PUSHING THE LIMITS OF SUB-10M MULTI-DOMAIN SPEECH RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Edge speech recognition requires multi-domain robustness under strict parameter caps, yet standard uniform compression degrades deep acoustic representations. In this paper, we propose AcuFormer, the first sub-10M ASR model designed for robust multi-domain performance within a complete 9.97M parameter envelope through an acoustically informed architecture. Guided by acoustic sensitivity, AcuFormer preserves full encoder depth while reallocating released feed-forward parameters into compact bidirectional GRU decoders that reuse acoustic token embeddings, targeted local recovery modules, and a blank-invariant prior for multi-view selection. Evaluated across all seven domains of the OpenASR benchmark, AcuFormer achieves 9.70% average WER, breaking the 10% error threshold for sub-10M models, yielding a 59.0% relative improvement over the compact state of the art, and outperforming baselines up to 7.8 times its size. Furthermore, real-world hardware profiling on a low-power Raspberry Pi Zero 2 W confirms real-time execution with an RTF of 0.53 and minimal memory residency. Taken together, these results push the limits of sub-10M multi-domain speech recognition by uniting deep acoustic accuracy with extreme on-device efficiency, facilitating private, always-available voice interfaces on ubiquitous edge devices.

1 INTRODUCTION

Natural speech interfaces offer a promising interaction layer for the physical world, enabling smart appliances, robotics, and wearable devices to understand natural human speech rather than rigid command keywords (Clark et al., 2019). To render this capability practical across billions of microphone-equipped edge devices, executing automatic speech recognition (ASR) locally on the device has become an essential requirement (Radford et al., 2023). Local inference guarantees complete data privacy by keeping audio strictly on-device, eliminates network dependency, and avoids round-trip latency (Jeffries et al., 2024). Furthermore, bringing reliable on-device ASR directly to extreme edge platforms provides a crucial foundation for truly ubiquitous and always-available voice interaction.

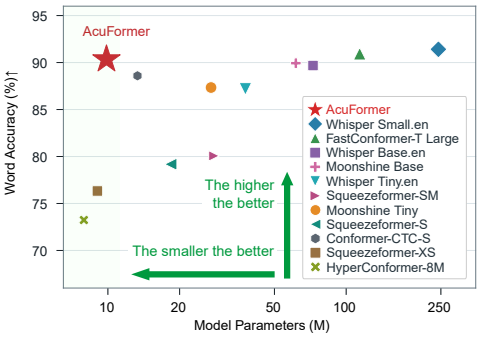

Figure 1: Accuracy versus model parameters on the seven-domain OpenASR benchmark. ACUFORMER advances the Pareto frontier.

However, achieving robust multi-domain accuracy under a strict parameter budget, like sub-10M, remains an unaddressed challenge. On resource-limited hardware, the recognizer shares storage with firmware, where memory access dominates inference cost (Lin et al., 2020). Existing autoregressive models, such as Whisper and Moonshine, consume most of this budget on token embeddings alone, while their attention caches introduce heavy runtime memory overhead (OpenAI, 2023; Useful Sensors, 2024). Conversely, uniformly shrinking non-autoregressive architectures severely degrades acoustic modeling capacity, as standard reduction ignores block-wise sensitivity variations.

In this paper, we introduce ACUFORMER, the first sub-10M ASR model designed for robust multi-domain speech recognition through an acoustically informed architecture. Rather than uniformly

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

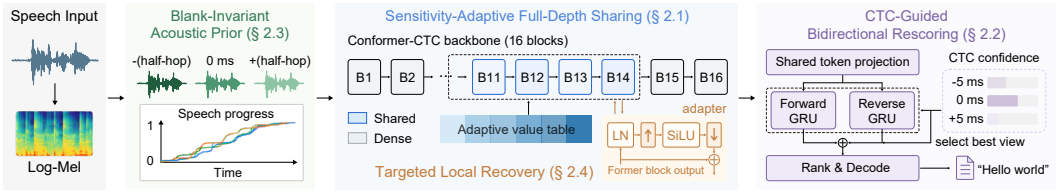

Figure 2: AcuFormer overview. Sensitivity-adaptive sharing preserves full backbone capacity, reallocating budget into bidirectional rescoring, blank-invariant priors, and targeted recovery.

shrinking model components, AcuFormer allows acoustic evidence to dynamically allocate the parameter budget end to end. Specifically, AcuFormer leverages Connectionist Temporal Classification (CTC) outputs to evaluate recognition sensitivity across the backbone (Graves et al., 2006). By identifying redundant feed-forward paths, it preserves all Conformer blocks at full width while releasing over 4M parameters. Furthermore, these released parameters are reinvested into two compact recurrent decoders that reuse the acoustic token matrix to capture bidirectional sequence context. Moreover, a blank-invariant prior aligns token evidence along speech progress, while local and temporal recovery modules restore the acoustic features affected by parameter sharing. Collectively, as shown in Figure 1, this design enables AcuFormer to significantly advance the Pareto frontier.

We evaluate AcuFormer across the seven-domain OpenASR benchmark against twelve public baselines. AcuFormer is the first sub-10M ASR model to break the 10% average word error rate (WER) threshold on this benchmark. This performance represents a 59.0% relative WER reduction over the sub-10M SOTA and surpasses all comparison systems up to 7.8 times its parameter size, including Whisper Base.en and Moonshine Base. Furthermore, AcuFormer achieves comparable accuracy to models up to 11.4 times larger. Beyond benchmark accuracy, real-world hardware profiling on a low-power Raspberry Pi Zero 2W validates its deployment viability at the extreme edge. Specifically, AcuFormer operates with minimal peak memory and delivers real-time local inference, processing 8s audio in 4.2s with a real-time factor (RTF) of 0.53, outperforming autoregressive baselines.

To summarize, our work makes the following key contributions:

- AcuFormer provides the first sub-10M ASR solution designed for robust multi-domain speech recognition, facilitating private and real-time voice interfaces for resource-constrained hardware.
- We introduce an acoustically informed ASR architecture that integrates four synergistic core mechanisms, including sensitivity-adaptive full-depth sharing to preserve backbone capacity, CTC-guided bidirectional rescoring with shared token projection, a blank-invariant acoustic prior for temporal shift stability, and targeted local recovery for feature restoration.
- We conduct comprehensive evaluations across seven domains against twelve public baselines alongside real-world edge deployment profiling. Results confirm that AcuFormer sets a new accuracy benchmark for sub-10M ASR while achieving lower latency and peak memory overhead.

2 AcuFormer Design

As shown in Figure 2, AcuFormer enables acoustic evidence to dynamically allocate the sub-10M parameter budget end to end. Sensitivity-adaptive scalar sharing first refits the 16-block Conformer-CTC backbone at full depth and width. Subsequently, the released parameters fund two gated recurrent unit (GRU) decoders and targeted local recovery modules. In addition, a blank-invariant prior stabilizes token evidence across temporal shifts while local recovery restores shared features. Nonblank CTC probabilities provide a unified signal that weights sensitivity metrics, constructs the speech-progress axis, conditions the forward decoder, and selects optimal inference views.

2.1 Sensitivity-Adaptive Full-Depth Sharing

We build upon the pretrained Conformer-CTC-S model trained on NeMo ASRSET (NVIDIA, 2023a). The baseline acoustic encoder contains $\sim$13M parameters, exceeding the 10M budget by 31% even without a language model. Because feed-forward networks (FFNs) constitute the dominant parameter mass, parameter sharing specifically targets these FFNs to free up capacity for sequence modeling.



Figure 3: Blank-invariant acoustic prior. Waveform shifts displace frame-level CTC event timing, whereas mapping onto the shared speech-progress axis preserves ordered token spans.

Specifically, we retain all 16 Conformer blocks with hidden width $d = 176$ and two width-704 Macaron FFNs per block (Gulati et al., 2020). Connectionist Temporal Classification (CTC) posteriors $p_t(v \mid x)$ over vocabulary tokens and a blank symbol $\varnothing$ are computed once per view and reused across candidates (Graves et al., 2006).

Sensitivity-guided assignment. We evaluate block sensitivity by replacing individual FFNs and measuring the CTC output change across 40 development utterances from five domains. Nonblank weighting focuses evaluation on active token frames, yielding the sensitivity metric

$$D_{\text{emit}}^{(\ell)} = \frac{\sum_{x,t}(1 - p_T(\varnothing \mid x, t))D_{\text{KL}}(p_T \| p_\ell)}{\sum_{x,t}(1 - p_T(\varnothing \mid x, t))}. \tag{1}$$

The ten blocks with the lowest sensitivity scores are $\{0, 15, 1, 2, 3, 4, 14, 8, 5, 6\}$, which are designated for scalar sharing. For each selected matrix $W \in \mathbb{R}^{o \times i}$, every weight uses a fixed uint8 index pointing into 255 shared values, while each row group learns an individual scaling factor, formulated as

$$\widehat{W}_{rj} = s_{r,g(j)} c_{I_{rj}}, \qquad I_{rj} \in \{0, \ldots, 254\}, \tag{2}$$

where $c \in \mathbb{R}^{255}$ is the shared value table and $s_{r,g}$ denotes the row-group scale. Quantization precision adapts to block sensitivity. The seven least sensitive blocks use row groups of 64 weights, blocks 8, 5, and 6 use groups of 8 for finer control, and the remaining six blocks remain fully dense. This mechanism applies to both FFN linear projections, replacing $\sim$4.96M independent weights with roughly 243k scalar values and scales, yielding a 95.1% parameter reduction.

Shared updates and local control. Keeping indices fixed, we fine-tune shared values and scales using domain-robust CTC training with GroupDRO (Sagawa et al., 2020) alongside knowledge distillation from source outputs and intermediate features (Kim & Rush, 2016). A projection regularizer prevents shared values from drifting from initial assignments (Han et al., 2016; Desai & Shrivastava, 2024). Consequently, the full-depth acoustic trunk comprises $\sim$8M parameters. Appendix B.1 derives shared gradient updates and scalar initialization. Releasing $\sim$4.71M parameters from the 13.15M baseline enables reallocating $\sim$0.39M to acoustic repair and $\sim$1.14M to sequence modeling, completing a compact 9.97M recognizer with a net saving of $\sim$3.18M parameters..

2.2 CTC-Guided Bidirectional Rescoring

CTC guides bidirectional rescoring by utilizing a shared acoustic token projection and an ordered summary of speech progress (Hori et al., 2017). Bidirectional rescoring incorporates two single-layer 512-dim GRUs scoring hypotheses in forward and reverse directions. Crucially, the forward language model contains 579,761 parameters and reuses the acoustic nonblank token projection $E \in \mathbb{R}^{1024 \times 176}$ via a learned 512-to-176 bridge. While embedding matrices in Whisper Tiny.en and Moonshine Tiny consume 19.92M and 9.44M parameters (OpenAI, 2023; Useful Sensors, 2024), this shared matrix requires only 180,224 parameters. Consequently, the system stores this projection matrix once, allowing both GRUs to operate without runtime attention caches.

In addition, the forward language model receives acoustic context features structured by speech progress, defined as the normalized cumulative nonblank CTC probability partitioned into three equal-mass temporal bins. For each bin $\mathcal{B}_k$, the context vector is computed as

$$a_k = \sum_v q_{k,v} E_v, \qquad q_{k,v} = \frac{\sum_{t \in \mathcal{B}_k} p_t(v)}{\sum_{t \in \mathcal{B}_k} \sum_{u \neq \varnothing} p_t(u)}. \tag{3}$$

A global feature together with the three bin context vectors initializes the forward GRU state through a rank-16 residual layer. The recurrent core is pretrained on a 44M-token English corpus, after which MWER training (Prabhavalkar et al., 2018) optimizes acoustic conditioning across target domains.

The reverse language model consists of 0.56M parameters trained on reversed sequences, using an MWER gate to weight right-side context. Within both GRU decoders, recurrent matrices share 256 scalar values with uint8 indices and group-4 scales, yielding $\sim$1M parameters in total. Appendix C.3 details storage formats and an optional sub-2MB low-bit variant. Token calibration adds 128 parameters as 64 shared biases and 64 sparse residuals. Appendix C.2 describes candidate search.

For any candidate y, the final scoring function is formulated as

$$\mathcal{J}(y \mid x) = s_{\mathrm{CTC}}(y \mid x) + 0.15 \log p_{\rightarrow}(y \mid x) + 0.15 \log p_{\leftarrow}(\mathrm{rev}(y)) + 0.6|y| + 0.1 b_{\mathrm{cal}}(y). \quad (4)$$

CTC prefix search uses a beam width of 32, token pruning threshold of 12, blank penalty of 0.8, and 12 candidates for reranking. A single fixed set of scoring weights serves all domains.

Complete model allocation. As listed in Table 1, the complete AcuFormer model comprises four components totaling 9,972,129 parameters, strictly counting shared parameters once under the sub-10M budget. Acoustic-only inference loads solely the encoder, whereas complete inference modes load all four components.

Table 1: Model parameter breakdown.

Component	Parameters (M)	Tensor Bytes (MB)
Acoustic model	8.830	36.971
Forward GRU	0.580	1.380
Reverse GRU	0.563	1.361
Calibration	< 0.001	0.009
AcuFormer	**9.972**	**39.722**

2.3 Blank-Invariant Acoustic Prior

Sub-frame temporal shifts can alter frame-level CTC predictions in compact models. To enforce shift robustness, we model temporal perturbations as equivalent acoustic views using waveform offsets $\mathcal{D} = \{-80, 0, +80\}$ samples (corresponding to -5, 0, and $+5$ ms) on 16-kHz audio x. Here $x^{(\delta)}$ denotes the zero-padded shifted waveform and $L^{(\delta)}$ represents its acoustic log-probabilities. As illustrated in Figure 3, shifting the waveform moves blank intervals while preserving the relative sequence order and probability mass of nonblank speech events.

During this training stage, all three acoustic views share the encoder and CTC head, yielding individual view losses $\ell_\delta(x, y)$. To align speech events across shifted frames, token evidence is evaluated at matching positions along the speech-progress axis. We define nonblank activity $a_{\delta,t} = 1 - p_\delta(\varnothing \mid t)$ and its normalized cumulative sum $c_{\delta,t} = \sum_{j<t} a_{\delta,j} / \sum_j a_{\delta,j}$. The overlap between frame t and the k-th of $K = 8$ equal-activity intervals is computed as

$$\omega_{\delta,t,k} = \left| [c_{\delta,t}, c_{\delta,t+1}] \cap [(k-1)/K, k/K] \right|. \quad (5)$$

For any nonblank token v, the interval token distribution follows

$$r_{\delta,k}(v) = \frac{\sum_t \omega_{\delta,t,k} p_\delta(v \mid t) / \max(a_{\delta,t}, \epsilon)}{\sum_t \omega_{\delta,t,k}}. \quad (6)$$

Letting JS_δ denote the generalized Jensen-Shannon divergence over the three views and $\bar{a}_\delta = T^{-1} \sum_t a_{\delta,t}$, the ordered-event regularization and overall training objective are formulated as

$$\mathcal{L}_{\mathrm{ord}} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{JS}_\delta(r_{\delta,k}) + 0.1\, \mathrm{JS}_\delta(\mathrm{Bern}(\bar{a}_\delta)), \quad (7a)$$

$$\mathcal{L}_{\mathrm{orbit}} = 0.5\, \mathbb{E}[\ell_0] + \mathbb{E}\left[\max_{\delta \in \mathcal{D}} \ell_\delta\right] + 0.05\, \mathcal{L}_{\mathrm{ord}} + 0.1\, \mathcal{L}_{\mathrm{anchor}}, \quad (7b)$$

where $\mathcal{L}_{\mathrm{anchor}}$ matches the center output of the pre-trained model at temperature two to maintain temporal accuracy, while the worst-case term optimizes the hardest shift. Combined with cumulative activity, this objective establishes a sub-frame acoustic prior that stabilizes token evidence across shifts without modifying the deployed architecture.

Blank-timing invariance. Given total nonblank mass $M_\delta = \sum_t a_{\delta,t} > 0$, frame t occupies a progress interval of length $a_{\delta,t}/M_\delta$. Pure blank frames occupy zero length, whereas blank-dominated frames receive proportionally small weights. Proposition 1 formalizes this invariance property.

Table 2: Seven-domain speech recognition on the OpenASR benchmark. Model parameters, eight-second peak activation memory, and macro WER on all eight evaluation splits.

Method	Model Parameters (M) ↓	Peak activation (MiB) ↓	WER (%) ↓								AVG WER (%) ↓
			AMI	E22	Giga	LS-c	LS-o	SPGI	TED	Vox	
Whisper Small.en	244.000	123.41	17.93	12.97	11.35	3.05	7.25	3.60	4.07	8.50	8.59
FastConformer-T Large	114.000	32.61	19.09	19.41	12.31	1.80	3.97	4.97	4.46	6.45	9.06
Squeezeformer-M (NeMo)	77.972	48.27	39.17	35.71	21.58	2.53	6.13	15.60	11.56	17.79	18.76
Whisper Base.en	72.600	48.59	21.13	15.09	12.83	4.25	10.35	4.26	4.87	9.76	10.32
Moonshine Base	61.500	11.78	17.79	17.65	12.19	3.23	8.18	5.46	5.22	10.81	10.07
Whisper Tiny.en	37.760	30.96	24.24	19.12	14.08	5.66	15.45	5.93	5.97	12.00	12.81
Squeezeformer-SM (NeMo)	28.185	32.61	41.16	37.85	22.87	2.76	6.91	17.51	12.61	18.80	20.06
Moonshine Tiny	27.093	10.18	22.77	21.25	14.41	4.52	11.71	7.70	5.64	13.27	12.66
Squeezeformer-S (NeMo)	18.566	25.27	42.06	39.90	23.54	2.96	7.30	18.68	13.21	19.93	20.95
Conformer-CTC-S	13.154	22.82	20.53	18.83	14.49	3.57	7.88	7.76	7.14	9.07	11.16
Squeezeformer-XS (NeMo)	9.032	18.90	45.80	44.38	26.41	3.78	9.24	22.43	15.26	22.06	23.67
HyperConformer-8M	7.925	20.31	50.13	48.21	29.32	4.82	11.59	25.96	18.90	25.15	26.76
AcuFormer	9.972	24.15	16.81	16.79	12.87	3.06	7.00	6.69	5.74	8.63	**9.70**

Proposition 1 (Blank-timing invariance) *Consider the normalized event representation in Equation 6 with positive total nonblank mass. Relocating, inserting, or removing zero-activity frames leaves every event representation unchanged. The same property holds when an event is split into adjacent frames sharing the same conditional token distribution and total activity mass. Views related by these operations yield zero token JS loss for any choice of bin count K.*

By aligning features along the speech-progress axis rather than frame indices, this mechanism evaluates evidence at equivalent sequence positions, preserving token identity while eliminating sensitivity to blank timing shifts. Appendix D.2 provides the mathematical proof and an averaging-stability bound. During inference, each view proposes a greedy CTC transcript $\widehat{y}^{(\delta)} = \mathcal{B}(\arg\max_v L^{(\delta)}_{t,v})$, where $\mathcal{B}$ denotes ordinary CTC collapse. We evaluate candidate proposals using their native view confidence

$$q_\delta = \frac{\log p_{\mathrm{CTC}}(\widehat{y}^{(\delta)} \mid L^{(\delta)})}{|\widehat{y}^{(\delta)}|}, \qquad \delta^* = \arg\max_{\delta \in \mathcal{D}} q_\delta, \tag{8}$$

with ties resolved toward the center view. The primary beam search and language models then decode solely the selected representations $L^{(\delta^*)}$. This selection relies strictly on internal CTC confidence without adding model parameters or extra decoder passes. Furthermore, a compute-routed variant evaluates side views only when center confidence falls below a validation threshold, allocating additional acoustic computation to challenging utterances as analyzed in Section 3.3.

2.4 Targeted Local Recovery

To prevent scalar-sharing quantization errors from propagating across Conformer blocks, targeted local recovery repairs features immediately following each shared block. For a selected block output $h_\ell \in \mathbb{R}^{T\times 176}$, the bottleneck adapter introduces the residual transformation

$$h^+_\ell = h_\ell + A_\ell(h_\ell), \qquad A_\ell(h) = U_\ell \,\mathrm{SiLU}\left(D_\ell \,\mathrm{LN}_\ell(h)\right), \tag{9}$$

where $D_\ell \in \mathbb{R}^{103\times 176}$ projects features to a low-dimensional bottleneck and $U_\ell \in \mathbb{R}^{176\times 103}$ restores original dimensions. One adapter follows each scalar-shared block, with up-projection weights initialized to zero. The ten adapters comprise 366,080 parameters in total and are exported as per-row INT4 values with FP16 scales requiring only $\sim$194 KB as detailed in Appendix B.3.

Local error correction. Bottleneck adapters provide compact residual corrections that counteract inter-layer error accumulation (Houlsby et al., 2019; Karimi Mahabadi et al., 2021). Tying adapter placement directly to shared blocks corrects quantization error e_ℓ immediately after block ℓ, preventing subsequent encoder layers from amplifying noise, as derived in Appendix B.3.

A small temporal residual branch complements these local corrections by refining final CTC logits using neighboring frame context. Using 22,895 parameters with rank $r = 19$ and depthwise kernel

System	Params (M)	WER (%)	Δ WER
Folded FFN-448 + GRU	9.994	14.80	–
+ full-depth sharing	8.683	10.64	−4.16
+ acoustic recovery	9.072	10.38	−0.26
+ bidirectional rescoring	9.972	**9.81**	−0.57

Table 3: Cumulative single-view design ablations.

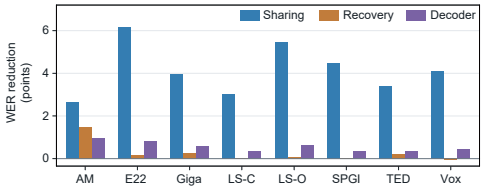

Figure 4: Per-domain WER drop comparison.

size three, this branch performs local temporal refinement via

$$z_t = W_{\text{down}} h_t, \qquad \ell'_t = \ell_t + W_{\text{up}}\,\text{SiLU}(\text{DWConv}_3(z)_t)\,. \tag{10}$$

Initializing $W_{\text{up}} = 0$, this branch is trained on balanced acoustic data using CTC loss, online N-best risk, and listwise ranking. Finally, a carefully designed five-step joint training stage optimizes both recovery modules alongside the fixed low-precision decoders under simulated INT4 adapter weights, producing a fully converged model that strictly conforms to the sub-10M parameter limit.

3 Experiments

3.1 Experimental Setup

Model landscape and benchmarks. We evaluate AcuFormer against twelve public recognizers across three parameter regimes. These encompass sub-10M acoustic architectures such as HyperConformer-8M and Squeezeformer-XS, established compact systems ranging from 13M to 78M parameters including Conformer-CTC-S, Squeezeformer variants, Whisper Tiny.en, Whisper Base.en, Moonshine Tiny, and Moonshine Base, alongside two large reference models above 100M parameters, namely Whisper Small.en and FastConformer-Transducer Large (Mai et al., 2023; Kim et al., 2022; Gulati et al., 2020; Radford et al., 2023; Jeffries et al., 2024; NVIDIA, 2023b). Parameter counts include all active inference components for each recognizer, accounting for both language models in AcuFormer. The OpenASR benchmark evaluates 82,116 total utterances across seven diverse domains (Srivastav et al., 2025), comprising AMI for meetings, Earnings-22 and SPGISpeech for financial speech, GigaSpeech for general audio, LibriSpeech for audiobooks evaluated on separate test-clean and test-other splits, TED-LIUM for presentations, and VoxPopuli for parliamentary speech. We report the equal-split macro WER across these eight evaluation splits as defined in Appendix F.1.

Evaluation and decoding. All AcuFormer evaluations and added baseline runs utilize the official OpenASR 2024 scoring protocol (Jeffries et al., 2024). For established compact systems, we report metrics directly from published literature, whereas official OpenASR leaderboard entries are retained for larger references. Local reruns verify these published macro WERs within a margin of 0.13 points. Baselines run using their native decoding setups, while added floating-point runs omit external language models to align with AcuFormer in audio ordering, references, and utterance counts. Complete AcuFormer inference incorporates both GRU decoders, calibration, and one or three acoustic views. Appendix F.2 details individual decoding configurations for all baseline systems.

Training and coverage. AcuFormer initializes from the pre-trained 13.15M-parameter NVIDIA Conformer-CTC-S checkpoint (NVIDIA, 2023a). Calibration, local recovery, temporal refinement, joint tuning, and ordered-event training process 14,640 sampled utterance instances, utilizing a fixed four-domain development set for model selection. Both GRU decoders are trained on a fixed 44M-token text corpus with all normalized 8-gram overlaps matching evaluation references removed. AcuFormer undergoes local acoustic adaptation on AMI, GigaSpeech, LibriSpeech, and TED-LIUM. Conversely, Earnings-22 and SPGISpeech receive no local acoustic updates, and VoxPopuli parameters are inherited directly from the source checkpoint, establishing a strict zero-shot protocol to evaluate cross-domain generalization.

Peak activation memory. Peak activation memory measures the maximum live activation memory during single-batch CPU inference on an 8s LibriSpeech dev-clean input in FP32. This metric encompasses input features, intermediate representations, output posteriors, and decoder state caches.

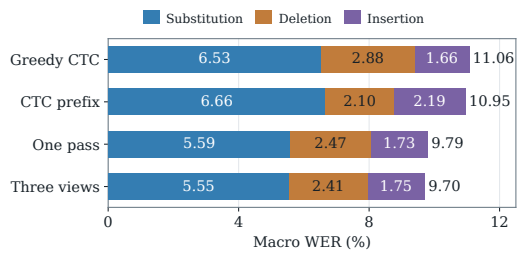

Figure 5: Word-error composition. Error types across decoding choices for the final checkpoint.

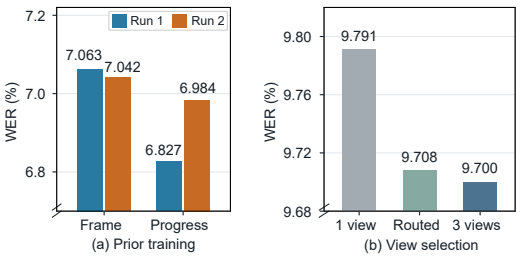

Figure 6: Prior and view comparisons. Matched training runs and inference view choices.

3.2 Main Results

As listed in Table 2, AcuFormer is evaluated against twelve public recognizers across all eight OpenASR evaluation splits. The complete AcuFormer configuration incorporates three-view selection, bidirectional GRU decoders, and token calibration. AcuFormer achieves a state-of-the-art 9.70 % AVG WER using only 9.972M model parameters. Compared to the 13.15M Conformer baseline, AcuFormer reduces absolute WER by 1.46 points, representing a 13.1% relative error reduction while using 24.2% fewer parameters. These performance gains cover all eight evaluation splits against the native Conformer as well as seven of eight splits against Moonshine Tiny and Whisper Tiny.en. Crucially, AcuFormer outperforms much larger baselines such as Moonshine Base and Whisper Base.en despite operating with 6.2 and 7.3 times fewer parameters. As shown on the parameter-efficiency frontier in Figure 1, AcuFormer comes within 0.64 WER points of FastConformer-Transducer Large (9.06% WER with 114M parameters) and closely trails Whisper Small.en (8.59% WER with 244M parameters), bridging the performance gap to models containing up to 24.5 times more parameters.

Comparison with sub-10M architectures. On the full seven-domain evaluation, compact baselines achieve 26.76% WER for the 7.93M HyperConformer-8M and 23.67% WER for the 9.03M Squeezeformer-XS (Mai et al., 2023; Kim et al., 2022). AcuFormer outperforms these architectures with relative WER reductions of 63.8% and 59.0%, demonstrating consistent gains across every evaluation split. Furthermore, larger NeMo Squeezeformer variants scale from 20.95% to 18.76% WER as parameter footprints expand from 18.6M to 78.0M. AcuFormer achieves superior performance at 9.70 % WER under a strict 10M parameter limit, surpassing even the 78M Squeezeformer-M on six of the eight evaluation splits.

Inference operating points. Decoupled from language models, the 8.8M-parameter encoder achieves 11.06% WER under greedy CTC decoding and 10.95% via prefix beam search, outperforming the 13.15M baseline Conformer with 32.9% fewer parameters. Integrating the bidirectional GRU decoder lowers WER to 9.79%, while three-view selection reaches 9.70% with zero extra parameters, allowing both complete configurations to surpass all sub-100M baselines. AcuFormer requires 24.15 MiB peak activation memory, undercutting Whisper Tiny.en (30.96 MiB) and Small.en (123.41 MiB) as listed in Table 2. Mixed-precision deployment retains the 9.70 % WER while compressing tensor storage by 14.6%. Finally, mean WER reductions over native Conformer reach 1.6 points on adapted domains and 1.5 points on unadapted financial domains, confirming robust zero-shot transfer across all evaluation utterances.

3.3 Ablation Studies

Cumulative design sequence. To isolate individual structural contributions, we evaluate scalar sharing, local recovery adapters, bidirectional rescoring, and shift regularization cumulatively across all 82,116 evaluation utterances under single-view inference. All ablation models maintain parameter counts below the 10M threshold using a standardized decoding search with beam width 32, token pruning 12, blank penalty 0.8, language model weight 0.30, and length weight 0.60. As quantified in Table 3, the progression starts from a constrained baseline featuring feed-forward networks narrowed to dimension 448, structural block folding through a bottleneck, and a tied 0.24M-parameter GRU decoder. This initial system achieves a macro WER of 14.80% at 9.9M parameters, establishing the baseline from which each architectural component restores speech recognition accuracy.

Sensitivity-adaptive full-depth sharing. Replacing structural block folding with full-depth scalar sharing preserves the complete acoustic layer stack while expanding feed-forward networks to their full dimension. This transition reduces overall parameter storage to 8.683M parameters while driving average WER down to 10.64%. The resulting 4.16-point absolute WER drop represents 83.4% of the total cumulative improvement across Table 3, confirming that preserving vertical acoustic depth is far more vital for speech representations than maintaining unshared shallow weights. As illustrated in Figure 4, full-depth sharing benefits all eight evaluation splits, with WER drops ranging from 2.64 points on AMI meeting speech to 6.19 points on Earnings-22 financial audio. Substitution errors fall from 9.35% to 6.19%, demonstrating that deep acoustic feature extraction provides fundamental phonetic distinction across diverse acoustic environments.

Targeted local recovery. Following depth preservation, targeted local recovery addresses parameter quantization noise and localized feature distortion introduced by scalar sharing. The recovery mechanism incorporates 366,080 bottleneck adapter parameters positioned immediately after shared acoustic blocks, alongside 22,895 temporal parameters operating on nonblank logit predictions. As listed in Table 3, local recovery lowers average WER from 10.64% to 10.38% while maintaining a compact 9.072M parameter footprint. Per-domain results in Figure 4 reveal that local recovery yields its strongest gain on the complex conversational interactions of AMI, reducing WER by 1.47 points. Meanwhile, GigaSpeech and TED-LIUM improve by 0.27 and 0.21 points, while unadapted financial splits retain steady accuracy, confirming that local recovery adapters refine complex acoustic boundaries without causing performance degradation on zero-shot domains.

CTC-guided bidirectional rescoring. Integrating acoustic-conditioned bidirectional GRU decoders adds 1.14M parameters and brings the total model size to 9.972M parameters. This module lowers cumulative average WER from 10.38% to 9.81%, completing a 33.7% relative WER reduction over the initial narrowed baseline under the 10M parameter limit in Table 3. As shown in Figure 4, directional language scoring brings consistent accuracy gains across all eight evaluation splits, with WER reductions spanning from 0.34 to 0.96 points. Analyzing error composition in Figure 5 highlights the specific mechanics of this improvement. Greedy CTC search yields 11.06% WER with 6.53% substitutions and 2.88% deletions, whereas prefix beam search achieves 10.95% WER with 6.66% substitutions and 2.10% deletions. Incorporating bidirectional GRU rescoring lowers macro WER to 9.79%, reducing substitution errors to 5.59% and deletion errors to 2.47%. This demonstrates that bidirectional sequence modeling resolves phonetic ambiguities that purely frame-independent CTC decoders cannot resolve.

Blank-invariant prior and multi-view selection. Applying blank-invariant shift regularization during training aligns acoustic evidence along the speech-progress axis rather than rigid frame indices. This training strategy lowers full-benchmark WER from 9.81% to 9.79% with zero added parameters or inference computation. Matched training runs in Figure 6(a) demonstrate that speech-progress regularization reduces mean development set WER from 7.05% to 6.91%, providing consistent gains across independent training runs. During inference, multi-view acoustic selection further lowers WER to 9.70% by evaluating temporally shifted acoustic views, as displayed in Figure 6(b). Confidence-based routing evaluates side views on 57.35% of utterances, achieving 9.71% WER while consuming only 2.147 mean acoustic passes and preserving over 90% of full three-view gains. As confirmed in Figure 5, multi-view selection primarily recovers missing speech tokens, reducing deletion errors from 2.47% to 2.41% and substitution errors from 5.59% to 5.55% across eight evaluation splits.

3.4 On-Device Profiling

We evaluate real-world edge deployment on a Raspberry Pi Zero 2 W featuring four Cortex-A53 CPU cores at 1 GHz and 512 MB RAM (Raspberry Pi Ltd, 2024). Benchmark measurements reflect warm FP32 inference across four CPU threads without memory swapping, evaluated across five audio durations using 16 kHz recordings. Complete end-to-end latency accounts for feature extraction, acoustic encoding, candidate search, and language model decoding. Board power consumption remains stable between 1.9 W and 2.2 W under full load (Raspberry Pi, n.d.; Geerling, 2021).

As profiled in Table 4, AcuFormer combines superior memory efficiency with competitive execution speed against representative edge ASR models. Operating with both language models and calibration enabled, complete AcuFormer inference reaches 9.79 % WER while maintaining the smallest model

Table 4: On-device profiling of lightweight ASR models on Pi Zero 2W.

Model	Size (MB) ↓	Latency (s) ↓					Peak RAM (MB) ↓					RTF ↓	Energy (mJ) ↓
		1 s	2 s	4 s	8 s	16 s	1 s	2 s	4 s	8 s	16 s		
Whisper Tiny.en (Radford et al., 2023)	151.09	9.8	9.9	10.8	11.4	14.3	276	276	277	276	278	1.43	22,413
Moonshine Tiny (Jeffries et al., 2024)	186.05	0.7	1.1	2.3	4.5	9.4	278	279	281	290	320	0.56	8,587
Conformer-CTC-S (Gulati et al., 2020)	49.02	0.4	0.7	1.3	2.3	5.4	121	124	132	146	175	0.29	4,869
ACUFORMER	34.39	0.6	1.0	2.1	4.2	8.9	133	134	135	137	165	0.53	8,704

Latency and energy entries are means over five runs after warmup; peak RAM is the maximum resident set size, covering model state, activations, and the native runtime. RTF and energy use 8-second input.

footprint at 34.39 MB and the lowest 8-second peak memory at 137 MB. Compared to autoregressive baselines, AcuFormer decodes an 8-second input in 4.2 seconds, outperforming Whisper Tiny.en (11.4 seconds) and Moonshine Tiny (4.5 seconds). Across increasing audio lengths from 1 to 16 seconds, AcuFormer exhibits minimal RAM expansion, growing from 133 MB to 165 MB compared to Moonshine Tiny which expands from 278 MB to 320 MB. On 16s audio inputs, AcuFormer completes inference in 8.9s with a real-time factor of 0.56 while remaining the only dual-language-model system operating below 170 MB memory residency on resource-constrained hardware.

4 Related Work

Compact acoustic architectures and parameter compression. Efficient speech recognition relies on streamlined attention-convolution encoders (Gulati et al., 2020; Burchi & Vielzeuf, 2021; Kim et al., 2022; Yao et al., 2024), pure 1D convolutional models (Kriman et al., 2020; Majumdar et al., 2021), and alternative state-space or hypernetwork generators (Mai et al., 2023; Pei, 2025). In parallel, edge systems like Moonshine and Whisper target local deployment and multi-domain robustness across benchmarks such as ESB and OpenASR (Jeffries et al., 2024; Radford et al., 2023; Gandhi et al., 2022; Srivastav et al., 2025). To further shrink model footprints, structured pruning, low-rank factorization, quantization, and knowledge distillation systematically remove parameter redundancy (Gong et al., 2014; Han et al., 2016; Jacob et al., 2018; Egiazarian et al., 2024; Kim et al., 2024; Shao et al., 2024; Frantar & Alistarh, 2023; Hinton et al., 2015; Kim & Rush, 2016; Gandhi et al., 2023), while lightweight adapters provide parameter-efficient residual updates (Houlsby et al., 2019; Karimi Mahabadi et al., 2021; Hu et al., 2022). Distinct from uniform compression or isolated component sizing, AcuFormer applies sensitivity-adaptive scalar sharing to preserve complete acoustic depth, reinvesting saved parameters into localized recovery adapters.

Hybrid sequence rescoring and alignment stability. Hybrid CTC-attention systems, recurrent neural network transducers, and autoregressive encoder-decoders integrate language context with acoustic frame predictions (Graves, 2012; Battenberg et al., 2017; Hori et al., 2017; Chan et al., 2016; Chiu et al., 2018). However, frame-synchronous CTC training often produces peaky posteriors and temporal instability, motivating relaxed independence assumptions and intermediate supervision (Zeyer et al., 2021; Nozaki & Komatsu, 2021; Lee & Watanabe, 2021). In parallel, sampling-aware anti-aliasing and test-time confidence adaptation address acoustic timing shifts (Chaman & Dokmanić, 2021; Rojas-Gomez et al., 2022; Bruguier et al., 2020; Liu et al., 2024). AcuFormer unifies these modeling requirements compactly by sharing the acoustic projection matrix across bidirectional GRU decoders and introducing a blank-invariant prior that aligns token evidence along speech progress, delivering robust multi-domain rescoring in a single inference pass.

5 Conclusion

In this paper, we present AcuFormer, the first sub-10M ASR model designed for robust multi-domain speech recognition through an acoustically informed architecture. Guided by acoustic sensitivity, AcuFormer preserves full encoder depth while reallocating released feed-forward parameters into compact bidirectional GRU decoders, targeted local recovery modules, and a blank-invariant prior. Evaluated across the seven-domain OpenASR benchmark, AcuFormer achieves 9.70% average WER, breaking the 10% error threshold for sub-10M models and outperforming all compared baselines up to 7.8 times its size. Moreover, hardware profiling on a low-power Raspberry Pi Zero 2 W demonstrates real-time inference with an RTF of 0.53 and an ultra-compact memory footprint, confirming the practical feasibility of high-accuracy speech recognition on extreme edge platforms.

AI USE STATEMENT

Generative AI tools were used to assist with mathematical formulation and proof development, the design of experiments and diagnostic analyses, and code implementation and debugging. The core technical design of this work, and all manuscript drafting and editing, were completed by the authors. The authors take responsibility for the final content of this work.

REFERENCES

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. Common Voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC)*, pp. 4218–4222, 2020. URL https://aclanthology.org/2020.lrec-1.520/.

Eric Battenberg, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur, Yi Li, Hairong Liu, Sanjeev Satheesh, Anuroop Sriram, and Zhenyao Zhu. Exploring neural transducers for end-to-end speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 206–213, 2017. doi: 10.1109/ASRU.2017.8268937.

Antoine Bruguier, Ananya Misra, Arun Narayanan, and Rohit Prabhavalkar. Anti-aliasing regularization in stacking layers. In *Proc. Interspeech 2020*, pp. 314–318, 2020. doi: 10.21437/Interspeech.2020-1497.

Maxime Burchi and Valentin Vielzeuf. Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 8–15, 2021. doi: 10.1109/ASRU51503.2021.9687874.

Jean Carletta. Unleashing the killer corpus: Experiences in creating the multi-everything AMI meeting corpus. *Language Resources and Evaluation*, 41:181–190, 2007. doi: 10.1007/s10579-007-9040-x.

Anadi Chaman and Ivan Dokmanić. Truly shift-invariant convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3773–3783, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Chaman_Truly_Shift-Invariant_Convolutional_Neural_Networks_CVPR_2021_paper.html.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, 2016. doi: 10.1109/ICASSP.2016.7472621.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio. In *Proc. Interspeech 2021*, pp. 3670–3674, 2021. doi: 10.21437/Interspeech.2021-1965.

Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. State-of-the-art speech recognition with sequence-to-sequence models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4774–4778, 2018. doi: 10.1109/ICASSP.2018.8462105.

Leigh Clark, Philip Doyle, Diego Garaialde, Emer Gilmartin, Stephan Schlögl, Jens Edlund, Matthew Aylett, João Cabral, Cosmin Munteanu, Justin Edwards, and Benjamin R. Cowan. The state of speech in hci: Trends, themes and challenges. *Interacting with Computers*, 31(4):349–371, 2019. doi: 10.1093/iwc/iwz016.

Aditya Desai and Anshumali Shrivastava. In defense of parameter sharing for model-compression. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ypAT2ixD4X.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract.html.

Linhao Dong and Bo Xu. CIF: Continuous integrate-and-fire for end-to-end speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6079–6083, 2020. doi: 10.1109/ICASSP40776.2020.9054250.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/egiazarian24a.html.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Sanchit Gandhi, Patrick von Platen, and Alexander M. Rush. ESB: A benchmark for multi-domain end-to-end speech recognition. *arXiv preprint arXiv:2210.13352*, 2022. doi: 10.48550/arXiv.2210.13352.

Sanchit Gandhi, Patrick von Platen, and Alexander M. Rush. Distil-whisper: Robust knowledge distillation via large-scale pseudo labelling. *arXiv preprint arXiv:2311.00430*, 2023. doi: 10.48550/arXiv.2311.00430.

Jeff Geerling. Disabling cores to reduce the Pi Zero 2 W's power consumption by half. Direct board power measurements, 2021. URL https://www.jeffgeerling.com/blog/2021/disabling-cores-reduce-pi-zero-2-ws-power-consumption-half/. Accessed September 23, 2026.

Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014. doi: 10.48550/arXiv.1412.6115.

Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012. doi: 10.48550/arXiv.1211.3711.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pp. 369–376, 2006. doi: 10.1145/1143844.1143891.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In *Proc. Interspeech 2020*, pp. 5036–5040, 2020. doi: 10.21437/Interspeech.2020-3015.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016. URL https://arxiv.org/abs/1510.00149.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer (SPECOM)*, pp. 198–208, 2018. doi: 10.1007/978-3-319-99579-3_21.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. doi: 10.48550/arXiv.1503.02531.

Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan. Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM. In *Proc. Interspeech 2017*, pp. 949–953, 2017. doi: 10.21437/Interspeech.2017-1296.

Neil Houlsby, Andrei Giurgiu, Stanisław Jastrzębski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2704–2713, 2018.

Nat Jeffries, Evan King, Manjunath Kudlur, Guy Nicholson, James Wang, and Pete Warden. Moonshine: Speech recognition for live transcription and voice commands. *arXiv preprint arXiv:2410.15608*, 2024. doi: 10.48550/arXiv.2410.15608.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/081be9fdff07f3bc808f935906ef70c0-Abstract.html.

Sehoon Kim, Amir Gholami, Albert Shaw, Nicholas Lee, Karttikeya Mangalam, Jitendra Malik, Michael W. Mahoney, and Kurt Keutzer. Squeezeformer: An efficient transformer for automatic speech recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 9361–9373, 2022. doi: 10.52202/068431-0680. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/3ccf6da39eeb8fefc8bbb1b0124adbd1-Abstract.html.

Sehoon Kim, Coleman Richard Charles Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. SqueezeLLM: Dense-and-sparse quantization. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/kim24f.html.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1317–1327, 2016. doi: 10.18653/v1/D16-1139.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021. URL https://proceedings.mlr.press/v139/koh21a.html.

Samuel Kriman, Stanislav Beliaev, Boris Ginsburg, Jocelyn Huang, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, and Yang Zhang. QuartzNet: Deep automatic speech recognition with 1D time-channel separable convolutions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6124–6128, 2020. doi: 10.1109/ICASSP40776.2020.9053889.

Jaesong Lee and Shinji Watanabe. Intermediate loss regularization for CTC-based speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6224–6228, 2021. doi: 10.1109/ICASSP39728.2021.9414594.

Ji Lin, Wei-Ming Chen, Yujun Lin, John Cohn, Chuang Gan, and Song Han. MCUNet: Tiny deep learning on IoT devices. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.neurips.cc/paper_files/paper/2020/hash/86c51678350f656dcc7f490a43946ee5-Abstract.html.

Hongfu Liu, Hengguan Huang, and Ye Wang. Advancing test-time adaptation in wild acoustic test settings. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7138–7155, 2024. doi: 10.18653/v1/2024.emnlp-main.405.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e5b1c0d4866f72393c522c8a00eed4eb-Abstract-Conference.html.

Florian Mai, Juan Zuluaga-Gomez, Titouan Parcollet, and Petr Motlicek. HyperConformer: Multi-head HyperMixer for efficient speech recognition. In *Proceedings of INTERSPEECH*, pp. 2213–2217, 2023. doi: 10.21437/Interspeech.2023-1611. URL https://www.isca-archive.org/interspeech_2023/mai23_interspeech.pdf.

Somshubra Majumdar, Jagadeesh Balam, Oleksii Hrinchuk, Vitaly Lavrukhin, Vahid Noroozi, and Boris Ginsburg. Citrinet: Closing the gap between non-autoregressive and autoregressive end-to-end models for automatic speech recognition. *arXiv preprint arXiv:2104.01721*, 2021. doi: 10.48550/arXiv.2104.01721.

Mohammad Mozaffari, Amir Yazdanbakhsh, and Maryam Mehri Dehnavi. SLiM: One-shot quantization and sparsity with low-rank approximation for LLM weight compression. In *International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/mozaffari25a.html.

Jumon Nozaki and Tatsuya Komatsu. Relaxing the conditional independence assumption of CTC-based ASR by conditioning on intermediate predictions. In *Proc. Interspeech 2021*, pp. 3735–3739, 2021. doi: 10.21437/Interspeech.2021-911.

NVIDIA. English Conformer-CTC small model card. https://huggingface.co/nvidia/stt_en_conformer_ctc_small, 2023a. Model card first published June 2023, accessed 2026-09-23.

NVIDIA. English FastConformer-Transducer large model card. https://huggingface.co/nvidia/stt_en_fastconformer_transducer_large, 2023b. Accessed 2026-08-23.

OpenAI. Whisper Tiny.en model configuration. https://huggingface.co/openai/whisper-tiny.en/blob/b70535dc924e7075c1d34736c0c825424301dc98/config.json, 2023. Released configuration, accessed 2026-09-20.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Proc. Interspeech 2019*, pp. 2613–2617, 2019. doi: 10.21437/Interspeech.2019-2680.

Yan Ru Pei. Let SSMs be ConvNets: State-space modeling with optimal tensor contractions. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/4c12e97f2e05304a451e18c9c945036f-Abstract-Conference.html.

Rohit Prabhavalkar, Tara N. Sainath, Yonghui Wu, Patrick Nguyen, Zhifeng Chen, Chung-Cheng Chiu, and Anjuli Kannan. Minimum word error rate training for attention-based sequence-to-sequence models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018. URL https://research.google/pubs/minimum-word-error-rate-training-for-attention-based-sequence-to-sequence-models/.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 28492–28518, 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Raspberry Pi. Typical power requirements. Raspberry Pi computer hardware documentation, n.d. URL https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#typical-power-requirements. Continuously updated documentation, accessed September 23, 2026.

Raspberry Pi Ltd. Raspberry Pi Zero 2 W product brief. Manufacturer product brief, April 2024. URL https://datasheets.raspberrypi.com/rpizero2/raspberry-pi-zero-2-w-product-brief.pdf. Accessed September 23, 2026.

Renan A. Rojas-Gomez, Teck-Yian Lim, Alexander G. Schwing, Minh N. Do, and Raymond A. Yeh. Learnable polyphase sampling for shift invariant and equivariant convolutional networks. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/e87b1e06be8c3594c810e8991e77ea40-Abstract-Conference.html.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Gao Peng, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c6483c8a68083af3383f91ee0dc6db95-Abstract-Conference.html.

Vaibhav Srivastav, Steven Zheng, Eric Bezzam, Eustache Le Bihan, Nithin Rao Koluguri, Piotr Żelasko, Somshubra Majumdar, Adel Moumen, and Sanchit Gandhi. Open ASR leaderboard: Towards reproducible and transparent multilingual and long-form speech recognition evaluation. *arXiv preprint arXiv:2510.06961*, 2025. doi: 10.48550/arXiv.2510.06961.

Useful Sensors. Moonshine Tiny model configuration and inference code. https://huggingface.co/UsefulSensors/moonshine-tiny/tree/efcb0fa94b87687337e972bd5978591b25acbc0d, 2024. Released configuration and native key-value cache implementation, accessed 2026-09-20.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey, and Tomoki Hayashi. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253, 2017. doi: 10.1109/JSTSP.2017.2763455.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. QA-LoRA: Quantization-aware low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e6c2e85db1f1039177c4495ccd399ac4-Abstract-Conference.html.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. Zipformer: A faster and better encoder for automatic speech recognition. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=9WD9KwssyT.

Albert Zeyer, Ralf Schlüter, and Hermann Ney. Why does CTC result in peaky behavior? *arXiv preprint arXiv:2105.14849*, 2021.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

APPENDIX CONTENTS

A Overview **16**

B Capacity-Preserving Encoder **16**

B.1 Scalar Representation and Initialization 16

B.2 Sensitivity and Group Allocation 17

B.3 Local Correction and Packed Recovery 18

C Shared Sequence Decoder **18**

C.1 Token Projection and Acoustic Conditioning 18

C.2 CTC Search and Bidirectional Scoring 19

C.3 Recurrent Sharing and Complete Capacity 19

D Blank-Invariant Acoustic Prior **20**

D.1 Event Pooling and Speech Progress 20

D.2 Blank-Timing Invariance and Order Sensitivity 21

D.3 Objective Roles and Numerical Evaluation 22

E Training Construction and Data **23**

E.1 Inherited Representation and Acoustic Updates 23

E.2 Optimization and Development Selection 24

E.3 Text Training and Overlap Filtering 24

F Seven-Domain Evaluation and Design Evidence **25**

F.1 Corpus Coverage and Aggregation 25

F.2 Native-System Domain Landscape 26

F.3 Accuracy per Unit Model Size 26

F.4 Cumulative Architecture and Error Types 28

F.5 Fixed-Checkpoint Decoding and Timing Effects 29

G Confidence Selection and Adaptive Computation **30**

G.1 Whole-View Confidence and Selection 30

G.2 Accuracy and Acoustic Compute Allocation 31

G.3 Domain-Level View Use 32

A OVERVIEW

The appendix develops the mechanisms behind ACUFORMER and expands the evidence for its compact multi-domain design. It follows the acoustic encoder, sequence decoder, and timing prior through training, evaluation, and Pi Zero 2 W deployment.

Section B derives the shared FFN representation and local recovery modules. Matrix-level counts and the sensitivity-based allocation show how the encoder retains its depth and width within a smaller budget.

Section C explains token-matrix reuse, acoustic conditioning, and recurrent sharing. Together, these mechanisms fit left and right language context into the capacity released by the encoder.

Section D develops the speech-progress coordinate and its blank-timing invariance. Constructed examples explain its treatment of timing and event order, and matched training runs test the recognition benefit.

Section E gives the inherited checkpoint, local training stages, sampling mixture, and filtered text corpus. The recipe connects each update stage to the component it trains.

Section F reports exact corpus counts, domain-level comparisons, and error decompositions. These results show where each design improves recognition across the seven datasets.

Section G analyzes confidence-based view selection. Routing results and view-use counts explain how the same model directs additional acoustic computation toward less confident utterances.

B CAPACITY-PRESERVING ENCODER

B.1 SCALAR REPRESENTATION AND INITIALIZATION

The source Conformer uses attention for longer-range relationships and convolution for local acoustic structure (Gulati et al., 2020). Its feed-forward paths transform these features between the sequence-mixing operations. AcuFormer changes the representation of selected FFN weights while retaining those paths, their widths, and their positions in the encoder, so full-size acoustic transformations are preserved while weight values are reused.

For a matrix with o rows and i columns, let g be the number of weights sharing a row scale. The final group of a row may contain fewer than g weights. Its representation requires

$$N_I = oi, \qquad N_s = o\,\lceil i/g \rceil\,, \qquad \widehat{W}_{rj} = s_{r,g(j)} c_{I_{rj}}. \tag{11}$$

The integer matrix I retains every weight position. Each position selects a shared value, and its row scale supplies local magnitude control. Reconstruction produces the full dense linear map.

Initialization starts from the pretrained matrix. Each row group receives the maximum absolute value of its weights as its scale. The shared table contains 255 uniformly spaced values on $[-1, 1]$. Each normalized source weight is assigned to the nearest entry. These assignments stay fixed during calibration, while scales and table values receive gradients through the reconstructed matrix. The source thus supplies both a useful acoustic representation and the initial pattern of value reuse.

Each shared value aggregates gradients from every position assigned to it. Writing $G_{rj} = \partial \mathcal{L} / \partial \widehat{W}_{rj}$, the two updates have the form

$$\frac{\partial \mathcal{L}}{\partial c_k} = \sum_{r,j:I_{rj}=k} s_{r,g(j)} G_{rj}, \qquad \frac{\partial \mathcal{L}}{\partial s_{r,g}} = \sum_{j:g(j)=g} c_{I_{rj}} G_{rj}. \tag{12}$$

The first update coordinates many matrix entries. The second lets each row group adapt to its own feature scale, and smaller groups give the more sensitive shared blocks more local freedom. Fixed indices keep this optimization continuous and retain the source model's weight assignments.

Each of the ten selected blocks contains two FFNs, each with two linear maps. Consequently, the number of replaced dense weights is

$$N_{\mathrm{FFN}} = 10 \times 4 \times 176 \times 704 = 4{,}956{,}160. \tag{13}$$

Sharing retains the FFN biases, all unselected modules, and the original feature channels. It reduces the number of values needed to specify the selected matrices.

Low-bit methods improve compact representations through range balancing, nonuniform values, calibration, and rotations (Xiao et al., 2023; Kim et al., 2024; Shao et al., 2024; Liu et al., 2025). Additive codebooks and parameter sharing also reuse values across larger weight structures (Egiazarian et al., 2024; Desai & Shrivastava, 2024). ACUFORMER uses recognition sensitivity to place sharing where it best preserves speech features, then allocates the saved parameters to recovery and decoding.

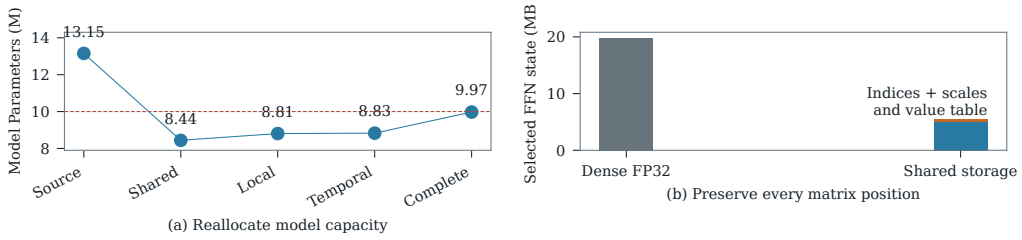

Figure 7: Capacity reallocation. Shared FFNs preserve acoustic width and release parameters for recovery and decoding.

Figure 7 traces the allocation from shared weights to acoustic recovery and sequence modeling, with storage including the fixed assignment indices.

B.2 SENSITIVITY AND GROUP ALLOCATION

The probe measures the output change caused by replacing one block's FFNs. Its 40 utterances contain eight examples from each of LibriSpeech dev-clean, LibriSpeech dev-other, TED-LIUM, AMI, and GigaSpeech. Weighting the output divergence by source nonblank probability directs the score toward frames carrying speech evidence.

The resulting order from least to most sensitive is

`0, 15, 1, 2, 3, 4, 14, 8, 5, 6, 7, 9, 12, 13, 10, 11.`

The first seven blocks use group size 64. The next three use group size 8, which gives them finer scale control. The last six keep their dense FFN weights. One acoustic criterion thus guides both sharing locations and scale precision. Weight pruning and combined sparse, quantized, and low-rank representations offer other forms of compression (Frantar & Alistarh, 2023; Mozaffari et al., 2025). Here the allocation preserves every FFN channel and block position.

Table 5 lists the block groups, matrix entries, and scale counts used by this allocation.

Table 5: FFN precision allocation. Sensitivity determines which blocks share values and their scale-group sizes.

Block indices	Group size	Matrix entries	Group scales
0, 15, 1, 2, 3, 4, 14	64	3,469,312	56,672
8, 5, 6	8	1,486,848	185,856
7, 9, 10, 11, 12, 13	Dense	2,973,696	–

For the actual rectangular matrices, ceiling terms matter. A width of 176 requires three groups at size 64, whereas a width of 704 requires eleven. Across both FFNs, one group-64 block uses 8,096 scales and one group-8 block uses 61,952 scales. The complete count is

$$\begin{aligned} N_s &= 7 \times 2\,[704\lceil 176/64\rceil + 176\lceil 704/64\rceil] \\ &\quad + 3 \times 2\,[704\lceil 176/8\rceil + 176\lceil 704/8\rceil] \\ &= 56{,}672 + 185{,}856 = 242{,}528. \end{aligned} \tag{14}$$

Adding the common 255-value table gives 242,783 model parameters in the selected matrices. The complete trunk consequently has

$$13{,}154{,}033 - 4{,}956{,}160 + 242{,}783 = 8{,}440{,}656 \tag{15}$$

model parameters. The reduction is 95.1% within the selected matrices, while attention, convolution, and all sixteen block positions remain intact.

B.3 Local Correction and Packed Recovery

The recovery branch is placed after each shared block. Its down projection maps 176 features to 103, and its up projection restores 176 features. A LayerNorm supplies scale and bias before the down projection. The up projection starts at zero, preserving the shared trunk's initial output. Training then learns a correction at each shared block.

Bottleneck adapters and low-rank updates provide compact adaptation paths (Houlsby et al., 2019; Karimi Mahabadi et al., 2021; Hu et al., 2022; Zhang et al., 2023). Quantized adaptation combines such updates with low-precision base weights (Dettmers et al., 2023; Xu et al., 2024). Our placement targets the features changed by sharing, correcting them before they reach later blocks.

The branch count follows from its two bias-free matrices and LayerNorm.

$$P_A = 10\,[2(176 \times 103) + 2 \times 176] = 366{,}080. \tag{16}$$

The count includes both projection matrices and LayerNorm values. INT4 export reduces their storage precision while retaining these parameters.

A local bound explains the benefit of correcting errors at their source. If the remainder of the encoder is locally L_ℓ-Lipschitz, a compression error e_ℓ and correction A_ℓ satisfy

$$\|F_{>\ell}(h_\ell + e_\ell + A_\ell) - F_{>\ell}(h_\ell)\| \le L_\ell \|e_\ell + A_\ell\|. \tag{17}$$

Reducing the residual $e_\ell + A_\ell$ tightens this bound before later blocks transform the features. The cumulative ablation measures the recognition gains from this placement.

The temporal branch repairs a different level of representation. Its rank-19 projection compresses the final encoder features, a three-tap depthwise convolution combines neighboring frames, and an output projection adds a correction to the nonblank logits. The parameter count is

$$P_T = 176 \times 19 + 19 + 3 \times 19 + 19 + 19 \times 1024 = 22{,}895. \tag{18}$$

The local branches act throughout the shared encoder. The temporal branch acts directly on nearby token evidence. Together, they produce the 8,829,631-parameter acoustic model.

At export, the adapter matrices use signed INT4 values with FP16 row scales. LayerNorm values use FP16. Their complete payload is

$$\begin{aligned} B_A = {} & 10(176 \times 103) + 2\,[10(103 + 176)] \\ & + 2\,[10(2 \times 176)] = 193{,}900 \text{ bytes}. \end{aligned} \tag{19}$$

The terms store packed matrix values, row scales, and normalization values, respectively. This compact payload supports targeted repair throughout the shared encoder.

C Shared Sequence Decoder

C.1 Token Projection and Acoustic Conditioning

The decoder uses the acoustic nonblank token matrix $E \in \mathbb{R}^{1024\times176}$ as a common representation. A token input looks up its row of E. A learned bridge maps the recurrent state back to 176 dimensions before output scoring with the same vocabulary matrix. The forward and reverse models have their own recurrent dynamics, but their large vocabulary-dependent matrix is stored only once in the acoustic model.

One token matrix contains $1024 \times 176 = 180{,}224$ values. Reusing it for both input and output in two language models avoids four additional copies, saving 720,896 values within this decoder architecture.

CTC scores complete transcripts through framewise token probabilities (Graves et al., 2006), and a recurrent model uses preceding words to distinguish candidates that have similar acoustic support. A reverse model adds the corresponding right context. Related CTC approaches strengthen context through intermediate predictions or joint attention decoding (Nozaki & Komatsu, 2021; Hori et al., 2017; Watanabe et al., 2017). ACUFORMER adds both sequence directions through compact recurrent scorers that share the acoustic token representation.

The forward initial state also depends on the utterance. A global summary averages token evidence over the acoustic output. Three ordered summaries divide cumulative nonblank mass into equal portions. For each portion, the weighted average of rows of E gives a vector in the same 176-dimensional space as the input token representation. A rank-16 residual map converts these vectors into an initial-state correction.

C.2 CTC SEARCH AND BIDIRECTIONAL SCORING

CTC assigns a transcript the sum of probabilities of all frame paths that collapse to it. Prefix search keeps candidate prefixes while maintaining separate blank-ending and nonblank-ending contributions. The deployed configuration uses beam width 32, at most 12 token proposals per frame, and blank penalty 0.8. Twelve complete candidates enter directional reranking. These settings stay fixed across the eight evaluation rows.

The fixed score in Equation 4 combines CTC, both sequence directions, length, and token calibration. The two sequence weights sum to 0.30, matching the small tied-GRU baseline's language-model weight. The cumulative ladder also retains its length coefficient of 0.60.

Candidate training relates sequence scores to recognition errors. Minimum word-error training uses a distribution over candidate hypotheses to make the training target closer to WER (Prabhavalkar et al., 2018). The forward acoustic-conditioning module ranks lower-error candidates above higher-error alternatives. The reverse candidate gate learns when the right-to-left score helps under domain-balanced training. At inference, the fixed score combines the candidate transcript with its acoustic evidence.

Token calibration is also compact. It contains 64 shared biases and 64 sparse residual corrections. Fixed lookup assignments identify the shared bias for each token, while a sparse index identifies tokens with a residual. The calibration contains 128 parameters, with lookup tensors included in its stored payload.

The fixed-checkpoint comparison isolates the decoder's practical effect. Greedy CTC reaches 11.065 exact macro WER, acoustic-only prefix search reaches 10.949, and complete one-pass decoding reaches 9.791. The improvement over prefix search therefore comes from the sequence-scoring path on the same final acoustic output. Section F.5 shows how that gain is distributed across error types and domains.

C.3 RECURRENT SHARING AND COMPLETE CAPACITY

Both sequence models use a hidden width of 512. Their recurrent hidden-to-hidden transform has three 512-by-512 matrices, one for each GRU gate or candidate-state operation. Each model represents this recurrent transform with a 256-value table, fixed uint8 indices, and group-4 scales. The geometry gives

$$N_{\mathrm{rec}} = 3 \times 512 \times 512 = 786{,}432, \tag{20}$$

$$N_{\mathrm{rec,scale}} = 3 \times 512 \times (512/4) = 196{,}608, \tag{21}$$

$$P_{\mathrm{rec,shared}} = 196{,}608 + 256 = 196{,}864. \tag{22}$$

Each direction has its own recurrent weights and shared-value table, preserving distinct forward and reverse sequence models.

At export, the group-scale matrix uses INT8 values with one FP16 scale per row. The shared table also uses FP16. One recurrent payload therefore occupies

$$B_{\mathrm{rec}} = 786{,}432 + 196{,}608 + 2(1{,}536 + 256) = 986{,}624 \text{ bytes}. \tag{23}$$

The terms account for fixed indices, quantized group scales, row scales, and shared values. Other large language-model matrices use the same row-wise INT8 format, while vectors use FP16. The

complete forward model contains 579,761 weight values and the reverse model contains 562,609. The difference also includes their distinct conditioning and scoring components.

The acoustic model and both sequence models together use

$$8{,}829{,}631 + 579{,}761 + 562{,}609 + 128 = 9{,}972{,}129 \tag{24}$$

model parameters across the acoustic model, both language models, and calibration. This leaves 27,871 parameters below the ten-million cap.

D Blank-Invariant Acoustic Prior

D.1 Event Pooling and Speech Progress

Moving a waveform by a few milliseconds changes its position relative to the feature grid, but leaves the spoken words and their order unchanged. The three training views use shifts of -80, 0, and $+80$ samples at 16 kHz. Zero padding keeps the input length fixed. All views use the same acoustic weights and the same transcript supervision.

Figure 8 follows one real utterance through the acoustic path. The waveform and short-time Fourier transform show the same recording in time and frequency. The model receives 80 normalized log-Mel channels. The CTC curves show nonblank activity under the three offsets. These views share the spoken sequence while changing its position within the fixed feature grid.

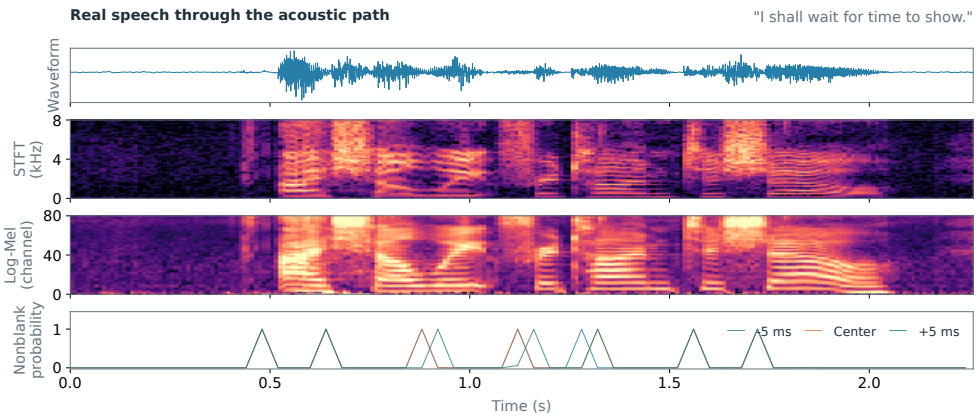

Figure 8: Acoustic views. Waveform, spectrum, log-Mel features, and CTC activity from one utterance under sub-frame shifts.

A CTC posterior contains both token evidence and blank intervals. The nonblank activity $a_t = 1 - p_t(\varnothing)$ measures how much token mass a frame emits. Conditional token probabilities separate its identity from that amount of activity.

$$f_t(v) = \frac{p_t(v)}{a_t}, \qquad \sum_{v \neq \varnothing} f_t(v) = 1 \quad \text{when } a_t > 0. \tag{25}$$

Normalizing the cumulative sum of a_t maps the utterance onto a unit speech-progress interval. A frame with larger activity occupies more of that interval. A blank frame occupies none.

For each of the eight equal progress bins, overlap weights average the conditional token probabilities. A frame crossing a bin boundary contributes to both neighboring bins in proportion to its overlap. This soft overlap avoids assigning a whole event to one bin merely because its boundary moved slightly. It also preserves a fixed number of summaries when the encoder emits different numbers of frames for different utterances.

The representation relates to monotonic accumulation methods such as continuous integrate-and-fire (Dong & Xu, 2020). Both use accumulated mass to organize acoustic evidence. Here the mass is the model's existing nonblank CTC probability, and the pooled distributions define a consistency target

across shifted waveforms using the existing acoustic output. This extends input-variation training (Park et al., 2019) with an explicit coordinate for matching speech evidence across frame boundaries.

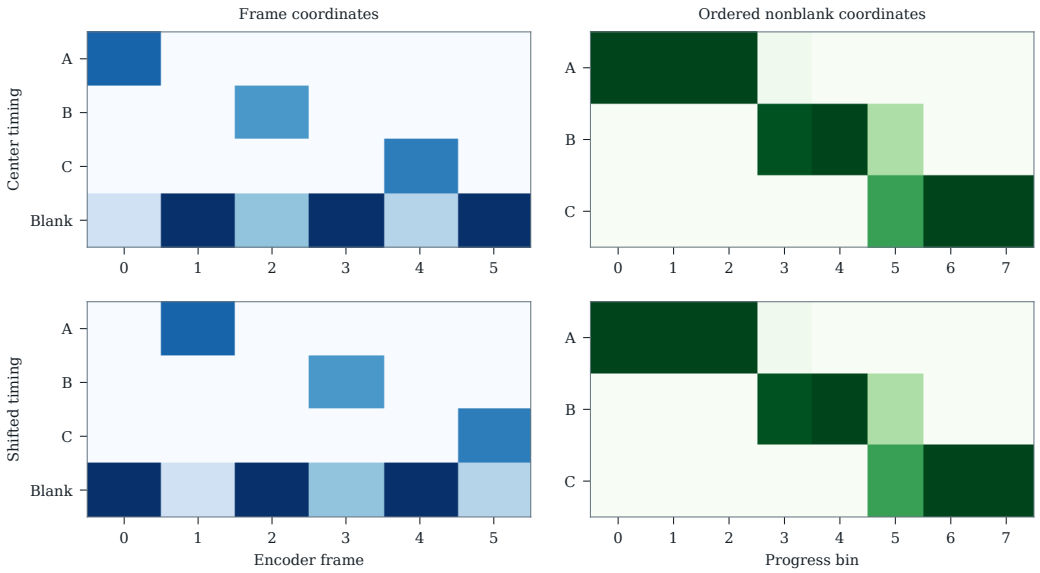

Figure 9: Constructed event alignment. Moving blank intervals preserves ordered speech-progress summaries.

Figure 9 uses constructed token and blank posteriors to make the coordinate change visible. The example has three ordered token events with activity masses 0.8, 0.6, and 0.7. Blank positions differ across the two frame sequences. Their event summaries match because their nonblank evidence has the same order and mass.

D.2 Blank-Timing Invariance and Order Sensitivity

Let $f(u, v)$ be the conditional token distribution at progress u. A posterior sequence defines a piecewise-constant function on $[0, 1]$, with one interval for each frame's normalized activity mass. The kth summary is its average over bin $I_k = [(k-1)/K, k/K]$.

$$r_k(v) = K \int_{I_k} f(u, v)\, du. \tag{26}$$

This form of Proposition 1 makes three properties explicit, namely blank-timing invariance, order sensitivity, and stability under changes in token evidence.

Blank relocation. Moving a zero-activity frame preserves every positive-length interval, leaving f and every r_k unchanged. The token consistency term is exactly zero for two such views. Splitting an event into adjacent frames also leaves the summaries unchanged when the split preserves the event's conditional distribution and total activity mass. Both changes alter the frame representation without altering the ordered token evidence.

Proof of Proposition 1. A zero-activity frame contributes no interval length. Changing its position or number therefore leaves the ordered positive-length intervals and their conditional distributions unchanged. Splitting an event only subdivides an interval on which f_δ is constant, so its total length and the integral over every bin remain unchanged. In both cases, f_δ is unchanged almost everywhere on $[0, 1]$, so all $r_{\delta,k}$ agree and their Jensen–Shannon divergence is zero. □

Order dependence. Swapping two distinct events generally changes the distributions over progress bins. When the changed event locations are resolved by the bins, the corresponding summaries differ

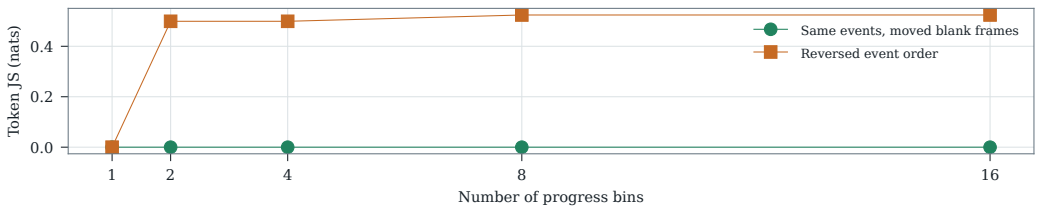

Figure 10: Constructed timing and order checks. Progress pooling preserves blank-timing invariance and detects reordered events.

and the JS loss becomes positive. A single bin averages the full sequence. Eight bins retain coarse event order by assigning successive portions of speech to separate summaries.

The constructed check in Figure 10 uses the same three-event sequence as the alignment diagram. Moving blanks produces zero token JS for all tested bin counts. Reversing the event order also produces zero with one bin, but produces 0.499 with two bins and 0.524 with eight bins. The example shows that progress pooling preserves timing invariance while responding to changes in event order.

Frame-index matching gives a different answer for the same timing example. Its posterior JS is 0.346 because a token in one sequence can be paired with a blank in the other. Progress pooling removes that mismatch by comparing the evidence at the same position in the event sequence. This isolates the mechanism that motivates the matched objective experiment.

Averaging stability. For two progress functions f and g, the triangle inequality yields

$$\|r_k^f - r_k^g\|_1 \le K \int_{I_k} \|f(u) - g(u)\|_1 \, du. \tag{27}$$

Averaging over bins gives

$$\frac{1}{K} \sum_{k=1}^{K} \|r_k^f - r_k^g\|_1 \le \int_0^1 \|f(u) - g(u)\|_1 \, du. \tag{28}$$

The mean change in pooled token evidence is bounded by its L_1 change on the progress axis. Together with blank-timing invariance, this gives the training target a stable representation of ordered speech evidence.

D.3 Objective Roles and Numerical Evaluation

The complete objective combines center CTC, hardest-view CTC, ordered-event consistency, and a center posterior anchor. The center term preserves accuracy at the original timing. The maximum CTC term supplies transcript supervision to the currently hardest shift. The event term compares the identity and order of token evidence. The anchor retains useful predictions from the model before this stage.

The generalized JS term compares each view distribution with their mixture. For three distributions r_δ and $m = \frac{1}{3}\sum_\delta r_\delta$, it is

$$\mathrm{JS}_\delta(r_\delta) = \frac{1}{3} \sum_\delta \sum_v r_\delta(v) \log \frac{r_\delta(v)}{m(v)}. \tag{29}$$

The value is zero when the distributions agree and is bounded above by $\log 3$. The implementation averages token JS over the eight progress bins. A separate Bernoulli JS term compares mean nonblank activity with weight 0.1. This second term tracks the amount of emitted evidence, while the first tracks its conditional token content.

The two terms control complementary properties. The token term is invariant to pure-blank insertion and relocation. Mean activity measures nonblank mass per frame, so its value also depends on sequence length. Fixed-length training views give the activity term a common duration while the token term follows speech progress.

Table 6: Ordered-event training. Matched runs compare consistency targets on the same development set.

Consistency target	Seed 38631	Seed 38613	Mean WER
Frame-aligned posterior JS	7.063	7.042	7.053
Ordered nonblank events	**6.827**	**6.984**	**6.906**
Paired change	−0.236	−0.058	−0.147

Table 6 compares two matched training runs. Only the consistency target changes between frame alignment and ordered events. Ordered events lower development WER in both runs, with paired reductions of 0.236 and 0.058 points. The mean reduction is 0.147 points. This experiment complements the constructed checks by measuring recognition under the actual training pipeline.

Numerical evaluation clamps denominators at $\epsilon = 10^{-7}$, masks padded frames, and averages consistency over utterances with positive nonblank mass. Zero-mass intervals contribute zero overlap. Implementation checks verify blank relocation, event splitting, order reversal, finite gradients, and zero-activity handling in the training loss.

E Training Construction and Data

E.1 Inherited Representation and Acoustic Updates

The starting model is `nvidia/stt_en_conformer_ctc_small`, with 13,154,033 parameters, sixteen blocks, hidden width 176, and FFN width 704. Its several-thousand-hour NeMo ASRSET pretraining supplies the acoustic representation (NVIDIA, 2023a). Local adaptation then reallocates its parameters, trains recovery modules, and learns the timing prior.

The source mixture combines read, conversational, meeting, and public-talk speech. LibriSpeech provides read audiobooks (Panayotov et al., 2015). AMI supplies meeting speech (Carletta, 2007), TED-LIUM supplies public talks (Hernandez et al., 2018), and GigaSpeech contributes diverse transcribed audio (Chen et al., 2021). Common Voice adds crowdsourced recordings (Ardila et al., 2020). The local MLS English slice further broadens the read-speech pool.

The sampler uses a virtual epoch of 1,280,000 entries. Figure 11 shows the source sampling probabilities. The five acoustic stages process 14,640 sampled utterance instances, including repeated draws. A fixed AMI pseudo-label manifest supplies local meeting-speech supervision.

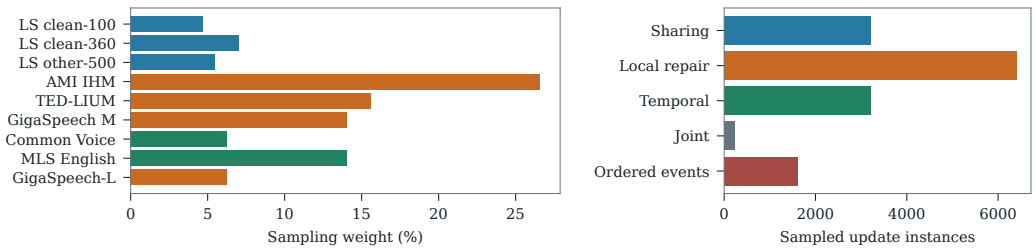

Figure 11: Local adaptation data. Source sampling probabilities and update counts for the selected checkpoint.

Meeting and public-talk sources receive substantial sampler weight during recovery. The final ordered-event stage uses LibriSpeech, Common Voice, and GigaSpeech-L. Its 200 selected steps at batch eight contribute 1,600 instances. The timing prior is therefore trained as a short final adaptation of the established acoustic representation.

Local adaptation covers the AMI, GigaSpeech, LibriSpeech, and TED-LIUM evaluation families. Earnings-22 and SPGISpeech receive no local acoustic adaptation. VoxPopuli is part of the released

source's reported supervision. This coverage separates locally adapted domains, newly evaluated domains, and a domain inherited from source pretraining.

Table 7 summarizes the released systems' acoustic supervision and its relation to the evaluation domains. Local update counts describe sampled instances, while source-model training scales describe inherited representation learning.

Table 7: Acoustic supervision. Training scale and evaluation-domain coverage for each recognizer.

System	Reported acoustic supervision	Test-domain relation
Released Conformer-CTC-S	Several thousand hours in NeMo ASRSET	LibriSpeech and VoxPopuli inherited
HyperConformer-8M	LibriSpeech 960-hour training split	LibriSpeech in-domain
Squeezeformer family (NeMo)	LibriSpeech 960-hour training split	LibriSpeech in-domain
Moonshine Tiny and Base	90k public plus over 100k internal hours	AMI, GigaSpeech, and LibriSpeech
FastConformer-Transducer Large	Several thousand hours in NeMo ASRSET 3.0	LibriSpeech and VoxPopuli inherited
Whisper Tiny.en, Base.en, and Small.en	438,218 English-ASR hours and 680k hours across all tasks	Published zero-shot evaluation
ACUFORMER	Released Conformer-CTC-S plus 14,640 sampled local instances	Local AMI, GigaSpeech, LibriSpeech, and TED-LIUM

E.2 Optimization and Development Selection

The stages follow the structure of the deployed model. Scalar calibration first adjusts table values and scales with fixed assignments. Recovery then trains the local branches against the shared trunk. Temporal training tunes the final CTC correction, followed by a short joint update of the existing recovery modules. Ordered-event training finally adapts the acoustic weights to the shifted views.

Table 8: Acoustic training recipe. Update settings for each stage of model construction.

Stage	Steps	Batch	Peak learning rate	Additional setting
Scalar calibration	100	32	5×10^{-6}	Penalty halved every 25 steps
Local recovery	200	32	5×10^{-4}	20-step warmup, EMA 0.99
Temporal recovery	25	128	5×10^{-5}	Fixed forward decoder
Joint recovery	5	48	2×10^{-5}	Simulated INT4 adapters
Ordered events	200	8	5×10^{-6}	25-step warmup, eight bins

Table 8 gives the training settings. Scalar calibration and local recovery use seeds 12913 and 17613. Ordered-event runs use seeds 38631 and 38613, train for 400 steps, and save every 100. A fixed four-domain development set selects step 200 for the primary run, seed 38631. The matched run uses the same step for the objective comparison and view-use analysis.

Domain-balanced updates use GroupDRO over five groups with minimum sampling probability 0.08 (Sagawa et al., 2020). Group-aware training addresses uneven performance across conditions (Koh et al., 2021). Source guidance uses output and feature distillation with the teacher gate defined in the main text (Hinton et al., 2015; Kim & Rush, 2016). The short joint stage then aligns the existing recovery modules with exported adapter rounding.

E.3 Text Training and Overlap Filtering

The sequence models learn token relationships from a fixed English text corpus. Filtering removes 2,103 lines that share a normalized 8-gram with any evaluation reference. The retained forward corpus contains 1,566,043 lines and 44,353,962 tokens. Reverse training reads reversed token sequences to learn right-to-left context.

Each evaluation reference contributes its normalized contiguous 8-grams to a blocked set. Text lines containing a blocked 8-gram are removed. Thirteen matching reverse-candidate rows are also replaced at their fixed positions. The retained text and candidates have no normalized 8-gram overlap with the evaluation references.

F Seven-Domain Evaluation and Design Evidence

F.1 Corpus Coverage and Aggregation

The benchmark contains seven datasets and eight evaluation rows because LibriSpeech test-clean and test-other remain separate. AMI tests meeting speech, Earnings-22 and SPGISpeech test financial speech, GigaSpeech adds diverse audio, TED-LIUM tests talks, and VoxPopuli tests parliamentary speech. This variety is central to the target use case of one compact recognizer across recording conditions and speaking styles (Gandhi et al., 2022; Srivastav et al., 2025).

For split d, let S_d, D_d, and I_d count substitutions, deletions, and insertions, and let N_d count reference words. Corpus WER divides the total substitutions, deletions, and insertions by the total reference-word count. The benchmark average is the equal-split macro WER

$$W_{\text{macro}} = \frac{1}{8}\sum_{d=1}^{8} \text{round}_2\left(100\frac{S_d + D_d + I_d}{N_d}\right), \tag{30}$$

which gives each domain split the same influence on AVG WER. Routing and error analyses use the same counts before rounding.

Scoring uses OpenASR runner revision `9bdea3164f93`, English normalizer revision `821e26a45128`, and dataset revision `b6bdcd0beb34`. Normalization runs once before word alignment. Local Conformer, Whisper Tiny.en, and Moonshine Tiny reruns reproduce published macro WER within 0.13 points, with Conformer matching all eight rows. Their normalized references and corpus counts match the AcuFormer evaluation.

Table 9: Complete corpus counts. Reference words and recognition errors for center-view and three-view inference.

			Word errors		WER (%)	
Row	Utterances	Ref. words	Center	Phase	Center	Phase
AMI	11,626	89,535	15,247	15,050	17.029095	16.809069
Earnings-22	2,737	48,899	8,272	8,212	16.916501	16.793799
GigaSpeech	19,856	403,606	52,494	51,955	13.006249	12.872703
LibriSpeech clean	2,620	53,027	1,652	1,623	3.115394	3.060705
LibriSpeech other	2,939	52,882	3,698	3,702	6.992928	7.000492
SPGISpeech	39,341	969,328	65,664	64,848	6.774178	6.689996
TED-LIUM	1,155	28,207	1,640	1,619	5.814160	5.739710
VoxPopuli	1,842	44,079	3,826	3,805	8.679870	8.632228
Total examples / exact macro	82,116	–	–	–	9.791047	9.699838

Table 9 covers all 82,116 utterances, and equal-row weighting gives each domain the same influence on AVG WER.

Rounding each row changes the macro average by at most 0.005 WER points,

$$\left|\frac{1}{8}\sum_{d} \text{round}_2(W_d) - \frac{1}{8}\sum_{d} W_d\right| \le 0.005. \tag{31}$$

The center-view score is 9.79125 under row rounding and 9.791047 before rounding. The three-view values are 9.69875 and 9.699838, respectively.

Recognition across training coverage. The eight benchmark rows in Table 2 form three groups based on local acoustic training. AMI, GigaSpeech, the two LibriSpeech splits, and TED-LIUM form the locally adapted group. Earnings-22 and SPGISpeech form a financial-speech group without local acoustic updates. VoxPopuli is shown separately because it is documented in source pretraining and

Table 10: Training-coverage groups. Mean WER within each group under equal split weights.

Group	Rows	Utterances	Native	One pass	Three views
				Mean WER (%)	
Locally adapted	5	38,196	10.722	9.192	9.096
Financial, no local updates	2	42,078	13.295	11.845	11.740
Inherited VoxPopuli	1	1,842	9.070	8.680	8.630

receives no local adaptation. The same checkpoint, decoder weights, and view-selection rule serve every row.

Table 10 averages the benchmark rows within each training-coverage group. Weighting these means by their row counts recovers the full OpenASR macro WER.

The one-pass mean reduction is 1.530 points in the locally adapted group and 1.450 points in the financial group. Three-view selection extends these reductions to 1.626 and 1.555 points. Their similar absolute gains show that the improvements extend beyond the locally adapted domains, and the VoxPopuli result adds an inherited source domain. One shared multi-domain deployment thus replaces separate acoustic models for each setting.

F.2 Native-System Domain Landscape

The established compact comparisons retain the Moonshine benchmark values (Jeffries et al., 2024), and the two larger references use official OpenASR records. Conformer and FastConformer use native greedy decoding, while Whisper and Moonshine use native generation. HyperConformer-8M retains its released CTC-attention search (beam 66, CTC weight 0.4, temperature 1.15), and Squeezeformer uses greedy CTC. HyperConformer-8M and Squeezeformer-XS add complete local evaluations of smaller public checkpoints. They reach 26.76 and 23.67 macro WER on the same 82,116 utterances. Squeezeformer S, SM, and M extend the comparison to larger releases trained on the same speech corpus. All twelve baselines retain the eight evaluation splits listed in Table 2.

Figure 12 shows gains across speaking styles and recording conditions. The three-view AcuFormer improves every Conformer row, including both financial-speech datasets and the inherited VoxPopuli domain. It also improves seven of the eight rows for each Tiny sequence-to-sequence baseline and every split for both HyperConformer-8M and Squeezeformer-XS, whose meeting and financial speech errors highlight the value of broad domain coverage.

Figure 13 expresses each reduction as a fraction of the baseline's errors. This shows gains relative to each domain's initial error rate, alongside the absolute WERs in the main table.

On Earnings-22 and SPGISpeech, two domains with no local acoustic adaptation, complete three-view recognition reduces WER from 18.83 to 16.79 and from 7.76 to 6.69, showing that the shared representation and decoder transfer beyond locally adapted domains.

The larger references define a practical accuracy scale. FastConformer Large reaches 9.06 macro WER at 114M parameters, and Whisper Small.en reaches 8.59 at 244M. AcuFormer reaches 9.70 with 1/11.4 and 1/24.5 of their parameters, respectively, placing it close to these larger accuracy references.

F.3 Accuracy per Unit Model Size

Figure 14 summarizes the word accuracy achieved per million model parameters for every model in Figure 1. For seven-domain macro WER $\overline{W}$ and complete parameter count P, the score is

$$\eta_{\text{param}} = \frac{100 - \overline{W}}{P/10^6}. \tag{32}$$

Both inputs come from the full evaluation in Table 2. The count includes every deployed component used by the recognizer.

AcuFormer reaches 9.06 accuracy points per million parameters, compared with 6.75 for Conformer-CTC-S and 8.45 for Squeezeformer-XS. These correspond to 1.34 and 1.07 times their

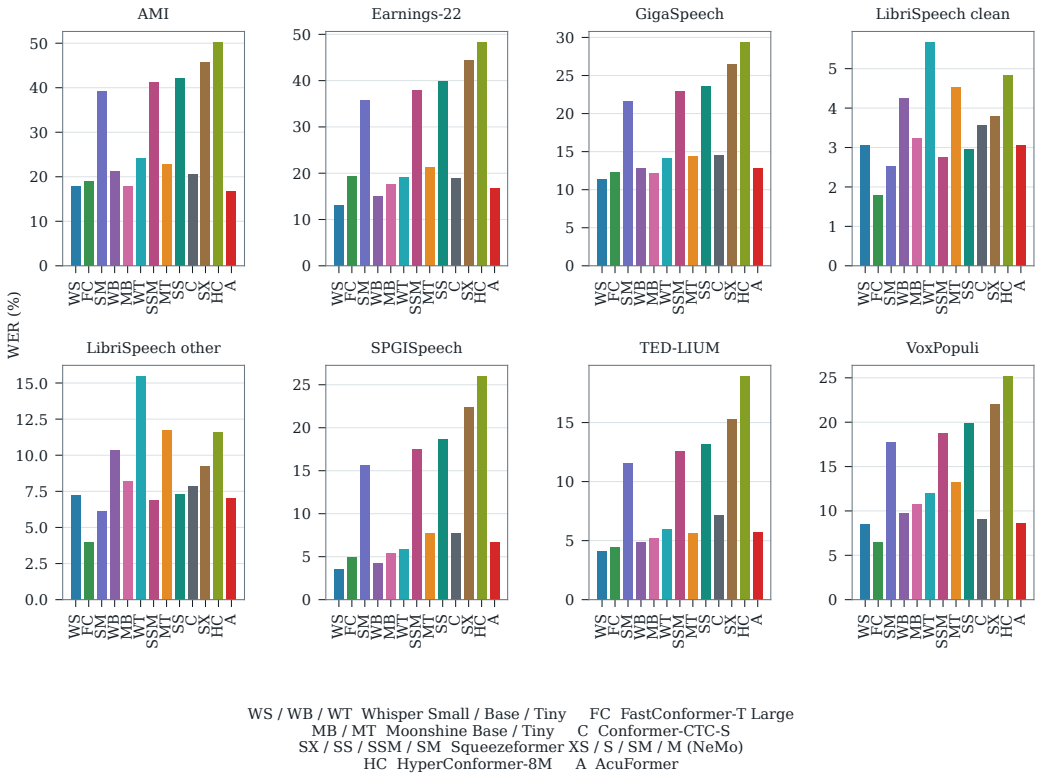

Figure 12: Domain-level recognition. AcuFormer, compact baselines, and larger accuracy references across the complete benchmark.

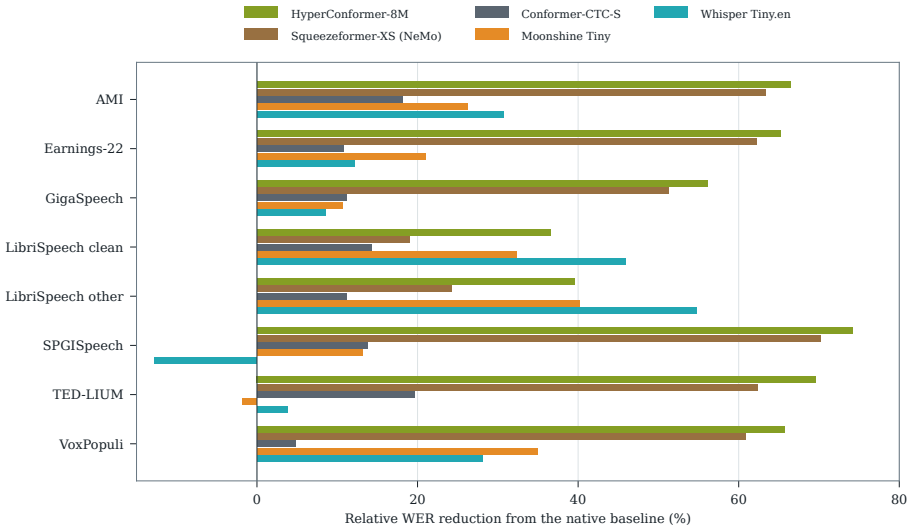

Figure 13: Relative domain gains. WER reductions over compact baselines across the evaluation splits.

parameter efficiency. HyperConformer-8M reaches 9.24 with 73.24% word accuracy, while ACUFORMER achieves 90.30%. Together with Figure 1, this separates accuracy per parameter from the accuracy achieved by the complete recognizer.

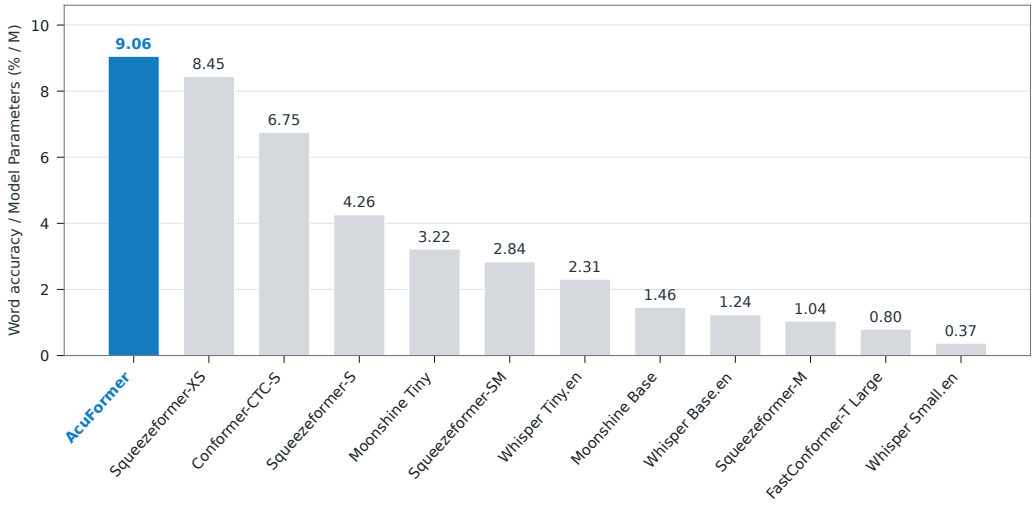

Figure 14: Accuracy per parameter. Word accuracy per million model parameters for the compared recognizers.

F.4 Cumulative Architecture and Error Types

Table 3 compares four cumulative architecture designs before prior training. Here, Table 11 expands those four systems into all eight evaluation splits. The same inputs and scorer expose where full-depth sharing, acoustic recovery, and bidirectional decoding improve recognition.

Table 11: Core ablations by domain. Recognition accuracy after each cumulative design addition.

Row	Folded	Shared	Recovered	Bidirectional
AMI	22.06	19.41	17.95	16.98
Earnings-22	24.07	17.88	17.70	16.88
GigaSpeech	17.89	13.92	13.66	13.07
LibriSpeech clean	6.48	3.47	3.50	3.13
LibriSpeech other	13.18	7.70	7.62	7.01
SPGISpeech	11.70	7.21	7.22	6.85
TED-LIUM	9.83	6.44	6.23	5.88
VoxPopuli	13.20	9.09	9.15	8.69
Exact macro	14.800	10.641	10.378	9.812

The full domain values in Table 11 show the breadth of this progression. Full-depth sharing improves all eight rows. Recovery has its largest effect on AMI, where it lowers WER by 1.47 points. Bidirectional scoring improves all eight rows again. This distribution matches the roles of the modules, with broad capacity preservation, targeted feature repair, and sequence-context gains across domains.

Figure 15 separates the three components of WER. The folded-to-shared change lowers the macro substitution rate from 9.35 to 6.19 and reduces both deletions and insertions. Recovery lowers substitutions by another 0.17 points and deletions by 0.09. The complete bidirectional stage then lowers all three macro error rates. The gains therefore involve token choice as well as transcript length.

Each error component uses the same corpus denominators as WER. Their exact macro rates sum to the exact macro WER at every stage, linking the domain-level explanation to the cumulative result.

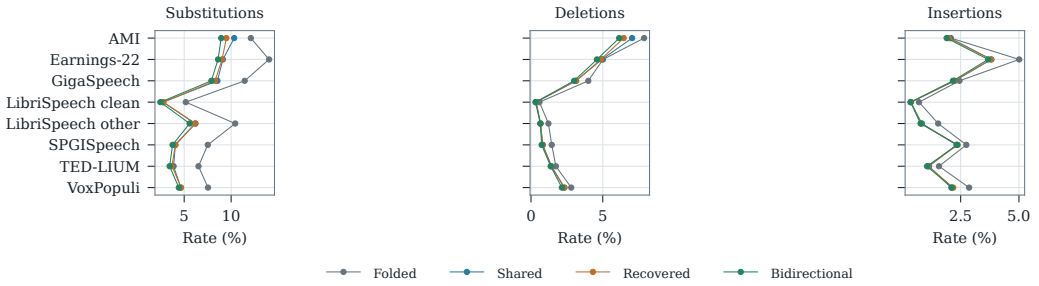

Figure 15: Error types across design stages. Domain-level substitutions, deletions, and insertions after each design addition.

F.5 Fixed-Checkpoint Decoding and Timing Effects

The second analysis holds the final acoustic checkpoint fixed. It compares greedy CTC, acoustic prefix search, complete center-view decoding, confidence routing, and complete three-view selection. This separates changes in inference from the structural and training changes in the cumulative ladder.

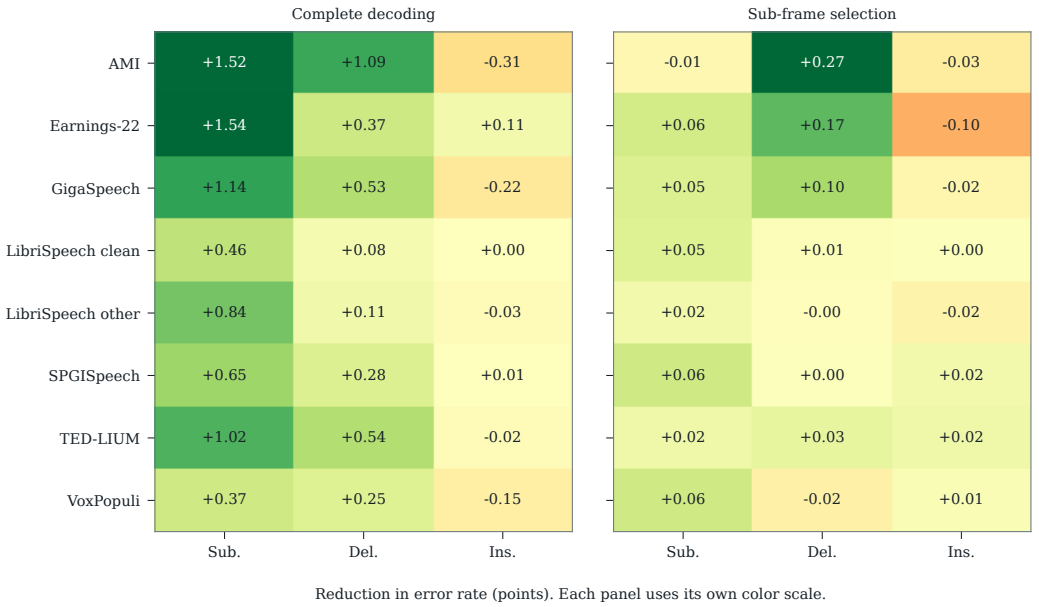

Figure 16: Fixed-checkpoint gains. Error changes from complete decoding and acoustic-view selection across domains.

The left panel of Figure 16 compares complete decoding with greedy CTC. The macro substitution reduction is 0.941 points, and the deletion reduction is 0.408 points. The insertion rate increases by 0.076 points, giving a net improvement of 1.274 points. The strongest contribution is therefore better token selection, supported by fewer missing words.

The right panel holds complete decoding fixed and changes only the acoustic view selected for it. Deletions fall by 0.068 points and substitutions by 0.038 points. Insertions change by 0.015 points, giving a net 0.091-point improvement. View selection thus refines token evidence after structural recovery and sequence scoring.

Table 12 separates the effects of search and sequence scoring. Complete decoding improves prefix-search WER by 1.158 points, reducing substitutions and insertions. The decoder therefore improves the choice among acoustically plausible word sequences.

Table 12: Fixed-checkpoint error composition. Macro error rates across search, decoding, and view-selection choices.

System	Substitutions	Deletions	Insertions	Total WER
Greedy CTC	6.527	2.882	1.656	11.065
Acoustic prefix search	6.665	2.096	2.188	10.949
Complete center view	5.585	2.474	1.732	9.791
Confidence-routed views	5.556	2.402	1.750	9.708
All three views	5.548	2.405	1.747	9.700

Reference	so speech recognition is most important as far as you are concerned
Greedy CTC	so speech recognition is most important as far as your concern
Complete	so speech recognition is most important as far as you are concerned

(a) AMI Complete sequence decoding

Reference	now the next slide shows the photovoltaic farm
Greedy CTC	now the next slide shows the photo vooltaic farm
Complete	now the next slide shows the photovoltaic farm

(b) Earnings-22 Complete sequence decoding

Reference	i lived my entire life waiting for this moment
Center view	i live my entire life waiting for this morning
Selected view	i lived my entire life waiting for this moment

(c) GigaSpeech Sub-frame view selection

Figure 17: Transcript corrections. Complete decoding and view selection improve word choices, with changed words highlighted.

At the domain level, three-view selection has its largest reductions on AMI, GigaSpeech, and Earnings-22, at 0.220, 0.134, and 0.123 points. Seven of eight rows improve, and the complete average improves with the same model state and decoder as center inference.

Transcript corrections. Figure 17 illustrates how the fixed-checkpoint gains change individual transcripts. The examples use the same English normalization as the benchmark. In the AMI example, complete decoding recovers the phrase *you are concerned* from the greedy output *your concern*. This restores the pronoun and verb form in a short conversational sentence. In Earnings-22, it recovers the specialized word *photovoltaic* from two incorrect fragments. The latter example comes from a domain without local acoustic adaptation.

The GigaSpeech example compares two outputs of the same complete recognizer. Selecting a shifted acoustic view changes *live* to *lived* and *morning* to *moment*, matching the reference with the same decoder. Confidence selection recovers better acoustic evidence before sequence scoring. Together, the examples show complementary effects, with complete decoding resolving word choices and view selection improving the evidence supplied to that decoder.

G Confidence Selection and Adaptive Computation

G.1 Whole-View Confidence and Selection

Each view first proposes its own greedy CTC transcript. The confidence is the CTC log probability of that transcript under the same view, divided by its token length. CTC probability sums compatible

alignments, so the score uses more information than the probability of one framewise best path. Length normalization makes scores for different proposal lengths more comparable. An empty proposal receives score $-\infty$, and a tie between empty views keeps the center.

Selection passes one complete posterior sequence to beam search and bidirectional decoding. Keeping the view intact preserves its CTC timing structure throughout candidate generation and scoring.

The routed path evaluates the center first. A score above $\tau = -0.0611302$ keeps the center. Otherwise, the two shifted views are evaluated and the largest self-CTC confidence wins. Ties favor the center. The threshold is the median center score on a held-out four-domain development set and is then fixed for every evaluation domain.

Extra views reuse the acoustic weights, and candidate decoding follows selection. All three operating points retain the same model state and invoke the complete decoder once per utterance.

G.2 Accuracy and Acoustic Compute Allocation

If a fraction r of utterances receives side views, the mean acoustic-pass count is $1 + 2r$. With $r = 0.57352$, the measured routing decisions give 2.14704 mean passes. This is an utterance-weighted operation count. Relative to evaluating all three views, it reduces acoustic passes by

$$1 - \frac{1 + 2r}{3} = 0.28432. \tag{33}$$

The decoder count remains one for every utterance.

Table 13: Adaptive acoustic computation. Accuracy and acoustic-pass counts for the same model under three inference policies.

Inference	Routed (%)	Mean AM fwd.	WER	Δ
Center view	0.00	1.000	9.791	–
Confidence-routed views	57.35	2.147	9.708	−0.083
All three views	100.00	3.000	**9.700**	−0.091

Table 13 connects this compute allocation to exact macro WER. The routed model reaches 9.707859, between the center value 9.791047 and the full-selection value 9.699838. Its retained fraction of the full WER improvement is

$$\frac{9.791046738 - 9.707858835}{9.791046738 - 9.699837720} = 0.9121. \tag{34}$$

Thus, the fixed threshold retains 91.2% of the observed selection gain while using 28.4% fewer acoustic passes.

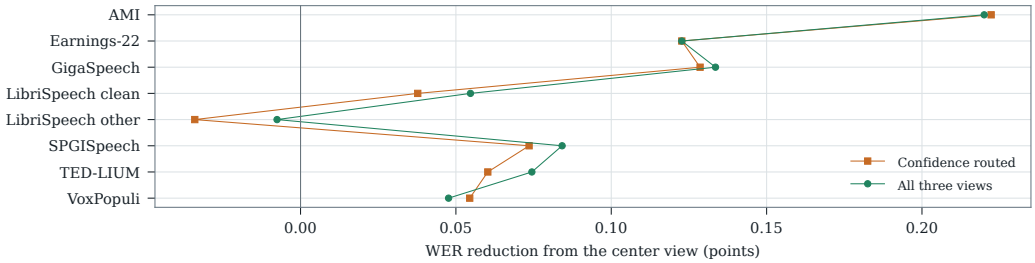

Figure 18: Domain-level routing gains. Confidence routing and full view selection relative to center-view inference.

Figure 18 shows where the gains occur. The domain curves remain close with fewer acoustic passes. A single threshold allocates these passes using utterance-level confidence across all domains. Section 3.4 profiles the one-view policy's execution and memory use on Pi Zero 2 W.

G.3 Domain-Level View Use

The second matched ordered-event run provides complete view counters for all 82,116 utterances. It selects the negative shift for 27,697 utterances, the center for 25,413, and the positive shift for 29,006. These correspond to 33.73%, 30.95%, and 35.32% of the evaluation.

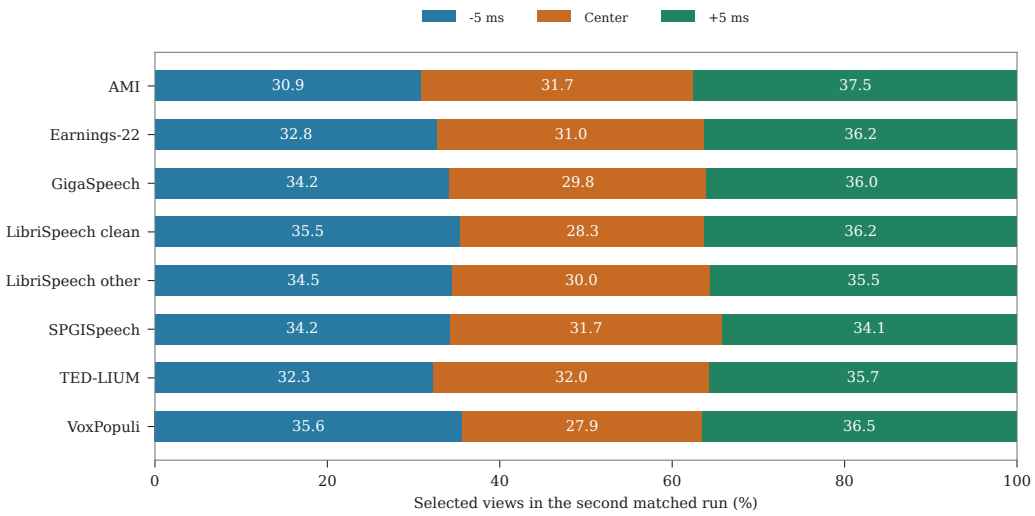

Figure 19: View use across domains. Selected shifts and center views for every utterance in the second matched run.

Figure 19 shows that both shifts and the center contribute across domains. These counts support utterance-level selection among nearby frame placements.

Each domain's counters cover its full utterance set, totaling 82,116 examples. The view-use analysis describes the second matched run, while primary WER and routing results use the primary checkpoint.