REDDIT: Correcting Model-Generated Timestamp Drift in ASR without Forgetting via Replay-Based Distribution Editing

Cheng-Kang Chou*[1], Ming-To CHUANG*[1], Ke-Han Lu[1], Chan-Jan Hsu[2], Hung-yi Lee[3]

[1]National Taiwan University [2]Carnegie Mellon University [3]NTU Artificial Intelligence Center of Research Excellence (NTU AI-CoRE)

***Abstract*—Modern autoregressive ASR systems can emit timestamps as decoded tokens, enabling timestamped transcription without frame-level aligners or inference-time post-processing. We show that these generated timestamps can drift across long non-speech spans: the transcript may remain plausible, but the decoded time axis drifts away from the audio. We study this non-speech-induced timestamp drift with self-built gap and long-gap benchmarks across 15 evaluated timestamp-producing ASR and audio-language systems. Naive timestamp-corrected fine-tuning improves alignment but can severely degrade non-target ASR behavior, exposing a forgetting problem. We propose *REDDIT* (REplay-based Distribution eDITing), a lightweight two-stage post-training framework that corrects timestamps while avoiding this catastrophic forgetting: it first edits timestamp targets under the model's own replayed decoder context while matching the frozen base distribution on non-timestamp tokens, then applies a short edited-prefix refinement stage. In this framework, we construct correction supervision without human transcripts or human timestamp annotations by combining VAD-trimmed speech spans with inserted non-speech gaps and known concatenation offsets. On Whisper-tiny, 34.9 hours of targeted correction audio used and only 1.6% of model parameters updated, raising long-gap mIoU from 38.7% to 95.0% and reducing mixed-gap out-of-domain AAS from 2752 ms to 223 ms while preserving CV-en MER at 41.3% (versus 524.2% for ordinary SFT decoder tuning).**

***Index Terms*—automatic speech recognition, model-generated timestamps, non-speech gap, replay-based learning, post-training, catastrophic forgetting**

I. Introduction

Automatic speech recognition (ASR) and audio-language models increasingly need to answer not only *what* was said, but also *when* it was said. Autoregressive speech systems expose this interface by generating time as part of the decoded output, either as timestamp tokens in ASR models such as Whisper or as word-level and text-form temporal predictions in speech-aware language models [1], [3]–[5]. This self-generated timestamp interface avoids the need for a separate VAD, frame-level aligner, or forced-alignment module at inference time.

This convenience introduces a failure mode that transcript-centric metrics can miss. Speech evaluation already extends beyond lexical content, e.g., tonality [46]; here, the missing dimension is temporal placement. Because the time axis is generated by the same decoder that emits text, a model can produce plausible content while assigning it to the wrong region of the audio. We focus on *non-speech-induced timestamp drift*: when speech follows a long silent or non-speech span, self-generated timestamps may place the following speech too early, too late, or under a globally displaced time axis, rendering subtitle timing unusable despite having an acceptable word error rate or task-level transcript quality.

A likely reason this failure is under-tested is the mismatch between common training and long-gap deployment conditions. Timestamped ASR examples often begin near speech onset, and VAD-based trimming can further increase speech density [11]. During training, this practice may reduce the model's exposure to long non-speech prefixes and strengthen a prior for early timestamp tokens near 0s.

Post-hoc alignment pipelines can refine boundaries after decoding, but they operate at a different level than correcting the model's native predictions. VAD segmentation, forced alignment, and attention-based alignment add inference-time modules and their own boundary assumptions [11]. Timestamp drift is also distinct from lexical hallucination: hallucination emits unsupported words, whereas timestamp drift can place supported words at the wrong time.

Naive timestamp-corrected fine-tuning reveals another problem. It can improve alignment on targeted timestamp-correction data while degrading recognition on non-target ASR data. The task is therefore not merely timestamp supervision, but forgetting-resistant temporal editing: move timestamp predictions to correct positions while preserving the decoder's original non-timestamp distribution.

We first benchmark non-speech-induced drift across 15 evaluated timestamp-producing ASR and audio-language systems using controlled gap and long-gap evaluations. The benchmark is diagnostic: it reports recognition quality, segment-level temporal overlap, fine-grained alignment, boundary error, large temporal displacement, and hallucination rate. The intervention study is then instantiated on Whisper-style timestamp-token ASR, where timestamp positions are explicit and token-level distribution editing is well-defined.

We propose *REDDIT*, REplay-based Distribution eDITing. *REDDIT* is a two-stage post-training pipeline for models that transcribe adequately but suffer from timestamp drift caused by non-speech gaps. Stage 1 edits timestamp targets under cached replay context while anchoring non-timestamp behavior to the frozen base distribution. Stage 2 then applies a short edited-prefix refinement from the Stage-1 checkpoint. In our Whisper-tiny experiments, *REDDIT* uses 34.9 hours of automatically constructed correction audio whose cached teacher replays are pre-filtered, updates only 0.59M parameters (last cross-attention with layernorms), and requires no inference-time aligner or post-processing.

Our contributions are:

- We identify non-speech-induced timestamp drift as a distinct failure mode of self-generated timestamps in ASR and audio-language systems, where plausible transcripts can be temporally misplaced across non-speech spans.
- We establish a diagnostic evaluation framework to benchmark timestamp-producing ASR systems on speech audio interleaved with challenging non-speech gaps, isolating joint alignment and recognition failures.
- We show that naive timestamp-corrected fine-tuning can cause severe forgetting on non-target ASR data, motivating timestamp adaptation as a model-editing problem.
- We propose *REDDIT*, a two-stage pipeline that constructs correction examples without human transcripts or human timestamp annotations, edits timestamp behavior under cached replay context, and refines the corrected behavior while preserving non-target recognition.

II. Related Work

A. Generated Timestamp Prediction

Autoregressive ASR systems increasingly expose time as part of the decoded sequence. Whisper emits timestamp tokens together with

* Equal contribution.

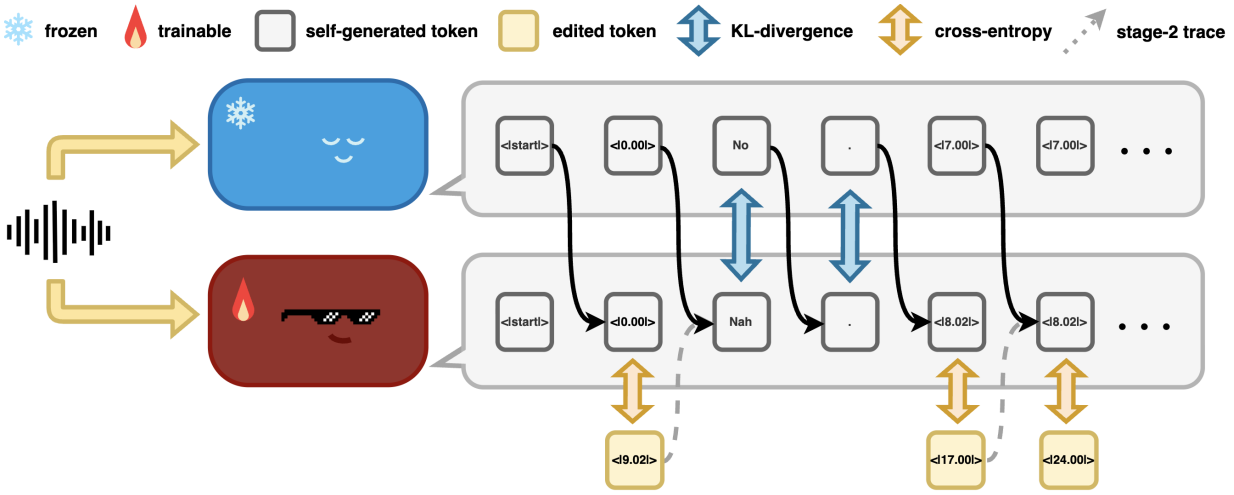

Fig. 1. Overview of the REDDIT framework. A cached base-model replay sequence serves as the decoder context. Timestamp targets are edited based on known synthetic offsets and optimized via cross-entropy loss, while non-timestamp positions are trained to match the corresponding frozen teacher distribution via KL divergence. The Stage-2 trace uses the edited tokens as the teacher-forcing prefix to refine the Stage-1 checkpoint.

text, enabling timestamped transcription without a separate frame-level alignment model [1]; later work extends this interface to explicit word-level timestamp generation [3]. These systems make time a native model output rather than metadata recovered after decoding.

The closest concurrent work is In-Sync, which adapts a speech-aware language model for joint ASR and word-level timestamp prediction [4]. It uses speech-length augmentation, timestamp embedding regularization, and reduced teacher forcing to address heavy-tailed timestamp token values, missing monotonic structure, and error propagation. In-Sync predicts end-of-word timestamps and reports that an explicit silence token degraded performance, leaving silence modeling for future work. Our diagnostic setting instead follows the start/end segment timecode format commonly used by timestamp-token ASR and LALM-style temporal outputs: non-speech is represented by the gap between adjacent end and start timecodes, as well as pre- and post-speech offsets. REDDIT targets this non-speech dimension in a post-training model-editing setting, where an already capable timestamp-producing ASR model is edited and forgetting becomes part of the objective.

Broader speech and audio-language benchmarks have also moved beyond conventional ASR toward instruction-following, regression, sequence generation, and general audio understanding tasks [30]–[45]. Our benchmark is complementary: rather than measuring overall spoken-language capability, it isolates temporal reliability in systems that emit timestamps or timestamp-like temporal outputs.

B. Post-Hoc and Internal Alignment

Many timestamping systems align transcript units after, or alongside, decoding. Forced aligners such as the Montreal Forced Aligner use acoustic and pronunciation constraints [12]–[16], and WhisperX combines VAD segmentation with forced phoneme alignment for long-form Whisper transcription [11]. CrisperWhisper fine-tunes Whisper for more verbatim transcription, but its word-level timestamps still rely on dynamic time warping over decoder cross-attention, attention supervision, and pause heuristics [17]. Useful alignments can also be extracted from selected Whisper cross-attention heads without additional training [18].

These methods ask where transcript units should be placed on the timeline. REDDIT asks whether the decoder's own generated timestamp-token distribution can be edited so that corrected timestamps are produced natively at inference time, without DTW, forced alignment, VAD segmentation, or attention-head selection.

C. Non-Speech Robustness

Non-speech audio can trigger unsupported text, repeated phrases, or boilerplate outputs in autoregressive ASR, especially when language-model priors dominate weak acoustic evidence. Prior work studies Whisper hallucinations under non-speech conditions and small targeted interventions for hallucination reduction [20]–[25]. Timestamp drift is related but not identical: the words may be acoustically supported, yet assigned to the wrong interval.

D. Forgetting-Resistant Editing

Adapting a pretrained speech model can improve a target behavior while damaging behavior outside the adaptation set. Speech adaptation work often uses self-refining synthetic data, domain-adaptive simulation, or parameter-efficient updates to improve robustness under limited data or domain mismatch [26], [27], [47]–[53]. Distillation and learning-without-forgetting objectives preserve prior behavior by matching teacher distributions or retaining previous-task responses [6]–[10]; speech-language evaluation further shows that speech-centric adaptation can harm broader instruction-following ability [28]. REDDIT applies this idea at the token level: timestamp positions receive edited temporal supervision, while non-timestamp positions replay the frozen base distribution under the same cached replay context. This separates temporal correction from recognizer preservation, making the method closer to distribution editing than ordinary timestamp fine-tuning.

III. Method

A. Problem Formulation

Let x be an input speech example and F_{θ_0} be a frozen pretrained autoregressive ASR model. In timestamp mode it generates a sequence over the vocabulary $\mathcal{V} = \mathcal{V}_{\text{text}} \sqcup \mathcal{V}_{\text{time}}$, the disjoint union of lexical and timestamp tokens, where consecutive timestamp tokens delimit decoded segment boundaries.

We define *model-generated timestamp drift* as a mismatch between the decoded and acoustic timelines: around non-speech gaps, the decoder can place acoustically supported words in the wrong time region, a failure distinct from lexical hallucination.

REDDIT corrects this drift from a human-annotation-free correction set of N samples,

$$\mathcal{D}_{\text{corr}} = \{(x_i, b_i)\}_{i=1}^{N}, \tag{1}$$

where $b_i = (b_{i,1}, \ldots, b_{i,2K_i}) \in \mathcal{V}_{\text{time}}^{2K_i}$ is a sequence of timestamp tokens encoding the corrected start/end boundaries of the K_i speech spans in x_i (Sec. III-F). No human transcript or timestamp annotation is required: pseudo-text and output distributions are supplied by F_{θ_0} itself, while b_i supplies only the timestamp edit targets. Figure 1 summarizes how these targets are used across the two stages.

B. Stage 1: Cached Replay Context and Target Editing

For each example we decode once with the frozen base model and cache the trajectory,

$$\tilde{y}_i = F_{\theta_0}(x_i) = (\tilde{y}_{i,1}, \ldots, \tilde{y}_{i,T_i}), \tag{2}$$

keeping it fixed throughout training. Let

$$\mathcal{T}_i = \{\, t \in \{1, \ldots, T_i\} : \tilde{y}_{i,t} \in \mathcal{V}_{\text{time}} \,\} \tag{3}$$

be the set of timestamp positions in the replay, with elements ordered as $t_1 < \cdots < t_{|\mathcal{T}_i|}$. Filtering (Sec. III-F) guarantees $|\mathcal{T}_i| = 2K_i$, so the j-th replay timestamp aligns with the j-th boundary token $b_{i,j}$. We edit the time axis by overwriting each replay timestamp with its corrected target, leaving every other position untouched:

$$y_{i,t}^{\text{tar}} = \begin{cases} b_{i,j}, & t = t_j,\ j = 1, \ldots, 2K_i, \\ \tilde{y}_{i,t}, & t \notin \mathcal{T}_i. \end{cases} \tag{4}$$

In Stage 1, the edited timestamps act *only as targets*: they are never fed back as the decoder input for the next step. The decoder is always conditioned on the cached replay prefix $\tilde{y}_{i,<t}$, so standard teacher forcing is repurposed purely as a context-injection mechanism. Let F_θ denote the student, a trainable copy of the base model initialized from θ_0, with output distribution p_θ. Timestamp supervision then optimizes the student under the replay prefix, $p_\theta(y_{i,t}^{\text{tar}} \mid \tilde{y}_{i,<t}, x_i)$, rather than the edited-prefix objective $p_\theta(y_{i,t}^{\text{tar}} \mid y_{i,<t}^{\text{tar}}, x_i)$. This is an offline approximation to extreme scheduled sampling [29]: the student is not rolled out live.

C. Stage 1 Replay Editing Objective

Student and teacher are scored under the same replay context

$$c_{i,t} = (\tilde{y}_{i,<t}, x_i). \tag{5}$$

Let $\mathcal{T}_i$ be the timestamp positions defined above and $\mathcal{X}_i$ the remaining valid (non-special, non-padding) positions. REDDIT edits the time axis on $\mathcal{T}_i$ with cross-entropy,

$$\mathcal{L}_{\text{time}} = -\frac{1}{\sum_i |\mathcal{T}_i|} \sum_i \sum_{t \in \mathcal{T}_i} \log p_\theta\left(y_{i,t}^{\text{tar}} \mid c_{i,t}\right), \tag{6}$$

and preserves base behavior on $\mathcal{X}_i$ by matching the frozen teacher under the identical context,

$$\mathcal{L}_{\text{text}} = \frac{1}{\sum_i |\mathcal{X}_i|} \sum_i \sum_{t \in \mathcal{X}_i} D_{\text{KL}}(p_{\theta_0}(\cdot \mid c_{i,t}) \,\|\, p_\theta(\cdot \mid c_{i,t})), \tag{7}$$

where $p_{\theta_0}(\cdot \mid c_{i,t})$ is obtained from the frozen base model under the replay context. The Stage-1 objective is

$$\mathcal{L}_{\text{S1}} = \lambda_{\text{time}} \mathcal{L}_{\text{time}} + \lambda_{\text{text}} \mathcal{L}_{\text{text}}. \tag{8}$$

Here, λ_{time} and λ_{text} are hyperparameters balancing the two loss terms.

Stage 1 edits timestamp targets without rewriting the temporal context. Because the decoder prefix is always the cached, potentially drifted replay, the learned timestamp transitions remain conditioned on an offline proxy for model-generated inference prefixes.

D. Stage 2: Edited-Prefix Refinement

The full REDDIT pipeline then runs a short Stage-2 refinement from the Stage-1 checkpoint. Let θ_1 denote the Stage-1 parameters. We form an edited-prefix context from y_i^{tar},

$$c_{i,t}^{\text{edit}} = (y_{i,<t}^{\text{tar}}, x_i). \tag{9}$$

Stage 2 keeps the same correction examples and timestamp targets, and uses the same timestamp-CE and non-timestamp-KL loss forms as Stage 1. The difference is contextual: both terms are computed under $c_{i,t}^{\text{edit}}$, and the non-timestamp KL teacher is the frozen Stage-1 checkpoint F_{θ_1} rather than the frozen base model F_{θ_0}. This stage adds no annotation or inference-time modules; it consolidates the corrected timestamp transitions after Stage 1 has exposed the model to drifted replay prefixes.

E. Model

In both stages, we freeze the encoder, token embeddings, decoder self-attention, decoder feed-forward layers, and output projection, and update only last cross-attn + LNs. Here, LNs denote the layer norms in the last decoder block and the final decoder layer norm.

F. Data Preparation

We describe construction for a generic example and drop the index i. Correction examples are built by VAD-trimming clean speech spans and inserting sampled non-speech audio before, between, and after them [11], [19], yielding *pre-speech*, *inter-speech*, and *post-speech* gaps. Let a_k denote a VAD-trimmed speech span and g_k a non-speech segment; a synthetic example with K spans is

$$x = g_0 \oplus a_1 \oplus g_1 \oplus \cdots \oplus a_K \oplus g_K, \tag{10}$$

with $\oplus$ audio concatenation. The start/end times of the k-th span follow directly from the inserted durations,

$$\tau_k^s = \sum_{r=0}^{k-1} |g_r| + \sum_{r=1}^{k-1} |a_r|, \qquad \tau_k^e = \tau_k^s + |a_k|, \tag{11}$$

and, since the spans are placed by construction, are exact and need no annotation. Quantizing each boundary to its nearest timestamp token via $q(\cdot)$ yields the edit target

$$b = \left(q(\tau_1^s), q(\tau_1^e), \ldots, q(\tau_K^s), q(\tau_K^e)\right) \in \mathcal{V}_{\text{time}}^{2K}. \tag{12}$$

The frozen base model supplies pseudo-text and cached replay timestamps for the same x. Before timestamp editing, we pre-filter the cached teacher replays offline to remove hallucinations, repetitions, boilerplate, empty or unusable text, and structurally inconsistent timestamps. Each retained replay therefore has exactly $2K$ well-ordered timestamp tokens to be overwritten by b.

IV. Experiments

We evaluate three questions: how severe generated timestamp drift is under controlled non-speech gaps, whether REDDIT can correct it with limited correction data, and whether the correction preserves non-target ASR behavior.

A. Evaluation Data

We use Common Voice zh-TW audio [2] for targeted correction data and in-domain evaluation. The *Gap* split (`test_gap`) concatenates two or three utterances with inserted non-speech regions, testing multi-segment temporal tracking. The *Long-Gap* split (`test_long_gap`) contains one utterance with a long leading or trailing non-speech region, isolating onset and offset anchoring.

Both in-domain evaluation splits contain 5,105 examples capped at 30 s. Reference timestamps are exact because they are computed from the splicing schedule. Non-speech audio is sampled from a held-out

TABLE I
TARGET CORRECTION AND IN-DOMAIN TIMESTAMP BENCHMARK DATA FROM WTFO/CMV_TIME_SHIFT. HOURS ARE COMPUTED FROM AUDIO AND TIMESTAMP LABELS.

Role	Samples	Audio (h)	Speech (h)	Non-speech (h)	Speech spans
Train	7,367	34.9	7.3	27.6	1–3
Gap eval	5,105	21.3	7.0	14.3	2–3
Long-gap eval	5,105	30.5	3.1	27.5	1

TABLE II
OUT-OF-DOMAIN RETENTION AND MIXED-GAP EVALUATION DATA. N IS SAMPLE COUNT; H IS DURATION IN HOURS. GAP SETS ARE GAP-STRESSED COUNTERPARTS OF THE NO-GAP SETS. CV-EN-MIN DENOTES COMMONVOICE-EN-MIN.

Dataset	No-gap N	No-gap h	Gap N	Gap h	Non-speech h
ASCEND-zh	578	0.3	578	2.6	2.1
ASCEND-en	214	0.1	214	1.0	0.8
ASCEND-mix	373	0.4	373	2.0	1.3
CV-en-min	2,997	3.2	2,997	16.7	10.8
Total	4,162	4.0	4,162	22.4	15.0

TABLE III
CONFIGURATION OF COMPARED WHISPER-TINY METHODS. LNS DENOTE THE LAST DECODER-BLOCK LAYER NORMS PLUS THE FINAL DECODER LAYER NORM; REDDIT FULL USES THE SAME SCOPE IN BOTH STAGES.

Method	Trainable scope	Trainable params
Base Whisper-tiny	0	0
SFT	last cross-attn + LNs	0.59M (1.6%)
SFT timestamp-only	last cross-attn + LNs	0.59M (1.6%)
SFT decoder	decoder	29.38M (77.8%)
Edited-time replay	last cross-attn + LNs	0.59M (1.6%)
Edited-time replay decoder	decoder	29.38M (77.8%)
Reduced Teacher Forcing	last cross-attn + LNs	0.59M (1.6%)
Replayed Reduced Teacher Forcing	last cross-attn + LNs	0.59M (1.6%)
REDDIT Stage 1 (Ours)	last cross-attn + LNs	0.59M (1.6%)
REDDIT Full (Ours)	last cross-attn + LNs	0.59M (1.6%)

non-speech pool, and test non-speech and speech sources are disjoint from training. Table I summarizes the targeted correction and in-domain benchmark splits.

To test retention and transfer, we also evaluate ASCEND-ZH, ASCEND-EN, ASCEND-MIXED [54], and CommonVoice-EN-min [2] in two versions: the original no-gap split and a gap stress split constructed with the same gap-insertion procedure. Table II summarizes these OOD evaluation sets.

B. Systems and Baselines

The diagnostic benchmark covers timestamp-producing ASR and audio-language systems, including Whisper-family models [1], Distil-Whisper [55], Qwen audio/omni models [56]–[58], MOSS-Audio models [59], VibeVoice-ASR [60], and DeSTA-style audio-language models [61]. Intervention experiments focus on Whisper-tiny to isolate whether a small timestamp-token ASR update can repair drift without adding inference-time alignment.

Baselines include base Whisper-tiny, supervised timestamp-corrected fine-tuning (SFT), timestamp-only fine-tuning, edited-context replay, REDDIT Stage 1, REDDIT Full, and an In-Sync-inspired reduced teacher forcing (RTF) variant adapted to Whisper-tiny. RTF corrupts the previous timestamp input with probability $p_{\mathrm{rtf}} = 0.2$ at timestamp-target positions while leaving text inputs unchanged. This is a baseline inspired by In-Sync [4], not a full reimplementation of its length augmentation, embedding regularization, or end-of-word timestamp formulation. REDDIT Stage 1 isolates replay-conditioned distribution editing; REDDIT Full is the complete two-stage pipeline, which starts from the Stage-1 checkpoint and further trains on the same correction data with edited-prefix refinement.

Table III summarizes the trainable scope used by the Whisper-tiny interventions.

C. Post-Training Setup

All timestamp intervention runs use 34.9 hours of targeted correction audio after cached-replay pre-filtering. Stage 1 uses the same training budget as the single-stage baselines, and REDDIT Full adds a short Stage-2 refinement on the same correction data. We select the reported full-pipeline checkpoint from this short refinement trajectory by validation behavior rather than treating a fixed step count as part of the method. We freeze the encoder, token embeddings, decoder self-attention, decoder feed-forward layers, and output projection, updating only last cross-attn + LNs. This scope contains 0.59M of 37.76M Whisper-tiny parameters (1.6%). The reported Whisper-large-v3 REDDIT runs use the same last-layer scope, corresponding to 6.57M of 1.54B parameters (0.43%).

We use AdamW, learning rate 1×10^{-5} after 10 warmup steps, and batch size 64 for all post-training runs. Single-stage baselines and REDDIT Stage 1 use the same duration. REDDIT Stage 1 uses $\lambda_{\mathrm{time}} = 1$ and $\lambda_{\mathrm{text}} = 5$. Stage 2 uses the same timestamp-CE and non-timestamp-KL loss forms under edited-prefix context, with the frozen Stage-1 checkpoint as the KL teacher. The replay KL uses temperature 1.0, no top-k truncation, and online teacher logits under the corresponding replay or edited-prefix context.

D. Evaluation Metrics

We report a compact diagnostic set in the main paper tables.

a) Temporal overlap.: For a matched hypothesis segment $p = [s_p, e_p]$ and reference segment $r = [s_r, e_r]$,

$$\mathrm{tIoU}(p, r) = \frac{|p \cap r|}{|p \cup r|}.$$

We report mean tIoU (mIoU) over matched pairs and omit match rate from the main tables because textual matching alone does not imply temporal overlap.

b) Boundary error.: *Start MAE* and *End MAE* measure onset and offset errors over matched segments, in seconds. Start MAE captures subtitles appearing too early or late after non-speech gaps; end MAE captures subtitles ending too early or extending into silence.

c) Token-level alignment.: We report AAS in milliseconds:

$$\mathrm{AAS} = \frac{\mathrm{ASTD} + \mathrm{AETD}}{2},$$

where ASTD and AETD are start- and end-time deviations over edit-distance paired text units. Insertions and deletions are excluded. MAL is the percentage of samples without a valid paired timed token, preventing low AAS from hiding missing alignments.

d) Large drift and hallucination.: Metrics *Drift>5s* and *Drift>10s* are the percentages of matched segments with absolute temporal displacement above 5 or 10 seconds. Hallucination Rate is a file-level diagnostic based on common hallucination phrases and repeated n-grams for $n \in \{3, 4, 5\}$, following prior non-speech hallucination analyses [20], [21]; it is reported separately because supported speech can be temporally misplaced without being hallucinated.

e) Retention.: For non-target ASR retention, we report MER according to benchmark convention. These text metrics are not timestamp metrics; they measure whether timestamp adaptation preserves recognition behavior outside the correction set.

E. Ablations

Ablations isolate timestamp supervision, replay, decoder context, and reduced teacher forcing. The key comparison is whether corrected timestamps are learned under teacher-forced edited prefixes or under cached base-model replay prefixes, with non-timestamp behavior preserved by distribution replay.

TABLE IV

TIMESTAMP AND NON-SPEECH RESULTS FOR ALL RETAINED MODEL FAMILIES. EACH PAIRED CELL REPORTS GAP / LONG-GAP. DRIFT AND HALLUCINATION COLUMNS ARE FILE-LEVEL RATES, WITH RAW COUNTS RETAINED IN THE SOURCE. GREEN AND RED MARK BEST AND WORST VALUES.

System	mIoU % ↑	MER % ↓	AAS (ms) ↓	MAL % ↓	Start MAE (s) ↓	End MAE (s) ↓	Drift>5s % ↓	Drift>10s % ↓	Halluc. % ↓
MOSS-Audio-4B	**81.0** / **85.0**	39.5 / 34.6	521 / 377	**0.0** / **0.1**	**0.23** / **0.12**	0.38 / 0.28	2.4 / 0.3	0.1 / 0.1	**5.9** / **0.5**
MOSS-Audio-8B	76.3 / 63.5	31.7 / 32.5	468 / 2849	0.5 / 1.7	0.63 / 3.59	0.36 / 0.29	11.2 / 25.5	0.1 / 15.8	7.0 / 3.1
Qwen3-Omni-30B	76.9 / 81.4	22.0 / **9.3**	814 / **300**	3.1 / 1.3	0.31 / 0.26	**0.31** / 0.23	0.7 / 0.3	0.0 / 0.2	9.5 / 2.4
VibeVoice-ASR	71.2 / 84.1	42.5 / 48.1	**465** / 568	3.8 / 12.8	0.87 / 0.48	0.80 / 0.92	9.5 / 0.9	0.8 / 0.4	11.5 / 14.4
Whisper-tiny	63.0 / 38.7	79.9 / **347.2**	951 / 4806	5.4 / 17.5	1.30 / 7.34	0.46 / 0.28	15.2 / 36.9	0.4 / 26.7	13.0 / 19.2
Whisper-base	61.8 / 39.8	56.9 / 232.8	1063 / 5171	3.0 / 13.2	1.48 / 7.50	0.42 / **0.21**	21.9 / 36.7	0.5 / 25.4	10.2 / 17.7
Whisper-medium	64.8 / 45.2	97.5 / 324.5	1209 / 4544	2.7 / 5.9	1.16 / 6.00	0.39 / 0.47	13.3 / 30.4	0.9 / 23.3	13.9 / 28.5
Whisper-large-v2	65.2 / 45.8	64.9 / 139.6	1334 / 4687	1.5 / 3.6	0.98 / 5.80	0.40 / 0.49	9.3 / 30.9	0.8 / 22.8	16.7 / 25.7
Whisper-large-v3	47.6 / 33.8	**20.7** / 39.9	1636 / 4099	0.2 / 0.5	**2.93** / 8.05	0.52 / **1.52**	**53.4** / 58.3	**5.1** / 44.8	8.4 / 8.7
Whisper-large-v3-turbo	49.4 / **32.5**	22.8 / 33.7	1518 / 4522	0.9 / 4.2	2.62 / **8.41**	0.48 / 1.35	46.8 / **58.6**	2.5 / **45.2**	8.8 / 5.7
Distil-Whisper-large-v3	48.5 / 39.1	**107.3** / 140.8	**5017** / 6911	9.1 / 8.2	1.18 / 4.22	1.10 / 1.29	16.8 / 30.8	3.1 / 23.7	**51.0** / **41.3**
Qwen2-Audio-7B-Instruct	49.0 / 66.9	58.3 / 122.2	2159 / 3743	**23.6** / **28.9**	1.10 / 0.76	1.17 / 0.98	2.5 / 0.1	**0.0** / **0.0**	13.9 / 20.1
Qwen2.5-Omni-3B	**45.8** / 57.3	50.7 / 48.8	2181 / 2773	23.4 / 15.3	1.43 / 1.34	**1.36** / 1.42	19.0 / 3.4	0.1 / 1.0	19.9 / 7.4
Qwen2.5-Omni-7B	59.8 / 64.6	39.8 / 46.1	1393 / 2784	5.4 / 5.0	0.99 / 1.63	0.89 / 1.23	8.2 / 5.7	0.1 / 2.5	27.8 / 11.9
DeSTA2.5-Audio-LLaMA-8B	46.4 / 66.2	41.6 / 32.6	4232 / **8992**	2.0 / 2.7	1.31 / 1.05	1.23 / 1.14	**0.4** / **0.1**	0.1 / 0.0	6.9 / 7.8

TABLE V

WHISPER-TINY TIMESTAMP CORRECTION ON THE SELF-BUILT GAP TEST SETS. EACH PAIRED METRIC REPORTS GAP / LONG-GAP. HALLUC. RATE IS FILE-LEVEL.

Method	mIoU % ↑	MER % ↓	AAS (ms) ↓	MAL % ↓	Start MAE (s) ↓	End MAE (s) ↓	Drift>5s % ↓	Drift>10s % ↓	Halluc. % ↓
Base Whisper-tiny	63.0 / **38.7**	79.9 / **347.2**	951 / **4806**	**5.4** / **17.5**	1.30 / **7.34**	0.46 / 0.28	**15.2** / 36.9	**0.4** / 26.7	13.0 / 19.2
SFT	92.7 / 93.5	38.3 / 52.2	72 / 186	0.2 / 0.3	0.15 / 0.27	0.21 / 0.26	0.5 / 2.0	**0.0** / 1.4	6.3 / 2.0
SFT timestamp-only	**53.0** / 70.0	**100.8** / 144.9	289 / 273	0.3 / 0.4	1.45 / 0.16	**2.32** / 0.81	5.0 / 0.9	0.0 / 0.7	**39.1** / **31.7**
SFT decoder	**95.3** / **96.3**	**35.7** / **43.4**	**46** / **37**	**0.1** / 0.1	**0.07** / 0.03	**0.11** / **0.06**	**0.2** / **0.0**	**0.0** / **0.0**	6.7 / **0.5**
Edited-time replay	58.2 / 47.4	73.2 / 306.0	952 / 3840	1.3 / 4.9	**1.63** / 5.93	.83 / **2.74**	8.7 / 43.3	0.2 / 32.3	12.8 / 19.1
Edited-time replay decoder	63.7 / 55.5	74.0 / 304.8	743 / 3081	1.3 / 4.7	1.36 / 4.84	0.72 / 2.44	6.6 / 35.6	0.3 / 26.9	13.6 / 18.0
Reduced Teacher Forcing	93.0 / 94.8	39.0 / 45.9	84 / 96	0.3 / 0.1	0.15 / 0.13	0.24 / 0.11	0.6 / 0.8	0.1 / 0.6	**6.1** / 0.8
Replayed Reduced Teacher Forcing	58.2 / 47.2	76.3 / 304.0	**961** / 3869	1.4 / 5.0	**1.63** / 5.99	.81 / 2.73	8.6 / **43.6**	0.2 / **32.7**	12.8 / 19.0
REDDIT Stage 1 (Ours)	93.0 / 94.6	45.1 / 56.6	69 / 74	0.4 / 0.4	0.16 / 0.07	0.22 / 0.13	0.4 / 0.4	0.1 / 0.3	8.3 / 2.0
REDDIT Full (Ours)	93.3 / 95.0	45.2 / 53.9	66 / 66	0.3 / 0.3	0.15 / 0.06	0.22 / 0.11	0.4 / 0.2	0.1 / 0.2	8.3 / 1.8

TABLE VI

OUT-OF-DOMAIN ASR RETENTION ON MIXED-GAP OOD EVALUATION SETS. MIXED-GAP COMBINES SINGLE-UTTERANCE LONG-GAP AND MULTI-UTTERANCE GAP SAMPLES. AAS IS SAMPLE-WEIGHTED; ERROR COLUMNS ARE MER %.

Method	AAS (ms) ↓	CV-en MER % ↓	ASCEND-zh MER % ↓	ASCEND-en MER % ↓	ASCEND-mixed MER % ↓
Base Whisper-tiny	**2752**	49.1	**136.9**	140.6	126.7
SFT	280	89.1	61.8	299.2	85.7
SFT timestamp-only	311	55.0	106.1	121.7	110.0
SFT decoder	315	**362.4**	49.9	324.5	92.6
Edited-time replay	2733	48.6	130.9	132.1	139.6
Edited-time replay decoder	2599	48.3	121.8	115.3	132.2
Reduced Teacher Forcing	301	241.4	69.1	**422.2**	97.9
Replayed Reduced Teacher Forcing	2734	49.6	132.1	135.0	**144.3**
REDDIT Stage 1 (Ours)	228	**38.7**	**48.1**	59.7	**65.7**
REDDIT Full (Ours)	**223**	38.8	58.2	**59.5**	70.2

TABLE VII

OUT-OF-DOMAIN ASR RETENTION ON NO-GAP OOD EVALUATION SETS. ERROR COLUMNS ARE MER %.

Method	CV-en MER % ↓	ASCEND-zh MER % ↓	ASCEND-en MER % ↓	ASCEND-mixed MER % ↓
Base Whisper-tiny	37.0	**53.9**	45.5	69.2
SFT	61.6	58.5	303.8	79.9
SFT timestamp-only	**34.7**	76.5	**44.4**	90.0
SFT decoder	**524.2**	70.5	**710.5**	**120.3**
Edited-time replay	37.5	56.9	45.8	73.6
Edited-time replay decoder	36.1	56.0	45.9	71.8
Reduced Teacher Forcing	105.6	**166.9**	450.5	112.4
Replayed Reduced Teacher Forcing	39.8	54.0	45.2	64.5
REDDIT Stage 1 (Ours)	38.9	58.9	47.2	**58.0**
REDDIT Full (Ours)	41.3	63.5	47.2	60.1

V. RESULTS

A. Non-Speech Gaps Expose Timestamp Drift

Table IV shows that transcript quality does not guarantee temporal grounding. Whisper-family models often produce recognizable text while long-gap start MAE and large-drift rates increase sharply; for example, Whisper-tiny mIoU drops from 63.0% to 38.7%, while start MAE rises from 1.30 s to 7.34 s. The strongest lexical systems are not automatically the strongest temporal systems: Whisper-large-v3 has the best gap MER among Whisper models, but it also has the worst gap Drift>5s rate. These results support evaluating timestamp drift as an acoustic-grounding error rather than as a byproduct of transcript quality. The table also separates drift from hallucination: large temporal displacements measure misplaced supported speech, whereas hallucination rate measures unsupported loops or boilerplate.

TABLE VIII
REDDIT MODEL-SCALE ABLATION. THE VERTICAL LINE SEPARATES IN-DOMAIN METRICS (LEFT) FROM OOD METRICS (RIGHT). TARGET (IN-DOMAIN) METRICS ARE GAP / LONG-GAP; OOD METRICS ARE NO-GAP / MIXED-GAP. CV: COMMON-VOICE; AS: ASCEND. CV AND AS COLUMNS REPORT MER %.

Method	mIoU % ↑	MER % ↓	AAS (ms) ↓	Drift>10s % ↓	CV-en % ↓	AS-zh % ↓	AS-mix % ↓	OOD AAS ↓
Whisper-tiny	63.0 / 38.7	79.9 / 347.2	951 / 4806	0.4 / 26.7	**37.0** / 49.1	**53.9** / 136.9	69.2 / 126.7	199 / 2752
REDDIT Stage 1	93.0 / 94.6	**45.1** / 56.6	69 / 74	**0.1** / 0.3	38.9 / **38.7**	58.9 / **48.1**	**58.0** / **65.7**	144 / 228
REDDIT Full	**93.3** / **95.0**	45.2 / **53.9**	**66** / **66**	**0.1** / **0.2**	41.3 / 38.8	63.5 / 58.2	60.1 / 70.2	**142** / **223**
Whisper-large-v3	47.6 / 33.8	**20.7** / 39.9	1636 / 4099	5.1 / 44.8	17.6 / 17.0	18.9 / **24.9**	**19.9** / **23.1**	**121** / 2393
REDDIT large-v3 Stage 1	81.4 / 83.3	37.2 / 49.0	658 / 462	3.1 / **0.4**	15.9 / 16.9	19.9 / 36.7	21.3 / 36.0	141 / **560**
REDDIT large-v3 Full	**81.9** / **84.0**	21.9 / **28.5**	**329** / **337**	**0.7** / **0.4**	**15.7** / **15.0**	**18.7** / 25.6	20.9 / 25.2	123 / 567

B. REDDIT Corrects Timestamp Drift

Table V evaluates Whisper-tiny interventions. SFT decoder tuning obtains the best target timestamp accuracy, but it does so by updating a much larger decoder scope and later degrades severely on OOD ASR. The restricted-scope baselines show that parameter efficiency alone is not enough: SFT timestamp-only, edited-time replay, and replayed RTF leave substantial long-gap drift. REDDIT Full reaches SFT-level temporal correction while keeping the replay-conditioned Stage-1 edit as its first step: long-gap mIoU improves from 38.7% to 95.0%, AAS from 4806 ms to 66 ms, and Drift>10s from 26.7% to 0.2%. It also reduces long-gap hallucination rate from 19.2% to 1.8%, which we treat as an auxiliary non-speech robustness effect rather than the definition of drift. The small gap between REDDIT Stage 1 and REDDIT Full indicates that replay-conditioned editing already supplies the main correction signal, while Stage 2 mainly consolidates the corrected prefix behavior.

C. OOD Robustness and Retention

Table VI tests mixed-gap OOD transfer. REDDIT Full reduces base mixed-gap AAS from 2752 ms to 223 ms while avoiding the recognition degradation of SFT and RTF. This tradeoff is the central retention result: teacher-forced timestamp correction can improve timing, but it does not consistently preserve recognition outside the correction distribution. Table VII confirms ordinary no-gap retention: REDDIT Full stays close to the base model on CV-en and ASCEND-en, whereas SFT decoder tuning reaches 524.2% CV-en MER and 710.5% ASCEND-en MER. The OOD numbers therefore support the distribution-editing view of the task: timestamp positions should be moved, while non-timestamp behavior should remain anchored to the base model under the same replay context.

D. Ablation Summary

The target and OOD tables isolate decoder context, replay, and reduced teacher forcing. Teacher-forced RTF is strong on the target timestamp set, but degrades on mixed-gap OOD recognition. Replayed RTF shows that adding replay to an In-Sync-inspired objective is insufficient by itself: without the non-timestamp KL anchor, the model can still move away from its base lexical behavior. REDDIT's Stage-1 advantage is the combination of cached replay context, edited timestamp targets, and KL retention on non-timestamp positions.

Stage 1 isolates the core replay-conditioned timestamp editing effect, while REDDIT Full adds the edited-prefix refinement used by the complete pipeline. On Whisper-tiny, Stage 2 gives modest additional timing and OOD alignment gains; on Whisper-large-v3, the REDDIT rows show that the two-stage schedule is more important for scaling. The Stage-1 large-v3 row already repairs much of the temporal displacement, but REDDIT Full improves target long-gap MER and most OOD MER values after the Stage-1 checkpoint, while mixed-gap OOD AAS remains close to Stage 1 and far below the base model. This suggests that edited-prefix refinement helps larger decoders use the corrected timestamp trajectory while retaining the Stage-1 replay anchor.

VI. CONCLUSION

We studied non-speech-induced drift in model-generated timestamps: autoregressive ASR systems can transcribe recognizable speech while placing it on the wrong part of the audio timeline. Controlled gap and long-gap evaluations across timestamp-producing ASR and audio-language systems show that this error is distinct from transcript error and hallucination. Naive timestamp-corrected fine-tuning can repair target alignment but causes severe forgetting on non-target ASR data, motivating timestamp adaptation as a distribution-editing problem. REDDIT addresses this with a two-stage pipeline: replay-conditioned timestamp editing preserves the frozen base distribution on non-timestamp tokens, and a short edited-prefix refinement consolidates the corrected timestamp behavior. On Whisper-tiny, this pipeline improves long-gap timestamp robustness and mixed-gap OOD alignment without adding inference-time VAD, forced alignment, or post-processing. On Whisper-large-v3, the Stage-2 refinement becomes more important, suggesting that REDDIT is a scalable strategy for preserving recognition while editing generated timestamps.

A. Future Work

Our intervention study focuses on explicit timestamp-token ASR, where token-level editing is well defined. Extending REDDIT to audio-language models is more complex because temporal information may be expressed in free-form text and non-ASR abilities such as speech instruction following and audio reasoning must be preserved. Future work should evaluate forgetting-resistant temporal editing for LALMs under broader protocols, including Speech-IFEval-style tests [28], and analyze how timestamp editing changes cross-attention and acoustic grounding.

REFERENCES

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," arXiv preprint arXiv:2212.04356, 2022.

[2] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common Voice: A massively-multilingual speech corpus," arXiv preprint arXiv:1912.06670, 2019.

[3] K. Hu, K. Puvvada, E. Rastorgueva, Z. Chen, H. Huang, S. Ding, K. Dhawan, H. Xu, J. Balam, and B. Ginsburg, "Word level timestamp generation for automatic speech recognition and translation," arXiv preprint arXiv:2505.15646, 2025.

[4] X. Fan, V. Sunder, S. Thomas, M. Hasegawa-Johnson, B. Kingsbury, and G. Saon, "In-Sync: Adaptation of speech aware large language models for ASR with word level timestamp predictions," arXiv preprint arXiv:2604.22817, 2026.

[5] H. Wang, Y. Li, S. Ma, H. Liu, and X. Wang, "Listening between the frames: Bridging temporal gaps in large audio-language models," arXiv preprint arXiv:2511.11039, 2025.

[6] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015.

[7] Z. Li and D. Hoiem, "Learning without forgetting," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.

[8] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, et al., "Overcoming catastrophic forgetting in neural networks," Proceedings of the National Academy of Sciences, 2017.

[9] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. NeurIPS*, 2017.

[10] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: A strong, simple baseline," in *Proc. NeurIPS*, 2020.

[11] M. Bain, J. Huh, T. Han, and A. Zisserman, "WhisperX: Time-accurate speech transcription of long-form audio," INTERSPEECH, 2023.

[12] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable text-speech alignment using Kaldi," INTERSPEECH, 2017.

[13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, et al., "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[14] R. Rousso, E. Cohen, J. Keshet, and E. Chodroff, "Tradition or innovation: A comparison of modern ASR methods for forced alignment," arXiv preprint arXiv:2406.19363, 2024.

[15] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "CTC-segmentation of large corpora for German end-to-end speech recognition," arXiv preprint arXiv:2007.09127, 2020.

[16] B. Mu, X. Shi, X. Wang, H. Liu, J. Xu, and L. Xie, "LLM-ForcedAligner: A non-autoregressive and accurate LLM-based forced aligner for multilingual and long-form speech," arXiv preprint arXiv:2601.18220, 2026.

[17] L. Wagner, B. Thallinger, and M. Zusag, "CrisperWhisper: Accurate timestamps on verbatim speech transcriptions," arXiv preprint arXiv:2408.16589, 2024.

[18] S.-L. Yeh, Y. Meng, and H. Tang, "Whisper has an internal word aligner," arXiv preprint arXiv:2509.09987, 2025.

[19] Z.-H. Tan, A. K. Sarkar, and N. Dehak, "rVAD: An unsupervised segment-based robust voice activity detection method," arXiv preprint arXiv:1906.03588, 2019.

[20] M. Barański, J. Jasiński, J. Bartolewska, S. Kacprzak, M. Witkowski, and K. Kowalczyk, "Investigation of Whisper ASR hallucinations induced by non-speech audio," ICASSP, 2025.

[21] Y. Wang, A. Alhmoud, S. Alsahly, M. Alqurishi, and M. Ravanelli, "Calm-Whisper: Reduce Whisper hallucination on non-speech by calming crazy heads down," arXiv preprint arXiv:2505.12969, 2025.

[22] H. Atwany, A. Waheed, R. Singh, M. Choudhury, and B. Raj, "Lost in transcription, found in distribution shift: Demystifying hallucination in speech foundation models," arXiv preprint arXiv:2502.12414, 2025.

[23] X. Cheng, D. Fu, C. Wen, S. Yu, Z. Wang, S. Ji, S. Arora, T. Jin, S. Watanabe, and Z. Zhao, "AHa-Bench: Benchmarking audio hallucinations in large audio-language models," in *Proc. NeurIPS*, 2025.

[24] J. Jasiński, M. Barański, J. Bartolewska, M. Witkowski, and K. Kowalczyk, "From text metrics to model internals: A study of Whisper ASR hallucination detection," arXiv preprint arXiv:2606.23060, 2026.

[25] G. Aparin, V. Popov, T. Sadekova, and A. Yermekova, "Whisper hallucination detection and mitigation via hidden representation steering and sparse autoencoders," arXiv preprint arXiv:2606.07473, 2026.

[26] C.-K. Chou, C.-J. Hsu, H.-L. Chung, L.-H. Tseng, H.-C. Cheng, Y.-K. Fu, K. P. Huang, and H.-Y. Lee, "A self-refining framework for enhancing ASR using TTS-synthesized data," arXiv preprint arXiv:2506.11130, 2025.

[27] C.-C. Wang, L.-W. Chen, C.-K. Chou, H.-S. Lee, B. Chen, and H.-M. Wang, "Channel-aware domain-adaptive generative adversarial network for robust speech recognition," ICASSP, 2025.

[28] K.-H. Lu, C.-Y. Kuan, and H.-Y. Lee, "Speech-IFEval: Evaluating instruction-following and quantifying catastrophic forgetting in speech-aware language models," arXiv preprint arXiv:2505.19037, 2025.

[29] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," Advances in Neural Information Processing Systems, 2015.

[30] C.-Y. Huang, W.-C. Chen, S.-W. Yang, A. T. Liu, C.-A. Li, Y.-X. Lin, W.-C. Tseng, et al., "Dynamic-SUPERB Phase-2: A collaboratively expanding benchmark for measuring the capabilities of spoken language models with 180 tasks," in *Proc. ICLR*, 2025.

[31] S.-W. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, et al., "SUPERB: Speech processing universal performance benchmark," in *Proc. INTERSPEECH*, 2021.

[32] J. Shi, D. Berrebbi, W. Chen, H.-L. Chung, E.-P. Hu, W.-P. Huang, X. Chang, et al., "ML-SUPERB: Multilingual speech universal performance benchmark," in *Proc. INTERSPEECH*, 2023.

[33] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, et al., "HEAR: Holistic evaluation of audio representations," in *Proc. NeurIPS Competitions and Demonstrations Track*, 2022.

[34] Q. Yang, J. Xu, W. Liu, Y. Chu, Z. Jiang, X. Zhou, et al., "AIR-Bench: Benchmarking large audio-language models via generative comprehension," arXiv preprint arXiv:2402.07729, 2024.

[35] B. Wang, X. Zou, G. Lin, S. Sun, Z. Liu, W. Zhang, Z. Liu, A. Aw, and N. F. Chen, "AudioBench: A universal benchmark for audio large language models," arXiv preprint arXiv:2406.16020, 2024.

[36] S. Sakshi, U. Tyagi, S. Kumar, A. Seth, R. Selvakumar, O. Nieto, R. Duraiswami, S. Ghosh, and D. Manocha, "MMAU: A massive multi-task audio understanding and reasoning benchmark," in *Proc. ICLR*, 2025.

[37] P. He, Z. Wen, Y. Wang, Y. Wang, X. Liu, J. Huang, Z. Lei, et al., "AudioMarathon: A comprehensive benchmark for long-context audio understanding and efficiency in audio LLMs," arXiv preprint arXiv:2510.07293, 2025.

[38] J. Yao, S. Liu, Y. Wang, R. Cheng, L. Mei, B. Bi, Z. Xiong, and X. Cheng, "Not in Sync: Unveiling temporal bias in audio chat models," arXiv preprint arXiv:2510.12185, 2025.

[39] L. Sun, X. Zhou, Z. Li, Y. Zhang, Y. Wang, and W. Xie, "SpotSound: Enhancing large audio-language models with fine-grained temporal grounding," arXiv preprint arXiv:2604.13023, 2026.

[40] K. Luo, L. Lin, Y. Zhang, M. Aloqaily, J. Tao, D. Wang, et al., "ChronosAudio: A comprehensive long-audio benchmark for evaluating audio-large language models," arXiv preprint arXiv:2601.04876, 2026.

[41] Z. Liu, Z. Niu, Q. Xiao, Z. Zheng, R. Yuan, Y. Zang, Y. Cao, et al., "STAR-Bench: Probing deep spatio-temporal reasoning as audio 4D intelligence," arXiv preprint arXiv:2510.24693, 2025.

[42] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, 2017.

[43] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proc. NAACL-HLT*, 2019.

[44] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *Proc. ICASSP*, 2020.

[45] S. Lipping, P. Sudarsanam, K. Drossos, and T. Virtanen, "Clotho-AQA: A crowdsourced dataset for audio question answering," in *Proc. EUSIPCO*, 2022.

[46] Y.-X. Luo, Y.-C. Lin, M.-T. Chuang, J.-H. Chen, I.-N. Tsai, P. X. Kiew, Y.-H. Huang, C.-F. Liu, Y.-C. Chen, B.-H. Feng, W. Ren, and H.-Y. Lee, "ToxicTone: A Mandarin audio dataset annotated for toxicity and toxic utterance tonality," in *Proc. INTERSPEECH*, 2025.

[47] S.-H. Wang, Z.-C. Chen, J. Shi, M.-T. Chuang, G.-T. Lin, K.-P. Huang, D. Harwath, S.-W. Li, and H.-Y. Lee, "How to learn a new language? An efficient solution for self-supervised learning models unseen languages adaption in low-resource scenario," arXiv preprint arXiv:2411.18217, 2024.

[48] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proc. ICML*, 2019.

[49] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2022.

[50] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. ACL*, 2021.

[51] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proc. EMNLP*, 2021.

[52] E. Ben-Zaken, S. Ravfogel, and Y. Goldberg, "BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proc. ACL*, 2022.

[53] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. Raffel, "Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning," in *Proc. NeurIPS*, 2022.

[54] H. Lovenia, S. Cahyawijaya, G. I. Winata, P. Xu, X. Yan, Z. Liu, R. Frieske, T. Yu, W. Dai, E. J. Barezi, Q. Chen, X. Ma, B. E. Shi, and P. Fung, "ASCEND: A spontaneous Chinese-English dataset for code-switching in multi-turn conversation," in *Proc. LREC*, 2022.

[55] S. Gandhi, P. von Platen, and A. M. Rush, "Distil-Whisper: Robust knowledge distillation via large-scale pseudo labelling," arXiv preprint arXiv:2311.00430, 2023.

[56] Y. Chu, J. Xu, Q. Yang, H. Wei, X. Wei, Z. Guo, Y. Leng, Y. Lv, J. He, J. Lin, C. Zhou, and J. Zhou, "Qwen2-Audio technical report," arXiv preprint arXiv:2407.10759, 2024.

[57] J. Xu, Z. Guo, J. He, H. Hu, T. He, S. Bai, K. Chen, J. Wang, Y. Fan, K. Dang, B. Zhang, X. Wang, Y. Chu, and J. Lin, "Qwen2.5-Omni technical report," arXiv preprint arXiv:2503.20215, 2025.

[58] J. Xu, Z. Guo, H. Hu, Y. Chu, X. Wang, J. He, Y. Wang, X. Shi, T. He, X. Zhu, Y. Lv, Y. Wang, D. Guo, et al., "Qwen3-Omni technical report," arXiv preprint arXiv:2509.17765, 2025.
[59] C. Yang, C. Yu, H. Chen, J. Zhu, J. Chen, K. Chen, W. Wang, Y. Wang, Y. Jiang, Y. Jiang, Z. Lin, Z. Chen, Z. Fei, et al., "MOSS-Audio technical report," arXiv preprint arXiv:2606.01802, 2026.
[60] Z. Peng, J. Yu, Y. Chang, Z. Wang, L. Dong, Y. Hao, Y. Tu, C. Yang, W. Wang, S. Xu, Y. Sun, H. Bao, W. Xu, et al., "VIBEVOICE-ASR technical report," arXiv preprint arXiv:2601.18184, 2026.
[61] K.-H. Lu, Z. Chen, S.-W. Fu, C.-H. H. Yang, S.-F. Huang, C.-K. Yang, C.-E. Yu, C.-W. Chen, W.-C. Chen, C.-Y. Huang, Y.-C. Lin, Y.-X. Lin, C.-A. Fu, et al., "DeSTA2.5-Audio: Toward general-purpose large audio language model with self-generated cross-modal alignment," arXiv preprint arXiv:2507.02768, 2025.