WIRELESS HEARABLES - PAPER REPORT

~Shreyas S (EP22B040)

Question 1. Summarize the paper, your learnings, and your overall impressions.

The paper proposes NeuralAids, a wireless hearable device for real-time speech enhancement that addresses strict hardware constraints in power (<100 mW), latency (<10 ms), and memory (1.5 MB). To meet these limits, the authors design a compact dual-path neural network that processes 6 ms audio chunks using a time-frequency (STFT) representation. The network alternates between a spectral stage for frequency modeling and a temporal stage for time dependencies, followed by a causal 2D deconvolution to reconstruct enhanced speech. To enable deployment on the low-power GAP9 accelerator, the model employs mixed-precision quantization, keeping the first convolution and final deconvolution layers in bfloat16 while quantizing all intermediate layers to int8. Performance is further improved through quantization-aware training (QAT) with the LSQ algorithm. Comparative evaluations show that NeuralAids achieves real-time inference (5.54 ms at 71.6 mW) and superior speech quality over TinyDenoiser, demonstrating the feasibility of running deep speech AI fully on wireless hearables.

Question 2. Discuss the main strengths and weaknesses of the work.

Strengths:

1. **Efficient caching mechanism:** The model reuses cached STFT frames and intermediate states from prior chunks, reducing redundant computation—a bold yet effective decision under tight on-chip memory limits.
2. **Lightweight recurrent units:** Replacing LSTMs with GRUs cuts computation and memory usage while retaining comparable temporal modelling ability.
3. **Quantization-Aware Training (QAT):** The use of QAT with LSQ fine-tuning ensures robustness to quantization errors, narrowing the performance gap with the floating-point baseline.
4. **Dataset diversity and fine-tuning:** Training with BRIRs, WHAM! noise, and simulated motion ensures robustness to real-world acoustic conditions and head movement.
5. **Comprehensive comparative analysis:** The paper benchmarks against TinyDenoiser across multiple quantization modes and metrics (SISDRi, PESQ, DNSMOS).

6. **Hardware-software co-design:** NeuralAids tightly couples model design with GAP9's architectural limits, achieving real-time operation within a 100mW budget.
7. **User study validation:** Subjective tests with 28 participants confirm tangible perceptual improvements, complementing quantitative metrics.

Weaknesses:

1. **Limited generalization scope:** Despite robust training, only single-microphone input was used, leaving multi-mic fusion unexplored.
2. **Continuous power assumption:** Reported power figures assume the AI accelerator runs continuously; duty-cycled or event-triggered operation was not demonstrated, limiting practical energy analysis.

Question 3. Compare NeuralAids with prior low-power speech enhancement systems such as TinyDenoiser.

NeuralAids achieves major gains over TinyDenoiser by redesigning both the architecture and quantization flow. It replaces TinyDenoiser's LSTM-based model (≈25 ms latency) with a dual-path GRU-based network that processes 6 ms audio chunks, achieving real-time (<10 ms) inference. Through frequency compression and efficient caching, it sustains higher throughput on the GAP9 accelerator.

TinyDenoiser uses post-training quantization, whereas NeuralAids applies mixed-precision QAT, reducing quantization loss and improving SISDRi from 5.97 dB to 8.19 dB. While power consumption rises to 71.6 mW (vs. 24 mW), NeuralAids offers far better perceptual quality (higher PESQ, DNSMOS, and MOS in user studies).

Overall, it trades a modest power increase for substantial improvements in audio fidelity, latency, and robustness.

Question 4. Algorithmic latency discrepancy in TinyDenoiser.

The 25 ms latency mentioned in Section 2.2.2 refers to the algorithmic latency of TinyDenoiser — the delay introduced by its internal architecture due to large chunk size, lookahead, and overlap-add processing. This is an inherent property of the model design and represents how long the network must wait before producing enhanced audio, making it unsuitable for real-time hearing-aid applications.

In contrast, Table 3 reports hardware runtime latency, i.e., the inference time per audio chunk when running the quantized TinyDenoiser on the GAP9 accelerator. This value (< 1 ms) is obtained during quantization evaluation, where various quantization configurations were profiled for runtime, power, and memory. Here, the network is heavily quantized (INT8/Mixed-Precision), and runtime refers only to the compute time for processing a 6 ms chunk — excluding algorithmic delays or buffering overheads.

TinyDenoiser therefore executes quickly on GAP9 but still cannot meet the real-time constraint imposed by its 25 ms algorithmic pipeline.

Question 5. Feasibility of integrating future AI algorithms under the 100 mW power budget.

- Run an always-on, tiny VAD/noise gate on the BLE SoC (which is already the clock/master for audio I/O) to monitor short windows (e.g., 6–12 ms) for SNR/noise thresholds. When the scene is quiet or stationary, gate off GAP9 and the external RAM via the existing load switches; when voice/noise crosses a threshold, wake the GAP9 cluster/NE16 and RAM.
- Apply DVFS/clock gating on GAP9 during brief idle spans inside “active” periods (FFT/iFFT on cluster vs. fabric-controller work), further saving dynamic power while maintaining the measured ~5.5 ms per-chunk runtime when needed.

Such hardware optimizations based on already available resources can provide avenues to increase computational complexity, while staying within the 100mW power budget.