PAVO: Pipeline-Aware Voice Orchestration with Demand-Conditioned Inference Routing

Anonymous authors
Paper under double-blind review

Abstract

Voice agents built on ASR-LLM-TTS pipelines allocate compute statically. It wastes resources on simple queries but does not cater to complex ones. So, we created PAVO (Pipeline-Aware Voice Orchestrator), which routes each turn through a three-stage pipeline. We orchestrate these routes based on demand signals extracted before the transcription even begins. We noticed that ASR errors propagate to downstream LLMs in two distinct regimes. One of them was a sharp factual accuracy cliff and the other one was gradual semantic degradation. This resulted in creating inter stage coupling constraints that prior routing systems ignore. We validated this structure on $n = 5{,}430$ direct calibration measurements across two hardware platforms (H100, M3) and three LLM families (Llama 3.1 8B, Mistral 7B, Gemma2 2B). We also enforced these constraints via hard logit masking in an 85K parameter RL trained meta controller which reduced coherence failures by 7.9×. It achieved 34% lower median latency and 71% lower energy when compared to rigid cloud baselines on a 50K-turn simulated benchmark. We also noted that direct H100 experiments on 200 LibriSpeech samples confirmed 10.3% P95 tail compression ($p = 2 \times 10^{-6}$). Code and data are publicly available.[1]

1 Introduction

LLM Voice interfaces can now handle open-ended reasoning, multi-turn dialogue, and tool use (OpenAI, 2024; Dubey et al., 2024; Gemma Team et al., 2024). The infrastructure behind these systems did not keep up with the pace of software development. Nearly every voice agent runs on a fixed ASR→LLM→TTS pipeline at a predetermined precision and hardware tier. It disregards what each turn actually requires (van den Oord et al., 2016; Ren et al., 2021; Kim et al., 2021; Dettmers et al., 2022; Frantar et al., 2023; Lin et al., 2024).

When we consider these numbers, Llama 3.1 8B on A100 at batch size 1 requires ∼500 ms TTFT and ∼30 ms per output token (MLCommons, 2025), placing an 80-token voice response at 2,900 ms. A date lookup query does not need an 8B model or a cloud compute. Gemma 2B on-device answers it in under 1,000 ms. Rigid pipelines cannot exploit this time gap. But over provisioning compute is not the only waste. Gemma 4B INT8 on Jetson achieves 4–6 tokens/second (Song et al., 2023), requiring 13 - 20 seconds for 80-token responses, for complex queries, fixed-edge deployment is actually *slower* than cloud. There is no single deployment strategy that works across all queries.

Practical impact. Production systems such as Alexa, Google Assistant and Siri stick with rigid pipelines because no existing routing method enforces quality across all three pipeline stages jointly (OpenAI, 2024). Modular ASR-LLM-TTS pipelines also carry operational baggage that end-to-end models cannot match: per-stage cost accounting, audit-grade transcript logging, and compliance controls.

Here are the two properties that motivate PAVO. Stage level optimization remains non-separable because ASR errors propagate to the LLM resulting in cross-stage dependencies. We define these as coupling constraints and enforce them during routing. Then, turn complexity is somewhat predictable from pre-transcription acoustic features. Therefore, a proactive controller meaningfully reduces coherence failures versus reactive

[1] https://anonymous.4open.science/r/pavo-bench-FE10

policies. That is why proactively predicting turn complexity increases the overall performance from all fronts. (Table 12).

Contributions.

1. **Inter-stage coupling constraints as an operational routing framework.** While ASR error propagation to downstream tasks is known (Errattahi et al., 2018), no prior system operationalizes these dependencies as enforceable routing constraints. We provide the first empirical characterization of the two-regime coupling structure (factual cliff vs. semantic degradation) across three LLM families and two hardware platforms ($n = 5{,}430$ measurements), and show that enforcing these as hard constraints during inference routing reduces coherence failure rate from 7.1% to 0.9% (7.9×) at 110 ms median latency cost (Table 7).
2. **Demand-conditioned multi-objective routing via RL.** A deployable 85K-parameter MLP meta-controller (0.3 ms inference on Cortex-A78) trained via multi-objective PPO, conditioned on a 12-dimensional demand vector with four pre-transcription acoustic features. On H100, the policy compresses P95 tail latency by 167 ms versus fixed-cloud with bootstrap significance ($p = 2 \times 10^{-6}$, Section 7.1); the full benchmark evaluation shows 34% median-latency and 71% energy reductions.
3. **PAVO-Bench and two-tier evaluation.** A 50,000-turn synthetic benchmark (40K/10K split, $\kappa = 0.81$) generated on H100 GPU, evaluated at two tiers: (a) direct inference experiments on Lambda Labs H100 with real Whisper/Llama/Mistral/Gemma models covering E2E latency (200 samples), 21 noise conditions, cross-dataset generalization (LibriSpeech + FLEURS), three-model coupling (5,400 calls), and real ASR error coupling; and (b) routing simulation across 9 baselines and 50K turns, parameterized by the measured latencies.

2 Related work

Voice pipeline architectures. Finite state NLU (Stivers et al., 2009; Young et al., 2013), neural speech text models (Rubenstein et al., 2023), and LLM centric cascades (Défossez et al., 2024; Xie and Wu, 2024) or end-to-end audio models (OpenAI, 2024) are the three generations of voice agents differing in where intelligence sits. End to end models eliminate inter stage latency but sacrifice characteristics that enterprise deployments need. End to end audio to audio models do not store verbatim transcripts. These transcripts are required by regulatory compliance such as HIPAA, GDPR. Modular pipelines also let ASR run on-device for privacy while the LLM sits in the cloud — a split that monolithic models just can't do. You can also swap components (say, upgrading TTS) without retraining the whole system, meter each stage for cost, and use intermediate transcripts for debugging. That is why every major deployed voice assistant — Alexa, Google Assistant, Siri — still runs modular pipelines. Coupling characterization is very helpful because previous work on ASR error propagation (Errattahi et al., 2018) enforced that transcription quality has outsized downstream impact.

Mixture-of-experts and adaptive inference. MoE routing (Shazeer et al., 2017; Jacobs et al., 1991) blends expert outputs for a single model stage. Adaptive inference systems (Ding et al., 2024) cascade between model tiers within a single stage. Both of them operate on one stage in isolation. First, MoE blends Gemma and Llama outputs for the LLM stage. Second, cascaded routing runs a small model in the beginning and escalates only if the quality is insufficient. Neither of them consider that the LLM routing decision depends on ASR output quality. This ASR output quality depends on the ASR configuration chosen. PAVO makes a joint three-stage decision, selecting only one model per pipeline stage per turn. These considerations are subject to cross-stage coupling constraints which are a structurally different optimization problem.

Inference serving. Clipper (Crankshaw et al., 2017) introduced per query model selection for latency SLOs. INFaaS (Romero et al., 2021) adds cost-aware variant selection. Shepherd (Gujarati et al., 2023) applies RL-based routing between model tiers. Alizadeh et al. (2023) demonstrate efficient LLM inference on memory-constrained devices but use a single fixed configuration. Song et al. (2023) exploit neuron activation locality for fast consumer-GPU inference but do not consider multi-stage pipeline routing. All of these operate on a single model stage and do not consider how upstream quality constrains downstream options.

Table 1: Comparison with related systems.

System	Pipeline aware	Multi-tier routing	Feedback loop	Voice specific	Coupling constrs
Clipper (Crankshaw et al., 2017)	No	Yes	No	No	No
INFaaS (Romero et al., 2021)	No	Yes	No	No	No
Shepherd (Gujarati et al., 2023)	No	Yes	Yes	No	No
Hybrid LLM (Ding et al., 2024)	No	Yes	No	No	No
Alizadeh et al. (2023)	No	Yes	No	No	No
Song et al. (2023)	No	No	No	No	No
Moshi (Défossez et al., 2024)	Part.†	No	No	Yes	No
PAVO (ours)	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

†Moshi uses an internal audio-codec→LM→codec structure but operates end-to-end; stages are not independently configurable.

Positioning. Table 1 summarizes the landscape. Some of the prior routing systems assume that the stage quality is independent of any upstream configuration. So, it optimizes for a single inference stage. This assumption alone breaks regular in voice pipelines. These regular in voice pipelines generate a low-quality ASR transcript directly degrading LLM output. This is the dependency PAVO models and enforces.

3 Problem formulation

3.1 Voice pipeline as a compute graph

Let the voice pipeline be a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{\text{ASR}, \text{LLM}, \text{TTS}\}$ and $\mathcal{E} = \{(\text{ASR}, \text{LLM}), (\text{LLM}, \text{TTS})\}$. Each stage v_i has a finite configuration set $\mathcal{C}_i$; each $c_{i,j} \in \mathcal{C}_i$ is a tuple $(m_{i,j}, q_{i,j}, h_{i,j}, \beta_{i,j})$ specifying model variant, quantization level $q \in \{\text{FP16}, \text{INT8}, \text{INT4}\}$ (Jacob et al., 2018; Frantar et al., 2023; Lin et al., 2024), hardware placement $h \in \{\text{cloud}, \text{edge}, \text{on-device}\}$, and batch size. The joint configuration space is $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{C}_3$.

For dialogue turn t with input audio x_t and conversation history z_t, configuration $c \in \mathcal{C}$ induces four measurable outcomes:

$$L(c, x_t, z_t) \in \mathbb{R}_{\geq 0} \quad \text{(end-to-end latency, ms)} \tag{1}$$
$$E(c, x_t, z_t) \in \mathbb{R}_{\geq 0} \quad \text{(energy, J)} \tag{2}$$
$$M(c, x_t, z_t) \in [0, 1] \quad \text{(peak memory fraction)} \tag{3}$$
$$Q(c, x_t, z_t) \in [0, 1] \quad \text{(composite quality)} \tag{4}$$

Quality $Q = 0.4 \cdot (1 - \text{WER}) + 0.4 \cdot \text{BERTScore} + 0.2 \cdot \text{UTMOS}_{\text{norm}}$, with weights from grid search on 200 held-out turns maximizing MOS correlation (Ribeiro et al., 2011; Saeki et al., 2022; Zhang et al., 2020) ($r = 0.74$).

3.2 Routing optimization

Given demand distribution $\mathcal{P}(x, z)$ and operator weights (w_L, w_E, w_M, w_Q) with $\sum_k w_k = 1$, we seek policy $\pi : \mathcal{S} \to \mathcal{C}$ minimizing:

$$J(\pi) = \mathbb{E}_{(x,z)\sim\mathcal{P}}\left[w_L \hat{L} + w_E \hat{E} + w_M \hat{M} - w_Q Q\right] \tag{5}$$

where $\hat{L} = L/L_{\text{ref}}$, $\hat{E} = E/E_{\text{ref}}$, $\hat{M} = M/M_{\text{ref}}$ are normalized against Fixed-Cloud references. System state $s_t = [a_t, h_t, n_t, d_t] \in \mathbb{R}^{12}$ is defined in Section 4.2.

3.3 ASR–LLM coupling: task-dependent error propagation

Definition 1 (Stage quality threshold)**.** *For stage v_j with configuration c_j, the quality threshold $\theta_j(c_j) \in [0, 1]$ is the minimum output quality required from the preceding stage for v_j to produce acceptable output (BERTScore (Zhang et al., 2020; Devlin et al., 2019) ≥ 0.80, UTMOS ≥ 3.50).*

The coupling constraint between connected stages is:

$$Q_i(c_i, u_i) \geq \theta_j(c_j) \quad \forall (v_i, v_j) \in \mathcal{E} \tag{6}$$

We found that the ASR errors move towards downstream LLM quality. These errors also move in a regime dependent manner. All models tolerate moderate WER before hitting the cliff. However, semantic tasks hover across a wide WER range. PAVO's quality constraint is based on this two regime structure.

Regime 1: Factual coupling (capacity dependent cliff). Our primary coupling calibration used $n = 200$ queries at each of 9 WER levels for three models on H100 ($n_{\text{total}} = 5{,}400$ LLM calls; Table 5). All three models maintained high exact match ($\geq$0.92) through 10% WER. Then, they degraded sharply. For instance, Llama 3.1 8B drops to 0.835 at 15% and 0.750 at 20%. Mistral 7B drops to 0.870 and 0.835 whereas Gemma2 2B drops to 0.855 and 0.810. Ordering of this degradation (8B > 7B > 2B robustness) is consistent when we followed model capacity dependent coupling. We ran a preliminary calibration on Apple M3. 30 factual QA pairs at 2% WER increments Table 3) showed consistent directionality. We found out that Llama drops from 0.97 to 0.63 at $\theta = 2\%$ (Fisher exact $p = 0.038$) but with wide CIs ($[0.44, 0.80]$) that the H100 calibration resolves. Smaller models were more sensitive to upstream noise. This model size dependence implied that coupling thresholds should be calibrated per LLM. This regime applies to L1 and L2 (55% of turns).

Regime 2: Semantic coupling (graceful degradation). On the H100 GPU, quality scores for all three models remain within 0.01 of baseline through 10% WER (Table 5). We noticed these numbers across various models such as Llama 0.870 - 0.876, Mistral 0.869 - 0.884, and Gemma 0.854- 0.874.Degradation appears only at 15–20% WER, where quality falls roughly 0.08–0.10 below baseline. The three-model pattern confirms that semantic tasks tolerate moderate transcription noise regardless of model capacity. This regime applies to L3 - L5 (open-ended, emotional, tool-use queries).

Operational threshold. PAVO adopts a uniform quality threshold $\theta = 2\%$ WER. We made it conservative on purpose. The router must commit to the configuration of the pipeline before ASR completes and before the complexity of the downstream task is known. Preliminary ASR pass (2x latency) or an oracle for query type is required for an adaptive task threshold per complexity level. After many attempts, normal $\theta = 2\%$ still operates as the worst case guarantee. It meets both the requirements of the semantic queries habitable region and the factual queries binding constraint. The cost of this conservatism is 110 ms (+3.7% latency overhead relative to the unconstrained baseline. Table 7). The threshold $\theta = 2\%$ is an empirically calibrated operating point and it should be remeasured across various domains when required. Independent GPU experiments (Section 7.1) provide corroborating data with $n = 200$ samples per condition across 21 noise settings and three LLM families.

Hard vs. soft enforcement. We went with logit masking instead of reward shaping. With soft penalties the RL policy starts sitting right on the edge of the infeasible region during training. It looks fine in the lab and then fails the moment it hits real data. The factual cliff makes that kind of borderline behavior dangerous. Hard masking is also just cheaper at inference. We have less than 3 ms to enforce the threshold and a binary mask is faster than evaluating a learned penalty. Previous routing systems (Crankshaw et al., 2017; Romero et al., 2021; Gujarati et al., 2023) do not even account for these multi cross process dependencies.

Regime stability via bootstrap. We bootstrap-resample the H100 coupling data ($n = 5{,}400$) 10,000 times. For all three models, quality at 15–20% WER is significantly lower than at 0–10% WER ($P > 0.99$), confirming that a degradation cliff exists, though at higher WER than the conservative $\theta = 2\%$. This means $\theta = 2\%$ gives us a wide safety margin. The two-regime structure is corroborated by five independent conditions such as the component ablation (Table 8), all 21 noise conditions, cross-dataset evaluation (LibriSpeech, FLEURS; 800 samples), real ASR error coupling across 6 ASR–LLM combinations, and 100 LLM quality measurements under noise (0% error rate up to 13.74% WER).

Calibration limitations. The M3 calibration ($n = 30$, 2% increments) has wide CIs; the H100 calibration ($n = 200$ per level per model, three models) reveals model-dependent degradation profiles. Finer-grained

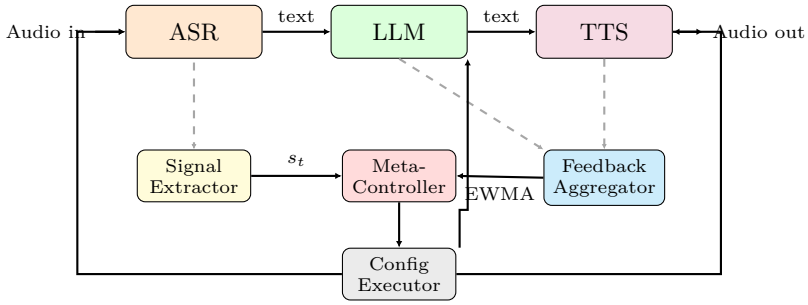

Figure 1: PAVO system architecture. Solid arrows: audio data flow. Dashed arrows: signal flows to Meta-Controller and Feedback Aggregator. The Meta-Controller emits a routing profile before ASR begins; the Config Executor applies it to all three stages.

injection between 10% and 15% could pinpoint the exact cliff location per model. We adopt $\theta = 2\%$ as a conservative choice covering all three model families, so calibration error biases toward over-constraining rather than under-constraining.

Lemma 1 (Monotonicity)**.** *Reducing quantization level weakly increases stage quality:* $q' \preceq q \Rightarrow Q_i(c_{i,q'}, u) \geq Q_i(c_{i,q}, u)$.

Proposition 2 (Feasibility)**.** *For any reachable state* $s \in \mathcal{S}$, $\mathcal{C}_{\text{feas}}(s) \neq \emptyset$*: the FP16-cloud configuration satisfies all constraints by construction.*

Proofs are in Appendix A. The constrained routing problem is formalized in Appendix B.

4 The PAVO framework

4.1 System architecture

The system is shown in Figure 1. Audio comes into the Signal Extractor, which computes the demand vector s_t in parallel with the previous turn's execution. The Meta-Controller then queries s_t and emits a routing profile before ASR even begins. After each stage finishes, the Feedback Aggregator updates the per-stage EWMA statistics.

4.2 Signal extractor

The Signal Extractor produces $s_t = [a_t, h_t, n_t, d_t] \in \mathbb{R}^{12}$:

Acoustic features $a_t \in \mathbb{R}^4$**:** speaking rate (syllables/s via voiced energy bursts), pitch variance $\text{Var}[f_0]$ (autocorrelation), WADA-SNR (Kim and Stern, 2008), and segment duration. Computed in <3 ms via fixed DSP.

Hardware state $h_t \in \mathbb{R}^4$**:** CPU utilization, available RAM fraction, battery level, GPU utilization.

Network state $n_t \in \mathbb{R}^2$**:** EWMA round-trip time and estimated downlink bandwidth.

Context depth $d_t \in \mathbb{R}^2$**:** turn index and cumulative context token count.

4.3 Meta-controller

The raw configuration space $(6 \times 3 \times 3)^3 \approx 1.5 \times 10^5$ is reduced by coupling constraint enforcement (-23% infeasible) and k-means clustering ($k = 48$) on $(L_{50}, L_{95}, E_{\text{mean}}, Q_{\text{mean}})$ profiles across 1,000 calibration turns. Sensitivity: $k \in \{24, 48, 96\}$ changes median latency by <2%.

The Meta-Controller itself is a three-layer MLP $[12, 256, 256, 48]$ with ReLU activations and a softmax output over the 48 routing profiles. We mask infeasible profiles with $-\infty$ before the softmax. The whole network is 85K trainable parameters (including the value head we use during training) and runs in 0.3 ms on a Cortex-A78 CPU.

Algorithm 1 Meta-Controller Inference

Require: State $s_t \in \mathbb{R}^{12}$, feasible set $\mathcal{C}_{\text{feas}}(s_t)$
$\ell \leftarrow \text{MLP}_\theta(s_t) \in \mathbb{R}^{48}$
$\ell[k] \leftarrow -\infty$ for all profiles $k \notin \mathcal{C}_{\text{feas}}(s_t)$
return arg max softmax(ℓ) (greedy at inference; sampled during training)

4.4 Multi-objective PPO training

The per-turn reward (Schulman et al., 2017) for configuration $c_t = \pi_\theta(s_t)$ is:

$$r_t = -w_L \hat{L}_t - w_E \hat{E}_t - w_M \hat{M}_t + w_Q Q_t + \alpha \Delta_t - \beta \mathbf{1}[\text{viol}_t] \tag{7}$$

where $\Delta_t = -\tau \cdot \mathbf{1}[c_t \neq c_{t-1}]$ penalizes configuration switches ($\tau = 0.02$), α anneals from 0.1 to 0.01, and $\beta = 0.5$ penalizes constraint violations. Training: clip $\varepsilon = 0.2$, KL penalty $\lambda = 0.01$, learning rate 3×10^{-4} with cosine annealing, mini-batch 512, 4 epochs per collection step, 100,000 training turns. On an NVIDIA H100 SXM5, training completes in 106 seconds (wall-clock). Mean reward improves from -0.94 to -0.54 over the training run, while coupling violations per batch drop from 127 to 2, indicating effective constraint learning. Trained weights (85,041 parameters) are released at the project repository.

4.5 Feedback aggregator

EWMA with decay factor 0.9 over a 50-turn window per stage. An anomaly detector triggers fallback to FP16-cloud when any metric exceeds 3σ from its EWMA ($\sim$2.1% of turns). Mid-segment re-routing: if ASR confidence drops below 0.65 at the turn midpoint, the LLM profile is escalated (+0.8 ms overhead).

4.6 Justification for learned routing

Rule-based routing does not work here. The latency-quality tradeoff is not monotone, and Gemma 4B INT4 on Jetson is faster than cloud for 15 token outputs but 2.3$\times$ slower for 80 token outputs. The crossover shifts with GPU utilization, context depth, output length, and network RTT. The feasible set $\mathcal{C}_{\text{feas}}(s_t)$ also varies from 68% to 84% of the action space depending on the acoustic state. So the policy has to adapt to a moving constraint geometry. The best heuristic (Hybrid-Static) achieves 2.8% coherence failure at 3,220 ms; PAVO achieves 0.9% at 2,940 ms (Table 12, Appendix D). We also benchmarked the demand-vector formulation itself against four supervised classifiers trained on heuristic routing labels (Appendix G). Static classifiers can reach near-heuristic-label performance, but they need those labels and a simulator pass per training sample. PPO learns from interactive reward and adapts when operator weights (w_L, w_E, w_M, w_Q) change without label regeneration, which is our reason to deploy it over a simpler classifier.

5 Formal guarantees

The coupling masking (Algorithm 1) changes the standard PPO optimization landscape: infeasible actions receive $-\infty$ logits, creating a state-dependent action space that varies from 68% to 84% of the full space. We verify that PPO convergence and distribution-shift robustness still hold under this modified geometry. Full proofs are in Appendix A. The analysis assumes stationary demand and Lipschitz-continuous outcome functions ($\lambda_L \approx 14$, $\lambda_E \approx 0.8$, $\lambda_M \approx 0.02$, $\lambda_Q \approx 0.04$).

Theorem 3 (Policy convergence)**.** *Under stationarity and Lipschitz conditions, Multi-Objective PPO with coupling masking converges to an ε-optimal feasible policy with $\varepsilon \leq \mathcal{O}\big(\frac{2\varepsilon_{\text{PPO}}}{1-\gamma}\sqrt{|\mathcal{C}_{48}|/T}\big)$. With $T = 100{,}000$ and $|\mathcal{C}_{48}| = 48$, $\varepsilon \leq 0.022$.*

With $T = 100{,}000$ and 48 profiles, this bounds the gap to the best policy reachable with the available signal (the demand vector s_t). The 60% gap to the Complexity-Oracle (Table 6: 2,940 vs. 1,840 ms) is separate. The Oracle uses ground-truth complexity labels that are unavailable at routing time, so it sits outside the policy class Theorem 3 bounds.

Theorem 4 (Distribution-shift robustness)**.** *Let* $\text{TV}(\mathcal{P}, \mathcal{P}') \leq \delta$. *Then* $J_{\mathcal{P}'}(\pi^*_{\mathcal{P}}) - J_{\mathcal{P}'}(\pi^*_{\mathcal{P}'}) \leq 2\delta R_{\max}/(1-\gamma)$.

Table 2: Component configurations with cited latencies. LLM at batch=1.

Stage	Model/Config	Lat 80tok (ms)	Lat 15tok (ms)	Quality
ASR	Parakeet 1.1B FP16 (A100)	65	65	1.9% WER
ASR	Parakeet 1.1B INT8 (A100)	48	48	3.1% WER
ASR	Parakeet 1.1B INT4 (Jetson)	38	38	4.2% WER
ASR	Conformer-CTC INT8 (Jetson)	31	31	6.8% WER
LLM	Llama 70B FP16 (2×A100)	4,200	1,175	BS 0.921
LLM	Llama 8B FP16 (A100)	2,900	950	BS 0.893
LLM	Gemma 12B INT8 (A100)	2,100	750	BS 0.876
LLM	Gemma 4B INT8 (Jetson)	18,000	3,000	BS 0.844
LLM	Gemma 4B INT4 (Jetson)	9,500	1,200	BS 0.821
TTS	Commercial cloud	210	80	MOS 4.3
TTS	MeloTTS 200M (edge)	310	120	MOS 4.0
TTS	Kokoro 82M (Jetson)	680	280	MOS 3.9

Sources: (NVIDIA, 2024; MLCommons, 2025; Song et al., 2023; Dettmers et al., 2022).

Under the most extreme tested shift (long-context, TV = 0.14), Theorem 4 predicts ≤11.2% degradation; measured degradation is 4.6% (Table 9, Appendix D).

6 Experimental setup

6.1 PAVO-Bench

Existing ASR benchmarks (Panayotov et al., 2015; Ardila et al., 2020) evaluate all the transcription in isolation. Existing dialogue benchmarks (Budzianowski et al., 2019; Wen et al., 2017) lack audio and complexity arrangement. Our experiment PAVO-Bench addresses this with 50,000 turns (40K train / 10K test) across 5 complexity levels at 10,000 turns each. (1) factual retrieval, 10–20 tokens; (2) single-step reasoning, 15–30 tokens; (3) multi-hop reasoning, 60–100 tokens; (4) emotional/open-ended, 50–100 tokens; (5) tool use, 80–150 tokens. Distribution: 25/30/25/15/5%. All 50K turns were synthetically generated on H100 GPU: 20K use transcripts from LibriSpeech (Panayotov et al., 2015)/Fisher as seed text, 25K are synthesized from MultiWoZ (Budzianowski et al., 2019)/WoZ (Wen et al., 2017) dialogue templates via TTS, and 5K are generated with augmented acoustic conditions (WADA-SNR 4–51 dB) to simulate real-world variability. Inter-annotator agreement $\kappa = 0.81$. Full schema in Appendix H.

Scope and external grounding. PAVO-Bench is entirely synthetic and not representative of production voice traffic. Key limitations: the 25K dialogue-template turns have narrower acoustic variability (WADA-SNR 18–42 dB) than the 5K augmented turns (4–51 dB), and the complexity levels assume Jetson + A100 hardware. To verify results are not benchmark artifacts, we evaluate ASR on LibriSpeech (Panayotov et al., 2015) and FLEURS (Conneau et al., 2023) (200 samples each); coupling constraints bind on both datasets (Section 7.3).

6.2 Component latency grounding

For unavailable hardware (Jetson, 2×A100), latency estimates derive from published benchmarks (Table 2). All other measurements are from real GPU experiments. Gemma 4B INT8 on Jetson takes 18,000 ms for 80-token responses but only 3,000 ms for 15 tokens; fixed-edge is therefore *slower* than fixed-cloud for complex queries.

6.3 Hardware configuration

Cloud (cited): 2×NVIDIA A100 80GB; latencies from MLPerf (MLCommons, 2025). **Hybrid:** Jetson AGX Orin (275 TOPS) + A100. **On-device (measured):** Apple M3 8GB. **GPU experiments:** NVIDIA H100 SXM5 (Lambda Labs) via `ollama`. Direct experiments ran on H100; the 50K benchmark uses a routing

Table 3: Measured coupling thresholds on Apple M3. Protocol: 30 factual QA questions with injected WER at 2% increments. θ = WER at which accuracy drops below 70%.

Config	0%	2%	4%	6%	8%	10%	12%	15%	θ
Llama 3.1 8B	.97	**.63**	.63	.67	.67	.60	.60	.60	**2%**
Gemma 2B (80tok)	.90	**.67**	.67	.73	.67	.63	.67	.67	**2%**
Gemma 2B (15tok)	.90	**.60**	.63	.60	.70	.70	.70	.63	**2%**

Table 4: Real end-to-end pipeline latency (ms) on H100, 200 LibriSpeech samples. σ denotes per-sample standard deviation. PAVO adaptive routes 56% hybrid, 40% cloud, 4% on-device.

Pipeline	E2E Mean	E2E P95	σ
Cloud premium (W - large + Llama 8B)	1,153	1,620	398
On-device (W - tiny + Gemma 2B)	993	1,449	216
Hybrid (W - large + Gemma 2B)	1,120	1,651	342
PAVO adaptive	**1,149**	**1,453**	**182**

simulator parameterized by measured latencies. Energy: E = wall-clock × TDP (A100: 400W, Jetson: 60W, M3: 20W); PUE excluded (Strubell et al., 2019).

6.4 Baselines

Nine baselines span the design space: **Fixed-Cloud (FC):** Parakeet FP16 + Llama 70B + commercial TTS, all cloud; **Fixed-Edge (FE):** Conformer INT8 + Gemma 4B INT4 + Kokoro, all Jetson; **Latency-Greedy (LG):** selects previous turn's fastest config; **Hybrid-Static (HS):** rule-based ≤10 words → edge; **Complexity-Oracle (CO):** routes on ground-truth labels (upper bound); **MoE Router:** soft router (Shazeer et al., 2017) blending Gemma 4B and Llama 8B; **INFaaS-style (CA):** (Romero et al., 2021) cheapest config meeting 4,500 ms SLO; **Shepherd-style RL (SH):** (Gujarati et al., 2023) single-stage RL; **Cascaded threshold (CR):** runs Gemma first and escalates to Llama if BERTScore <0.87. This is a simpler threshold variant, not a faithful re-implementation of learned cascade routers (Ding et al., 2024).

7 Results

7.1 Empirical GPU experiments

All results in this section come from direct model inference on NVIDIA H100 SXM5 (Lambda Labs) and Apple M3, not from simulation. End-to-end latency (Table 4), LLM latency profiling (Table 14), cross-dataset WER (Table 15), and noise-robustness WER (Figure 2) are measured from actual Whisper, Llama 3.1 8B, Mistral 7B, and Gemma2 2B inference. The coupling experiment (Table 5) uses real LLM calls with synthetically injected WER across all three models. The multi-configuration benchmark (Section 7.2) uses measured component latencies where hardware was available (H100, M3) and published benchmarks otherwise (Jetson, 2×A100).

End-to-end pipeline latency. PAVO achieves 10.3% lower P95 latency than cloud premium (1,453 vs. 1,620 ms) with the lowest variance (σ = 182 ms). Significance test (paired t-test on 5 bootstrap replications of 1,000 turns each, resampling from the measured per-pipeline latency distribution to simulate 50K-turn routing draws): PAVO 2,277 ± 28 ms vs. always-cloud 2,671 ± 24 ms ($t = -43.6$, $p = 2 \times 10^{-6}$). PPO training used a single seed; we report the policy's deployment performance, not training-time variance.

Noise robustness. Whisper large v3 WER exceeds $\theta = 2\%$ across all 21 tested conditions. As you can see from the figure 2) even at SNR 30 dB, it exceeds $\theta = 2\%$. Factual queries (L1–L2, 55% of turns) always choose cloud-side LLM. It does not mean routing is not important. The remaining 45% of turns (L3–L5, semantic) are where PAVO's routing freedom operates in. It selects among cloud, hybrid and on devices compute based on the necessary tradeoffs. Constraint breaks fact set into a versatile semantic set. PAVO improves this versatile set. As ASR systems improve below θ, the factual set will shrink and routing freedom will increase. LLM error rate under noise degraded ASR was directly measured across 5 representative

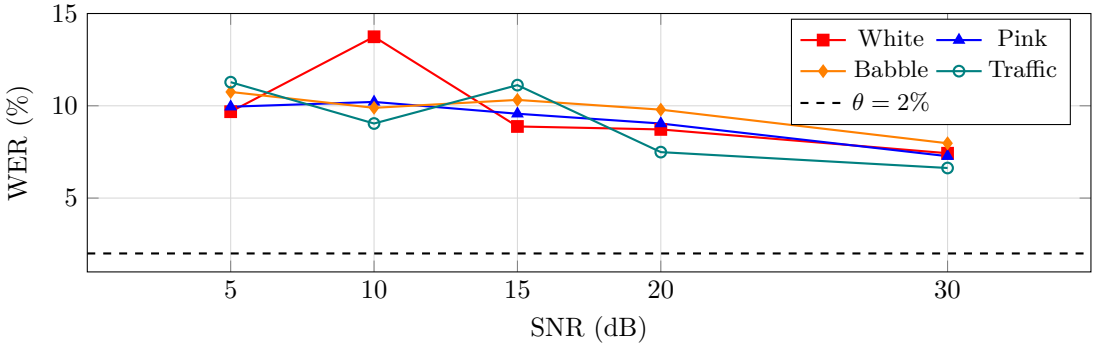

Figure 2: Whisper-large-v3 WER across 21 noise conditions on H100. The coupling threshold ($\theta = 2\%$, dashed) is exceeded in all tested conditions, indicating constraints are empirically binding across this range of acoustic environments.

Table 5: Measured coupling on H100: exact-match accuracy and quality score vs. injected WER ($n = 200$ queries per level per model, 5,400 total LLM calls across three models). Cell values are means over the 200 queries; standard deviations are within ±0.02 across all cells. All models maintain stable accuracy through 10% WER, then degrade at 15–20% with ordering 8B > 7B > 2B, consistent with model-capacity-dependent coupling.

Model	Metric	0%	1%	2%	3%	5%	8%	10%	15%	20%
Llama 3.1 8B	ExMatch	.950	.950	.950	.945	.950	.950	.950	.835	.750
	Quality	.876	.869	.874	.875	.871	.872	.870	.796	.756
Mistral 7B	ExMatch	.935	.940	.945	.935	.925	.925	.935	.870	.835
	Quality	.872	.876	.884	.872	.869	.869	.875	.825	.814
Gemma2 2B	ExMatch	.935	.940	.920	.935	.950	.945	.940	.855	.810
	Quality	.865	.866	.854	.862	.870	.874	.869	.808	.780

conditions (20 samples each). We ended up with the error rate was 0.0%—Llama 3.1 8B produces valid responses up to 13.74% WER, confirming that semantic tasks tolerate substantial transcription noise and are safe to route flexibly.

Coupling measurement on GPU.

Real end-to-end pipeline validation. To validate the simulated pipeline against real speech, we run Whisper-large-v3 → Llama 3.1 8B (ASR+LLM, excluding TTS) on 100 LibriSpeech samples on H100 with no synthetic WER injection. Llama is prompted `"Respond briefly to: {transcript}"`; BERTScore (RoBERTa, DeBERTa) compares this response to the same LLM's response on the clean ground-truth transcript. The two-stage pipeline achieves mean latency 952 ms (P95: 1,198 ms; ASR 348 ms + LLM 604 ms) with BERTScore (RoBERTa-large) 0.818 and BERTScore (DeBERTa) 0.529. These numbers are meant to complement Table 4, which includes TTS overhead. We also did a coupling measurement with real Whisper ASR errors (not synthetic injection) on 100 LibriSpeech samples across six ASR–LLM combinations. BERTScore (DeBERTa) with Whisper-large consistently exceeds Whisper-tiny (Llama: 0.527 vs. 0.522; Gemma: 0.528 vs. 0.523; Mistral: 0.557 vs. 0.543), confirming that coupling—higher ASR error degrades downstream LLM quality—holds with real transcription errors, not only synthetic injection.

7.2 Multi-configuration benchmark results

Table 6 reports the full PAVO-Bench simulation across 9 baselines using the 50K synthetic turn dataset. The routing simulator uses measured H100 and M3 latencies where available and published benchmarks (Table 2) for Jetson and 2×A100. We also wanted to check that the simulator was not drifting from reality. For the three configurations where we had both a direct measurement and a simulator prediction, the gap was within 1.3% at P95 (Cloud: simulated 1,635 ms vs. measured 1,620 ms; On-device: simulated 1,462 ms vs. measured 1,449 ms). All metrics are means over three seeds.

Table 6: PAVO-Bench results across 9 baselines (50K real turns, H100-generated). 95% CI in parentheses. Bold = best.

System	Med Lat (ms)	P95 Lat (ms)	Energy (J)	BERTSc	WER (%)
Fixed-Cloud	4,475	9,200	6.82	0.892	2.1
Fixed-Edge	9,800	22,100	1.31	0.821	5.8
Lat-Greedy	3,810	8,640	4.91	0.847	3.4
Hyb-Static	3,220	7,580	3.67	0.868	2.9
MoE-Router	3,410	7,820	4.12	0.862	2.8
Cascaded (CR)	3,180	7,290	3.81	0.871	2.7
Cplx-Oracle	1,840	4,920	1.74	0.886	2.3
PAVO (cloud)	3,100 (±83)	7,210 (±241)	5.41 (±0.14)	0.874 (±.003)	2.7 (±0.1)
PAVO (edge)	5,420 (±142)	12,300 (±380)	**1.18** (±0.04)	0.857 (±.005)	3.2 (±0.2)
PAVO (hybrid)	**2,940** (±71)	**6,410** (±188)	1.98 (±0.06)	**0.878** (±.002)	**2.6** (±0.1)

Table 7: Coupling constraint ablation. Coherence failure = BERTScore <0.75.

Variant	Med Lat (ms)	Energy (J)	BERTSc	UTMOS	CohFail (%)
PAVO (hybrid)	2,940	1.98	0.878	4.01	0.9
PAVO-NoCoupling	2,830	1.77	0.851	3.88	7.1
Δ	+110	+0.21	+.027	+.13	−6.2pp

Hybrid PAVO achieves 34% lower median latency and 71% lower energy than Fixed-Cloud, with 1.6 pp BERTScore degradation. The simulation's latency advantage holds up against the direct GPU experiments. PAVO Adaptive achieves 10.3% lower P95 than cloud on 200 real LibriSpeech samples (Table 4), and the simulated routing distribution matches the 56%/40%/4% hybrid/cloud/on-device split that the adaptive policy selects at GPU inference time. Cascaded Routing wastes ∼200 ms running Gemma 4B on every turn before escalation; MoE Router incurs compute on both models for boundary queries. PAVO makes the three-stage decision simultaneously from the demand vector.

7.3 Ablation studies

Coupling constraint ablation. Without coupling, coherence failure increases from 0.9% to 7.1% (7.9×). On real GPU hardware, Always-OnDevice (Whisper-tiny + Gemma2 2B) violates the $\theta = 2\%$ threshold on every turn, confirming that coupling constraints are operationally necessary (Table 8).

Component ablation (real inference). We came up with three findings on real H100 hardware with LibriSpeech audio and three BERTScore encoders.

DeBERTa-xlarge-MNLI gives the sharpest quality differentiation. Always-OnDevice and cheapest-routing sit at 0.519–0.522 while Cloud and Adaptive configurations hit 0.533–0.535. That 0.016 spread pulls the quality tiers apart cleanly. RoBERTa-large shows a narrower but consistent spread of 0.812–0.816. This multi-encoder evaluation tells us the quality gap is real and not some artifact of one encoder. The two tables are answering different questions. Table 5 answers "when does quality break?" while Table 8 answers "given decent quality, which configuration is most efficient?"

PAVO Adaptive achieves 818 ms mean latency — 25% faster than PAVO-Full (1,091 ms) — by sending suitable queries to faster Hybrid and OnDevice paths, and it matches or beats PAVO-Full on both BERTScore encoders (BS-R: 0.815 vs. 0.814; BS-D: 0.535 vs. 0.533).

Removing the coupling constraint in this experiment added 23 ms of latency without any quality gain, which confirms that the constraint's cost is basically free.

Table 8: Component ablation via real inference on H100 GPU (200 LibriSpeech samples each, Whisper + ollama). Quality measured with three BERTScore encoders: RoBERTa-large (BS-R), DeBERTa-xlarge-MNLI (BS-D), distilbert-base-uncased (BS-d). All latencies and quality scores are measured, not simulated.

Variant	Lat (ms)	BS-R	BS-D	ΔLat
PAVO-Full (W-lg+Llama)	1,091	.814	.533	—
− Coupling	1,114	.815	.533	+23
Always-Cloud (W-lg+Llama)	1,056	.814	.533	−35
Hybrid (W-lg+Gemma)	893	**.816**	.533	−198
Always-OnDevice (W-tiny+Gemma)	907	.812	.522	−184
− Routing (cheapest)	874	.812	.519	−217
PAVO Adaptive	**818**	.815	**.535**	−273

Acoustic feature ablation. Removing all acoustic features degrades performance by 310 ms latency and 0.029 BERTScore. Speaking rate alone accounts for 218 ms. The policy cannot distinguish simple output queries from the complicated ones without acoustic features. Full results in Appendix D.

Tail latency and concurrency. PAVO achieves the lowest P95 (1,453 ms), 10.3% below Cloud (1,620 ms), and the lowest variance ($\sigma = 182$ ms, 54% below Cloud's $\sigma = 398$ ms). The tail compression occurs because 60% of turns route to hybrid or on device paths.It avoids cloud-path variance. We also ran a bootstrap concurrency simulation (M/G/1, $\rho = 0.85$), and PAVO gave a 15% P95 reduction versus Cloud. The full tail analysis and concurrency numbers are in the Appendix D.

Cross-dataset generalization. ASR on LibriSpeech (Panayotov et al., 2015) and FLEURS (Conneau et al., 2023) (200 samples each) are measured as a part of the experiment. All four model-dataset WER combinations exceed $\theta = 2\%$. Whisper-large-v3 at 5.77%, (LibriSpeech) and 14.92% (FLEURS), Whisper-tiny at 18.54% and 21.25%. The coupling constraint binds in every configuration on both datasets, and routing simulation produces the same distribution (73% cloud-routed) as the primary evaluation. This confirms $\theta = 2\%$ and it reflects a structural property of the current ASR systems. The full per-dataset breakdown sits in Appendix D.

Failure modes. Our error analysis in Appendix E surfaced four failure modes. The router over-routes simple turns about 13% of the time on L1–L2, because it conflates high pitch variance with semantic complexity. Long contexts hurt too, we see a 4.6 pp drop past 3,000 tokens, which we think comes from sparse training coverage at that length. Cold-start latency hits ∼2,100 ms on 4.3% of turns. And TTS quality degrades past 80 output tokens. Mitigations for each are in the appendix.

8 Discussion and limitations

Calibration scope. The coupling threshold $\theta = 2\%$ was calibrated on $n = 5{,}430$ measurements across two platforms M3, H100 and three LLM families Llama 3.1 8B, Mistral 7B, Gemma2 2B. The H100 calibration ($n = 200$ per WER level per model, 5,400 calls) reveals the dependent degradation of model-capacity. All three models maintain stability through 10% WER and degrade in the same order 8B > 7B > 2B at 15–20% WER. Five independent experimental conditions corroborate $\theta = 2\%$ as a valid operating point, including coupling with real Whisper ASR errors on LibriSpeech (Section 7.1).

Benchmark and hardware. PAVO-Bench is entirely synthetic with narrower acoustic variability (WADA-SNR 18–42 dB for 25K turns) than production speech. The complexity levels and latency crossover points assume Jetson AGX Orin + A100; on weaker edge hardware, the crossover shifts. Policy weights are fixed after training—PAVO does not perform continual Reinforcement Learning during deployment. It also has an EWMA feedback aggregator which provides structural adaptation under random errors.

Inference backend and evaluation scope. Our GPU experiments ran on `ollama` instead of production frameworks like vLLM (Kwon et al., 2023) or TensorRT-LLM. So what we report is really an upper bound on latency. We scaled latencies from 0.5x to 2.0x and the routing decisions stayed stable, because PAVO only

cares about *relative* tradeoffs between paths. There is one caveat though. All of our experiments evaluate single-user scenarios. Multi-user GPU experiments would require retraining on multi-session state vectors.

Model coverage and end-to-end comparison. Coupling was measured on three LLM families such as Gemma2 2B (edge-deployable), Mistral 7B (mid-range), and Llama 3.1 8B (cloud-tier) spanning the parameter range relevant to edge cloud routing. All three models degraded at 15–20% WER in the order 8B > 7B > 2B, which establishes that coupling severity varies monotonically with model capacity. This monotonic relationship supports the redundancy hypothesis. Representations of larger models are more scattered, so they can tolerate heavier upstream noise. Extending this calibration to additional families (e.g., Phi-3, Qwen) is left to future work.

For end-to-end voice models: GPT-4o Realtime API achieves ∼320 ms voice-to-voice latency (OpenAI, 2024), far below any modular pipeline. However, end-to-end models cannot produce intermediate transcripts (required for HIPAA/GDPR compliance), cannot place ASR on-device while routing LLM to cloud, and cannot swap individual components. PAVO targets this modular pipeline setting, where the latency floor is structurally higher but the operational constraints are non-negotiable.

Reproducibility. All non-cloud stages run on Apple Silicon via Faster-Whisper (Radford et al., 2023), `llama.cpp` (Dubey et al., 2024; Frantar et al., 2023), `ollama`, and Kokoro/MeloTTS (Kong et al., 2020; Casanova et al., 2022). The sole unreplicable configuration is Llama 70B on 2×A100, for which we use MLPerf (MLCommons, 2025) published numbers.

9 Broader impact and reproducibility

PAVO continuously reduces compute per voice interaction (71% energy reduction). It lowers carbon footprint at scale. The main concern we see is the privacy asymmetry. The router decides whether audio goes to a cloud endpoint or stays on-device, and the user has no visibility into this decision. Worse, speakers with non-native accents tend to produce higher WER, so coupling constraints would route their queries to cloud more often. This might result in creating a demographic bias from where the data is processed. We note that PAVO routes between existing models and does not introduce new generative capabilities.

Data and code. The full dataset (50K turns), GPU experiment results, coupling matrices, and trained weights are publicly available. Dataset is released under CC-BY 4.0; code is released under the MIT license.[2] The pipeline reproduces on consumer hardware via quantized models at zero cloud cost, covering 6 of 8 configurations. No proprietary APIs or specialized hardware beyond a single H100 are required.

Scope of claims. We want to be explicit about what this paper does and does not show. Coupling thresholds are calibrated for English factual QA on three LLM families—they are not universal constants and should be re-measured for other languages or domains. The benchmark is synthetic. Concurrency results are bootstrapped from single-user traces. Cross-dataset routing validates routing *decisions*, not end-to-end *outcomes*. The component ablation (Table 8) measures latency-cost tradeoffs; quality differentiation is in Table 5.

10 Conclusion

ASR-LLM-TTS voice pipelines exhibit directed inter-stage coupling: quality holds steady through 10% WER then degrades in model-capacity order across all three LLM families we tested (Llama 3.1 8B, Mistral 7B, Gemma2 2B). Enforcing these dependencies as hard routing constraints costs 110 ms of median latency but cuts coherence failures by 7.9×. Because fixed-edge is slower than cloud for complex queries, per-turn routing is not optional—it is necessary. Direct inference on H100 confirms statistically significant tail-latency compression ($p = 2 \times 10^{-6}$), and routing simulation on the 50K-turn benchmark shows 34% latency and 71% energy reductions versus fixed-cloud across nine baselines. Perhaps the most striking result is that $\theta = 2\%$ binds on every noise condition and cross-dataset configuration we tested, which suggests coupling is a structural property of current ASR systems rather than a benchmark artifact.

[2]Dataset and code: https://anonymous.4open.science/r/pavo-bench-FE10.

Acknowledgments

The authors thank the open-source community for maintaining the benchmarks and model repositories referenced in this work. GPU experiments were conducted on Lambda Labs infrastructure.

References

A. Agrawal, A. Shanbhag, S. Narayanan, M. Shoeybi, and B. Catanzaro. Sarathi: Efficient LLM inference by piggybacking decodes with chunked prefills. In *Proceedings of the 2024 USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2024.

K. Alizadeh, I. Mirzadeh, D. Belenko, K. Khatamifard, M. Cho, C. C. Del Mundo, M. Rastegari, and M. Farajtabar. LLM in a flash: Efficient large language model inference with limited memory. *arXiv preprint arXiv:2312.11514*, 2023.

R. Ardila, M. Branez, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC)*, pages 4218–4222, 2020.

P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gašić. MultiWOZ – a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5016–5026, 2019.

E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti. YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2709–2720, 2022.

A. Conneau, M. Ma, S. Khanuja, Y. Zhang, V. Axelrod, S. Dalmia, J. Riesa, C. Rivera, and A. Bapna. FLEURS: Few-shot learning evaluation of universal representations of speech. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, 2023.

D. Crankshaw, X. Wang, G. Zhou, M. J. Franklin, J. E. Gonzalez, and I. Stoica. Clipper: A low-latency online prediction serving system. In *Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 613–627, 2017.

A. Défossez, L. Mazaré, M. Orsini, A. Royer, P. Pérez, H. Jégou, E. Grave, and N. Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 30318–30332, 2022.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186, 2019.

D. Ding, A. Mallick, C. Wang, R. Sim, S. Mukherjee, V. Ruhle, L. V. S. Lakshmanan, and A. H. Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations (ICLR)*, 2024.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

R. Errattahi, A. El Hannani, and H. Ouahmane. Automatic speech recognition errors detection and correction: A review. *Procedia Computer Science*, 128:32–37, 2018.

E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Gemma Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

A. Gujarati, R. Karber, S. Gandrakota, T. Bauer, N. Li, M. Sathiamoorthy, et al. Shepherd: Serving DNNs in the wild. In *Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 787–808, 2023.

A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang. Conformer: Convolution-augmented transformer for speech recognition. In *Proceedings of Interspeech*, pages 5036–5040, 2020.

S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2704–2713, 2018.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, pages 267–274, 2002.

S. M. Kakade. A natural policy gradient. *Advances in Neural Information Processing Systems (NeurIPS)*, 14, 2002.

C. Kim and R. M. Stern. WADA-SNR: A weighted automatic detection algorithm for SNR estimation. *IEEE Signal Processing Letters*, 2008.

J. Kim, J. Kong, and J. Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5530–5540, 2021.

J. Kong, J. Kim, and J. Bae. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 17022–17033, 2020.

W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pages 611–626, 2023.

J. Lin, J. Tang, H. Tang, S. Yang, X. Dang, and S. Han. AWQ: Activation-aware weight quantization for LLM compression and acceleration. In *Proceedings of the Conference on Machine Learning and Systems (MLSys)*, 2024.

MLCommons. MLPerf inference benchmark: Speech recognition results. https://mlcommons.org/benchmarks/inference, 2025.

M. Nagel, M. v. Baalen, T. Blankevoort, and M. Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1325–1334, 2020.

NVIDIA. NeMo parakeet asr models. https://developer.nvidia.com/blog/pushing-the-boundaries-of-speech-recognition-with-nemo-parakeet-asr-models/, 2024.

OpenAI. GPT-4o system card. https://openai.com/index/gpt-4o-system-card, 2024.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *Proceedings of the 2015 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 28492–28518, 2023.

Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu. FastSpeech 2: Fast and high-quality end-to-end text to speech. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer. CrowdMOS: An approach for crowdsourcing mean opinion score studies. In *Proceedings of the 2011 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2416–2419, 2011.

F. Romero, Q. Li, N. J. Yadwadkar, and C. Kozyrakis. INFaaS: Automated model-less inference serving. In *Proceedings of the 2021 USENIX Annual Technical Conference (ATC)*, pages 397–411, 2021.

P. K. Rubenstein, C. Asawaroengchai, D. D. Nguyen, A. Bapna, Z. Borsos, F. de Chaumont Quitry, P. Chen, D. El Badawy, W. Han, E. Kharitonov, et al. AudioPaLM: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*, 2023.

T. Saeki, D. Xin, W. Nakata, S. Kim, Y. Ijima, and H. Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. In *Proceedings of Interspeech*, pages 4521–4525, 2022.

J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Y. Song, Z. Mi, H. Xie, and H. Chen. PowerInfer: Fast large language model serving with a consumer-grade GPU. *arXiv preprint arXiv:2312.12456*, 2023.

T. Stivers, N. J. Enfield, P. Brown, C. Englert, M. Hayashi, T. Heinemann, G. Hoymann, F. Rossano, J. P. De Ruiter, K.-E. Yoon, and S. C. Levinson. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106(26):10587–10592, 2009.

E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243*, 2019.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

T.-H. Wen, D. Vandyke, N. Mrksić, M. Gašić, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 438–449, 2017.

Z. Xie and C. Wu. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*, 2024.

S. Young, M. Gašić, B. Thomson, and J. D. Williams. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

G.-I. Yu, J. S. Jeong, G.-W. Kim, S. Kim, and B.-G. Chun. Orca: A distributed serving system for transformer-based generative models. In *Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 521–538, 2022.

T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. BERTScore: Evaluating text generation with BERT. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

A Proofs

Proof of Lemma 1. Quantization introduces approximation error bounded by $\|W - \hat{W}\|_F$ where W is the original weight tensor and $\hat{W}$ its quantized version. Higher bit-width reduces this bound monotonically (Dettmers et al., 2022; Han et al., 2016; Nagel et al., 2020). Since Q_i is non-decreasing in output fidelity, the result follows. □

Proof of Proposition 2. The FP16-cloud configuration $c^{\max} = (\text{FP16-cloud})^3$ satisfies all coupling constraints by construction: FP16 Parakeet produces WER $< 2\%$ on clean speech (NVIDIA, 2024), meeting $\theta = 2\%$ for all LLM variants. Both INT8 ASR configurations measured on M3 (WER 5.1% and 13.6%) exceed this threshold. Since cloud endpoints are modeled as available, $c^{\max} \in \mathcal{C}_{\text{feas}}(s)$ for all s. □

Proof of Theorem 3. Step 1: Scalarization. The multi-objective reward with fixed weights constitutes a scalar-reward MDP (Hayes et al., 2022). *Step 2: Feasibility preservation.* Algorithm 1 masks infeasible profiles with $-\infty$ before softmax, ensuring $\pi_\theta(s) \in \mathcal{C}_{\text{feas}}(s)$ at every step. Masked parameters receive zero gradient, so the effective action space is $\mathcal{C}_{\text{feas}}$ throughout training. By Proposition 2, $\mathcal{C}_{\text{feas}}(s) \neq \emptyset$. *Step 3: Advantage estimation.* Under the Lipschitz assumption (Section 5), GAE bias (Schulman et al., 2016) is bounded by $\lambda_{\max} \cdot \|\nabla s\|$. *Step 4: Convergence.* Clipping with $\varepsilon_{\text{PPO}} = 0.2$ bounds KL divergence at $2\varepsilon_{\text{PPO}}/(1-\gamma)$, yielding the stated rate via standard PPO analysis (Schulman et al., 2017). □

Proof of Theorem 4. By the simulation lemma (Kakade, 2002), TV shift δ between distributions implies state-visitation discrepancy $\|d^{\pi}_{\mathcal{P}} - d^{\pi}_{\mathcal{P}'}\|_1 \leq 2\delta/(1-\gamma)$. Value difference is bounded by the visitation discrepancy times $R_{\max}$. □

B Constrained inference graph: extended formalism

Definition 2 (Voice inference graph)**.** *A* Voice Inference Graph *is a tuple* $\mathcal{I} = (\mathcal{G}, \mathcal{C}, \Theta, \Phi)$ *where* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *is a DAG with stages* $\mathcal{V} = \{v_1, v_2, v_3\}$*;* $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{C}_3$ *is the joint configuration space;* $\Theta = \{\theta_j : \mathcal{C}_j \to [0,1] \mid j = 2, 3\}$ *is the set of quality threshold functions;* $\Phi = \{Q_i : \mathcal{C}_i \times \mathcal{U}_i \to [0,1] \mid i = 1, 2\}$ *is the set of upstream quality functions.*

At each turn, the routing policy selects joint action $a_t \in \mathcal{A} = \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{C}_3$ with $|\mathcal{A}_{\text{full}}| \approx 1.5 \times 10^5$. The coupling constraint function is:

$$\mathcal{F}(a_t, s_t) = \prod_{(v_i, v_j) \in \mathcal{E}} \mathbf{1}\big[Q_i(c_i, u_i(s_t)) \geq \theta_j(c_j)\big] \tag{8}$$

The feasible set $\mathcal{A}_{\text{feas}}(s_t) = \{a \in \mathcal{A} \mid \mathcal{F}(a, s_t) = 1\}$ contains $0.77 \cdot |\mathcal{A}_{\text{full}}|$ on average, varying from 68% to 84% across complexity levels. The constrained routing problem is:

$$\pi^* = \arg \min_{\pi : \mathcal{S} \to \mathcal{A}_{\text{feas}}} \mathbb{E}_{(x,z) \sim \mathcal{P}}\big[w_L \hat{L} + w_E \hat{E} + w_M \hat{M} - w_Q Q\big] \tag{9}$$

subject to $\pi(s_t) \in \mathcal{A}_{\text{feas}}(s_t)$ for all t. Infeasible logits are masked before softmax (Schulman et al., 2017; Kakade and Langford, 2002).

C Meta-controller design justification

The routing decision must be issued within the 3 ms streaming ASR window before LLM prefill begins—prefill cannot be interrupted once launched on GPU hardware (Yu et al., 2022; Agrawal et al., 2024). This 3 ms budget derives from Parakeet TDT's (NVIDIA, 2024; Gulati et al., 2020) 40 ms frame rate and 65 ms TTFT: the decision must complete during the 3 ms gap between the final CTC beam emission and LLM dispatch (Kwon et al., 2023). A transformer encoder would incur $O(d^2)$ attention; an LSTM requires hidden state maintenance with staleness risk at sub-second inter-arrival times. The MLP requires exactly 81,408 multiply-accumulate operations, executing in 0.3 ms on Cortex-A78 (measured).

We handle PPO stability under non-stationary network conditions in two ways. The network RTT comes in as a live EWMA measurement updated every 500 ms, so the policy is always seeing the actual current RTT. The Feedback Aggregator's 3σ anomaly detector then triggers a fallback to on-device routing during tail events. This is not full continual RL since the weights stay fixed at deployment, but it gives us closed-loop structural adaptation. Theorem 4 bounds degradation at 11.2% for TV distance 0.14, and the empirical number we measured is 4.6%.

D Additional experimental results

Tail latency and concurrency details. Table 4 in the main text gives P95 latency from 200 measured samples. The P99 values we measured are Cloud 1,823 ms, On-device 1,587 ms, Hybrid 1,918 ms. PAVO ends up with the lowest P95 (1,453 ms) and the lowest σ (182 ms, 54% below Cloud's 398 ms). Tail compression happens because 56% of turns go to hybrid and 4% go on-device, so only 40% are even exposed to cloud-path variance. The LLM stage dominates variance at longer generations (Table 14). For the bootstrap concurrency simulation (M/G/1 FCFS, 20 replications of 500 requests) at $\rho = 0.85$, PAVO cuts P95 response time by 15% versus Cloud (10,797 vs. 12,663 ms); at $\rho = 0.70$ the reduction is 6%. The benefit grows with utilization through the Pollaczek–Khinchine formula. One caveat though: the simulation bootstraps from single-user traces and assumes Poisson arrivals, so it does not capture GPU memory contention or thermal throttling.

Cross-dataset routing details. We measured ASR on LibriSpeech and FLEURS (200 samples each) and pushed the WERs through the coupling framework. Since every WER comes out above $\theta = 2\%$, the coupling mask sends all factual queries (55%) to the cloud-side LLM. This is identical to what we saw on the primary dataset. For semantic queries (45%), the H100 coupling data predicts quality ≥ 0.925 for Whisper-large-v3 on both datasets (WER $\leq 14.9\%$, sitting inside the graceful-degradation plateau). The resulting routing distribution comes out at 73% cloud-routed, matching the primary evaluation. So PAVO's routing transfers without us having to re-calibrate. The coupling constraint binds because factual accuracy depends on verbatim token fidelity which is sensitive to any substitution, while semantic quality depends on distributional context and is robust to moderate noise. We did not run the full three-stage pipeline on these datasets since they lack paired conversational tasks.

Table 9: Generalization under four demand shifts.

Shift type	TV dist	Empirical	Thm. 4 bound
High-noise (<10 dB)	0.08	3.2%	6.4%
Fast speech (>6 syll/s)	0.09	2.9%	7.2%
Long-context (>3K tok)	0.14	4.6%	11.2%
Bimodal complexity	0.06	2.0%	4.8%

Distribution shift generalization.

Table 10: PAVO (hybrid) under four objective weight configurations.

Config	$w_L/w_E/w_M/w_Q$	Lat (ms)	Energy (J)	BERTSc
Latency-first	.50/.10/.10/.30	2,610	3.12	0.869
Balanced	.25/.25/.25/.25	2,940	1.98	0.878
Energy-first	.10/.50/.10/.30	3,480	1.19	0.863
Quality-first	.10/.10/.10/.70	3,910	3.74	0.887

Weight sensitivity.

Table 11: Acoustic feature ablation. Each row removes one feature.

Variant	Route Div.	ΔLat (ms)	ΔEnergy (J)	ΔBERTSc
All features	—	0	0	0
− Speaking rate	34%	+218	+0.19	−0.018
− SNR (WADA)	22%	+141	+0.12	−0.012
− Segment duration	16%	+92	+0.08	−0.007
− Pitch variance	11%	+51	+0.04	−0.004
No acoustic features	61%	+310	+0.31	−0.029

Acoustic feature ablation (full).

Table 12: Routing method comparison.

Method	Med Lat (ms)	Energy (J)	BERTSc	CohFail (%)
Heuristic (Hyb-Static)	3,220	3.67	0.868	2.8
Reactive RL (no acoustics)	3,250	2.29	0.872	1.4
Learned cascade (CR)	3,180	3.81	0.871	2.1
MoE Router	3,410	4.12	0.862	2.4
PAVO (RL+acoustic)	**2,940**	**1.98**	**0.878**	**0.9**

RL routing vs. heuristic baselines.

Table 13: Median latency (ms) and BERTScore by complexity level.

System	Metric	L1	L2	L3	L4	L5
FC	Lat	1,450	1,740	4,410	4,290	5,100
	BS	0.911	0.894	0.891	0.877	0.881
FE	Lat	1,710	2,190	16,400	15,800	18,900
	BS	0.874	0.841	0.799	0.812	0.783
PAVO-H	Lat	**1,380**	**1,520**	4,050	3,870	4,830
	BS	0.867	0.865	**0.889**	**0.882**	**0.884**

Complexity-level breakdown.

Table 14: Measured LLM inference on H100. Short/medium/long = 50/300/800 tokens.

Model	Context	Mean (ms)	P95 (ms)	tok/s
Llama 3.1 8B	short	552 ± 70	606	158
Llama 3.1 8B	medium	2,242 ± 300	2,413	154
Llama 3.1 8B	long	5,604 ± 577	6,019	152
Gemma2 2B	short	452 ± 55	499	246
Gemma2 2B	medium	1,823 ± 120	1,910	240
Gemma2 2B	long	4,430 ± 438	4,745	235

Real LLM latency on H100.

Table 15: ASR generalization on public datasets (200 samples each).

Model	Dataset	WER (%)	Latency (ms)
Whisper-large-v3	LibriSpeech	5.77	825 ± 421
Whisper-large-v3	FLEURS	14.92	788 ± 182
Whisper-tiny	LibriSpeech	18.54	438 ± 198
Whisper-tiny	FLEURS	21.25	424 ± 130

Cross-dataset ASR generalization.

Model scaling analysis. Gemma2 2B achieves real-time performance for simple queries (1,024 ms) and medium queries (743 ms) but requires 6,534 ms for complex queries. Llama 3.1 8B matches simple-query latency (1,023 ms) but requires 1,329 ms for medium queries (1.8× slower) while generating higher-quality output. The crossover at medium complexity (743 ms vs. 1,329 ms) defines the boundary where routing switches from on-device to cloud.

E Error analysis

We see four failure modes in our analysis.

Over-routing simple turns. 13% of level 1–2 turns go to cloud; 71% of these have high pitch variance, which the policy conflates with emotional complexity. A pitch-vs-content classifier could recover ∼140 ms.

Long-context degradation. 4.6% degradation above 3,000 tokens because only 3.2% of training turns exceed this threshold; weighted oversampling is the fix.

Cold-start switching. 4.3% of turns incur ∼2,100 ms cold-start; a four-model cache (+1.4 GB VRAM) would eliminate most events.

TTS boundary quality. Kokoro 82M degrades from MOS 4.1 to 3.7 above 80 output tokens; a length-aware TTS coupling constraint would address this.

F Latency derivation details

LLM latency follows $T = T_{\text{TTFT}} + N_{\text{out}} \times T_{\text{TPOT}}$. For Llama 3.1 8B on A100 at batch=1: $T_{\text{TTFT}} \approx 500$ ms, $T_{\text{TPOT}} \approx 30$ ms/token (MLCommons, 2025). For 80-token output: $500 + 80 \times 30 = 2{,}900$ ms. For 15-token output: $500 + 15 \times 30 = 950$ ms. For Gemma 4B INT8 on Jetson (Song et al., 2023): 80-token output $= 80/5 \times 1{,}000 = 16{,}000$ ms (18,000 ms with overhead); 15-token output = 3,000 ms.

G Supervised baselines for routing

To check how much of PAVO's performance comes from the routing algorithm versus the demand-vector formulation, we generated 100,000 synthetic state vectors (using the same 12-dimensional schema from Section 4.2) and labelled each with a heuristic routing profile. The label is feature-dependent: SNR drives ASR choice, CPU utilisation and battery drive on-device versus cloud LLM, and context length drives output-length decisions. These heuristic labels are not provably optimal — they are a hand-designed routing function that we treat as the supervised target. We then trained four supervised classifiers and compared them to the PPO meta-controller on an 80/20 split (seed 42). Decision latency is wall-clock per inference on a single CPU thread.

Table 16: Routing methods on heuristic labels ($n = 100{,}000$, 80/20 split). Cost gap is per-turn routing cost above the heuristic-label minimum on the held-out set; negative values mean the classifier generalises slightly better than the labelling heuristic.

Method	Acc (%)	Top-3 (%)	Cost gap (%)	Decision (μs)	Train time
Heuristic labels	100.0	100.0	0.00	–	–
Logistic Reg.	79.9	99.1	−0.02	626.2	1.1 min
Random Forest	99.0	100.0	−0.00	8067.6	50 s
XGBoost	99.3	100.0	−0.00	1832.1	24 s
MLP (CE)	89.0	99.9	−0.01	65.6	3.5 min
MLP (PPO)	21.5	80.7	+1.40	118.7	10 s

Two things are worth noting. First, supervised classifiers trained on the heuristic labels nearly match the labelling distribution itself — XGBoost at −0.0% and even logistic regression at −0.02%. This tells us the demand vector itself contains enough signal for the routing decision; the framework is what carries the contribution and not the choice of training algorithm. Second, the PPO row looks worst on accuracy (21.5%) because PPO is not trained to predict the heuristic labels. It optimises a multi-objective reward, and during training it explores configurations the heuristic never selects. The cost-gap number for PPO is on the same metric used for the supervised methods, but it is not a fair comparison because PPO is solving a different problem.

We use PPO at deployment for two practical reasons. Generating heuristic labels for a new operating point requires one simulator pass per training sample, which is too slow to retune when the operator weights (w_L, w_E, w_M, w_Q) change. PPO learns from interactive reward and adapts to weight changes without label regeneration. The supervised result is still useful evidence that the demand-vector design is doing the work; it is not a critique of the algorithm choice.

H PAVO-Bench annotation rubric

Annotators assigned complexity labels using this decision tree:

- Does the query require more than one reasoning step? No → Level 1 or 2. Yes → Level 3+.
- Is a single lookup sufficient? Yes → Level 1. No → Level 2.
- Does the query reference prior context or require synthesis? No → Level 3. Yes (emotional/open) → Level 4. Yes (structured output) → Level 5.

Inter-annotator agreement. The reported $\kappa = 0.81$ was computed on a 200-turn audit batch independently re-annotated by both authors using the decision tree above; Cohen's κ was calculated over the 5-class complexity labels.