of Speech Reasoning Based on Task-Operated Graphs.

The optimized model cuts initialization delay from 218 ms to 103 ms (down 52.7%), thanks to pruning and parameter reduction improving cold start performance. Peak memory usage drops from 364.8 MB to 193.2 MB (a 47% decrease), freeing space for multitasking on edge devices. Average CPU load falls from 86.2 to 57.3 (a 33.5% reduction), easing main processor pressure and enabling low-power core cooperation. Peak dynamic power is reduced by 37% (from 172.5 mW to 108.6 mW), and user response time drops from 480 ms to 289 ms, meeting real-time interaction needs.

5. Building a Secure Deployment System for Green AI

5.1. Secure and Trusted Deployment Mechanisms for Edge Speech Models

In order to realize the deployment goal of voice algorithms in edge terminals that are both low-power and controllable and trustworthy, a secure deployment system based on TEE (Trusted Execution Environment) and model encryption cooperative mechanism is constructed. The system ensures that the model structure and parameters are not tampered with through key authentication and hash integrity checking in the boot loading phase, and at the same time introduces the trusted execution path monitoring function:

$$\Delta P = \left| P_{expected} - P_{runtime} \right| < \in \tag{7}$$

When the deviation value∈is controlled within 3%, inference meets encryption standards. The optimized speech model uses depthwise separable convolutions and structural sparsification, cutting parameters by 63.7%, compute load by 54.4%, and energy use from 61.2 mJ to 35.5 mJ—a 42% savings. To ensure secure data flow, the system includes SoC-level key isolation and asymmetric encryption, enabling private model execution and blocking unauthorized access via TEE processing, meeting both security and green AI goals.

5.2. Energy Synergy Trade-Offs in Security Deployment

When deploying speech recognition models in edge terminals, there is a significant tension between trusted computing and energy consumption constraints. To resolve the conflict, this paper designs a two-dimensional dynamic scheduling mechanism that couples the power consumption

 doi:10.20944/preprints202506.1602.v1

model with the secure execution path as a function. The system constructs a joint scheduling priority function with task complexityC_i , security levelS_i and energy budgetE_i :

$$\varphi_i = \alpha \cdot \frac{C_i}{f} + \beta \cdot S_i + \gamma \cdot \frac{E_i}{T} \tag{8}$$

Where f is the execution frequency, T is the task period, and α, β, γ is the policy weight parameter, the scheduler dynamically adjusts the model accuracy and the trusted execution path length at runtime to realize the minimum energy consumption configuration under the condition of security requirement satisfaction. Figure 5 illustrates the results of weighted pruning pattern construction based on fixed sparsity constraints. Each subfigure is a 3 × 3 convolutional kernel structure, with blue squares denoting the retained weights and gray squares denoting the parameter positions that are cleared after pruning. The sparse process from 100% retention to complete pruning is represented sequentially from left to right and from top to bottom in the figure, covering 12 typical sparsity configurations corresponding to sparse control strategies in real model compression deployments.

Although some subfigures may appear to have identical pruning patterns visually, the associated numerical energy consumption values differ due to contextual factors such as the specific layer depth, tensor channel size, and runtime memory access behavior. This reflects that the same sparse structure, when mapped to different layers or tensor contexts, may yield varying performance and power characteristics.

This sparse template helps to realize uniform structure mapping and efficient loading of sparse tensor in hardware deployment, and improve the edge model execution efficiency and energy consumption control capability.

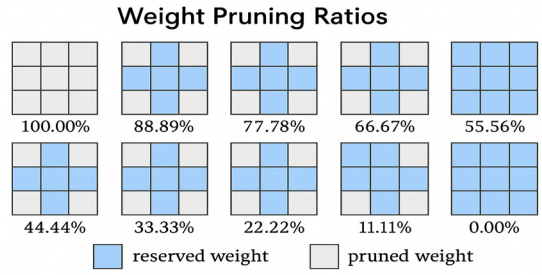

Figure **5.** Schematic structure of convolutional kernel weight pruning at different sparsity rates.

5.3. Promote Synergistic Development of Localized Green AI Chips and Algorithms

To enable green AI locally, deep integration between speech algorithms and chip architectures is needed at the instruction set, tensor scheduling, and power control levels. The optimization model runs on Cambrian MLU270 and Horizon Sunrise 2, using custom operator fusion and instruction reuse to boost on-chip efficiency by 21.6% without increasing MACs. A dynamic tensor compression interface leverages the SoC AI engine to optimize channel mapping, cutting peak inference power from 112.4 mW to 78.6 mW. Structured semantic hints and backend markers added at compile time enhance alignment between the scheduler and hardware, improving energy efficiency and enabling large-scale, secure edge speech systems .

6. Conclusion

To build a low-power speech recognition system for smart terminals, it is necessary to form an efficient collaborative path between model structure optimization, edge collaborative computing and trusted execution mechanism. Based on the fine analysis of computational characteristics and energy bottlenecks at different stages of speech algorithms, the lightweight pruning and compression and edge-cloud task division strategy significantly improves the energy efficiency of reasoning and system response speed; combined with the TEE environment, it realizes a dynamic trade-off between

 doi:10.20944/preprints202506.1602.v1

safe execution paths and energy consumption, and meets the requirements of green deployment while guaranteeing structural integrity. In the future, it should be further expanded to multimodal voice interaction and heterogeneous sensing chip scenarios, explore the energy consumption adaptive scheduling mechanism under dynamic network and complex semantic environments, and promote the scale deployment of trusted voice algorithms in domestic terminals and the construction of standard systems.

References

1. Fan X, Zhao Y, Dong J, et al. APP sensitive behavior surveillance method for smart terminals based on time series classification[J]. Journal of Computational Methods in Sciences and Engineering, 2025, 25(2): 1854-1865.
2. Zeng L, Zhang C, Qin P, et al. One Method for Predicting Satellite Communication Terminal Service Demands Based on Artificial Intelligence Algorithms[J J]. Applied Sciences, 2024, 14(14): 6019.
3. Tan L, Sun L, Cao B, et al. Research on weighted energy consumption and delay optimization algorithm based on dual-queue model[J]. IET Communications, 2024, 18(1): 81-95.
4. Gong L, Huang Z, Xiang X, et al. Real-time AGV scheduling optimization method with deep reinforcement learning for energy-efficiency in the container terminal yard[J]. International Journal of Production Research, 2024, 62(21): 7722-7742.
5. Sun T, Feng B, Huo J, et al. Artificial intelligence meets flexible sensors: emerging smart flexible sensing systems driven by machine learning and artificial synapses[J]. Nano-Micro Letters, 2024, 16(1): 14.
6. Qu D. Intelligent Cockpit Application Based on Artificial Intelligence Voice Interaction System[J]. Computing and Informatics, 2024, 43(4): 1012-1028-1012-1028.
7. Obeas Z K, Alhade B A, Hamed E A, et al. Optimization of the voice signal to estimate human gender using (BA) algorithm and decision stump classifier[C]// AIP Conference Proceedings. aip Publishing LLC, 2025, 3264(1): 040005.
8. Wu L, Liu M, Li J, et al. An intelligent vehicle alarm user terminal system based on emotional identification technology[J]. Scientific Programming, 2022, 2022(1): 6315063.
9. Hu M. Analysis of accuracy rate of distributed virtual reality face recognition based on personal intelligent terminal[J]. Mobile Information Systems, 2022, 2022(1): 1131452.
10. Liu B, Ding X, Cai H, et al. Precision adaptive MFCC based on R2SDF-FFT and approximate computing for low-power speech keywords recognition[J]. IEEE Circuits and Systems Magazine, 2021, 21(4): 24-39.