

Abstract Compression of Conversational Audio Context for LLM-based ASR

Shashi Kumar^{1,2}, Esaú Villatoro-Tello¹, Sergio Burdisso¹, Kadri Hacioglu³, Thibault Bañeras-Roux¹, Hasindri Watawana^{1,2}, Dairazalia Sanchez-Cortes¹, Srikanth Madikeri⁴, Petr Motlicek^{1,5}, Andreas Stolcke³

¹Idiap Research Institute, Switzerland; ²EPFL, Switzerland; ³Uniphore, U.S.A.
⁴University of Zurich, Switzerland; ⁵Brno University of Technology, Czech Republic

Abstract—Standard LLM-based speech recognition systems process utterances in isolation, limiting their ability to leverage conversational context. Here we study whether multimodal context from prior turns improves LLM-based ASR and how to represent that context efficiently. We find that, after supervised multi-turn training, conversational context mainly helps with the recognition of contextual entities. However, conditioning on raw context is expensive because the prior-turn audio embedding sequence grows rapidly with conversation length. To address this, we propose Abstract Compression, a learned latent abstraction of prior-turn audio that replaces each prior-turn audio sequence with a fixed number of latent tokens while retaining corresponding transcripts explicitly. On both in-domain and out-of-domain test sets, the compression-enabled model recovers part of the gains of raw-context conditioning with a smaller prior-turn audio footprint. We also provide targeted analyses of the compression setup and its tradeoffs.

Index Terms—Context-aware ASR, speech recognition, multimodal LLMs, audio compression, long-context modeling.

I. INTRODUCTION

Automatic speech recognition (ASR) is increasingly deployed in conversational settings, such as voice assistants, customer-support calls, meetings, and multimodal agents. In these scenarios, interpreting an utterance often depends on previous turns, which can establish domain-specific terminology, introduce named entities, or provide discourse cues that resolve ambiguity. Despite this dependence, most ASR systems process utterances independently [1]–[4]. This limitation is especially detrimental for rare, error-prone contextual entities (e.g., names, locations, products). Prior work on contextual and conversational ASR has addressed this through text-side contextual biasing via finite-state transducers, shallow fusion, or context-aware end-to-end architectures [5]–[11], as well as session-level language modeling [12]. While effective, these methods focus on text-side biasing signals or single-modality history rather than rich, multimodal conversational context.

Multimodal large language models (LLMs) provide a natural framework for revisiting this problem [13]–[20]. By mapping audio into the embedding space of an autoregressive LLM, these systems can natively process interleaved audio, text, and structured prompts. Most prior work in LLM-based ASR has focused on single-utterance recognition or general instruction-following. Extending this to conversational ASR allows prior turns to provide both lexical evidence (what was said via word choice) and acoustic evidence (how it was

said via audio). However, exploiting this multimodal context introduces a severe computational bottleneck: prior-turn audio is represented by long sequences of tokens, causing prompt lengths, key-value (KV) cache costs, and inference latency to grow rapidly with conversation length.

While general long-context modeling and learned compression, such as cross-attention bottlenecks [21]–[23] and memory summarization [24]–[27], have been explored for text and vision, how to effectively distill multimodal context specifically for ASR remains an open question. This raises two linked questions: Does conversational multimodal context actually improve LLM-based ASR? And can those gains be retained under a fixed, substantially smaller context budget?

In this work, we study this tradeoff. We first demonstrate that simply prepending raw conversational context degrades performance at inference time. However, after supervised multi-turn fine-tuning, the model benefits significantly from context, showing distinct gains on recovering contextual entities (measured via bias-word error rate). Because raw-audio context conditioning is expensive, we introduce **Abstract Compression**, a task-driven bottleneck that abstracts each prior-turn audio sequence into a small, fixed number of learned latent tokens while retaining prior-turn transcripts in explicit textual form (see Figure 1).

Taken together, our work lies at the intersection of contextual ASR, multimodal LLM-based speech recognition, and efficient sequence modeling. Our specific contributions are:

- We show that multimodal context from prior turns improves LLM-based ASR, with the largest gains on contextual entities after supervised multi-turn training.
- We propose Abstract Compression to distill prior-turn audio into a compact token footprint, mitigating the primary source of context cost.
- Through evaluations on both in-domain and out-of-domain datasets, we show that Abstract Compression recovers a meaningful fraction of the raw-context performance gains, and we provide detailed ablations on the factors governing this quality-efficiency tradeoff.

II. MULTIMODAL LLM-BASED ASR

In this work, we adopt PHI-4-MULTIMODAL [17] as our foundational backbone. Although the architecture supports interleaved image, audio, and text, we focus exclusively on

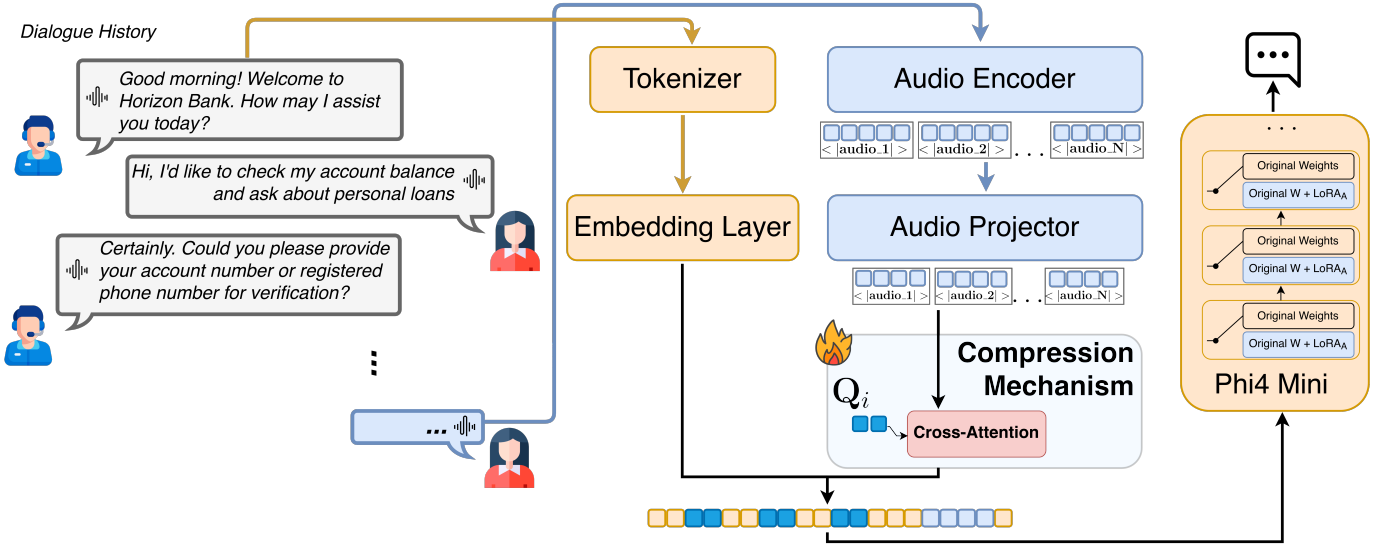


Fig. 1. **Overview of Abstract Compression for context-aware ASR.** Prior conversational turns are represented by both transcript and audio. In our implementation, the audio from each prior turn is distilled into a fixed number of latent tokens, while transcript is retained explicitly. These compressed context representations are provided to the LLM alongside the current turn’s full-resolution audio embeddings. This preserves part of the conversational context while reducing the cost of raw-context conditioning.

its speech-processing capabilities to cleanly isolate the impact of conversational context on speech recognition performance.

Model Architecture and Notation. The model processes an input audio waveform $\mathbf{A}$ through a dedicated audio encoder f_{enc} and a projection module f_{proj} . For a given audio segment, acoustic features are extracted and projected into the LLM’s input embedding space:

$$\mathbf{X} = f_{\text{proj}}(f_{\text{enc}}(\mathbf{A})), \quad (1)$$

where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L\}$ represents a sequence of L audio embeddings of dimension D . These embeddings are interleaved with text embeddings and processed by the PHI-4-MINI LLM [17].

Single-Turn ASR Baseline. In a standard, context-independent (single-turn) ASR setting, the input sequence $\mathcal{S}_{\text{single}}$ is structured using the model’s standard chat template:

$$\mathcal{S}_{\text{single}} = \langle \text{user} \rangle \mathbf{X} \backslash n \mathcal{P} \langle \text{end} \rangle \langle \text{assistant} \rangle, \quad (2)$$

where $\mathcal{P}$ is a natural language instruction to transcribe the audio. For all experiments, we use a fixed prompt $\mathcal{P}$: “Transcribe the audio clip into text.”, following the PHI-4-MULTIMODAL paper [17]. The text generated by the model immediately following the $\langle \text{assistant} \rangle$ token forms our predicted transcription. This single-turn formulation serves as the baseline for all subsequent contextual experiments.

III. CONTEXT-AWARE ASR

To move beyond isolated utterance recognition, we investigate the model’s ability to leverage cues from preceding conversational turns. Formally, a conversation is represented as a sequence of N turns, where each turn i consists of an audio segment $\mathbf{A}_i$ and its corresponding transcript Y_i . Our objective is to transcribe the N -th turn using the preceding $N - 1$ turns

as context, together with the current audio segment $\mathbf{A}_N$. Prior turns are indexed relative to the current turn being transcribed, rather than by their absolute positions in the conversation.

A. Inference-Time Context Conditioning

We first evaluate whether the base model can utilize conversational context through inference-time prompting alone. For this setting, we follow the model’s multi-turn chat prompt format, in which completed prior turns are prepended to the current transcription request. Let $\mathcal{T}_i$ denote a completed prior turn i :

$$\mathcal{T}_i = \underbrace{\langle \text{user} \rangle \mathbf{X}_i \backslash n \mathcal{P} \langle \text{end} \rangle}_{\text{Audio Context}} \underbrace{\langle \text{assistant} \rangle Y_i \langle \text{end} \rangle}_{\text{Transcription Context}}. \quad (3)$$

The full input sequence for the N -th turn is the concatenation:

$$\mathcal{S}_{\text{context}} = \mathcal{T}_1 \mathcal{T}_2 \dots \mathcal{T}_{N-1} \langle \text{user} \rangle \mathbf{X}_N \backslash n \mathcal{P} \langle \text{end} \rangle \langle \text{assistant} \rangle. \quad (4)$$

In our experiments, providing the model with raw conversational context at inference time degraded performance relative to the single-turn ASR baseline. This suggests that inference-time prompting alone is insufficient for reliable cross-turn context use without explicit training.

B. Supervised Fine-Tuning for Contextual Awareness

We therefore perform supervised fine-tuning (SFT) using the multi-turn format in Eq. 4. Concretely, the model is trained to predict the transcript of the final turn, Y_N , conditioned on the preceding conversational turns $\mathcal{T}_1, \dots, \mathcal{T}_{N-1}$ and the current audio input $\mathbf{A}_N$. This allows the model to learn from inputs that include prior turns, rather than from the single-turn formulation alone.

In our experiments, supervised multi-turn fine-tuning with these raw-contexts reduced bias-word error and enabled the model to better recover contextual entities from preceding turns. Detailed results are provided in Section VI-A. However, these gains come at the expense of a substantially larger context size: because each prior-turn audio input $\mathbf{X}_i$ is represented by a high-resolution sequence of audio embeddings, the total length of $\mathcal{S}_{\text{context}}$ grows rapidly with the number of prior turns.

IV. ABSTRACT COMPRESSION FOR CONTEXT-AWARE ASR

In our setting, audio embeddings dominate the prompt length, while transcripts are comparatively compact. Our compression strategy is therefore to replace only the audio portion of each context turn with a fixed-size latent representation and keep the transcripts explicit. This focus is complementary to textual context compression via projection [28]: we target the audio-token bottleneck, while joint use of compressed audio context and projected textual context is left for future work. Our training strategy is also inspired by recent work on context compression for RAG-based LLM systems [27].

A. Compression Mechanism

Instead of processing the full audio-token sequence of each context turn, we learn a compression function $g(\cdot)$ that maps a variable-length audio sequence to a fixed number of latent tokens.

For each context turn i , we compress the high-resolution audio embeddings $\mathbf{X}_i$ into K latent tokens. In this work, we implement $g(\cdot)$ as a learnable cross-attention mechanism with turn-index-specific query matrices [21]. These query matrices are defined for relative context positions with respect to the current turn, up to the maximum supported context length. Concretely, for each turn index i , we define a learnable query matrix $\mathbf{Q}_i \in \mathbb{R}^{K \times D}$. The compressed representation $\mathbf{Z}_i$ is then obtained by cross-attending $\mathbf{Q}_i$ to the original audio embeddings of that turn:

$$\mathbf{Z}_i = \text{CrossAttention}(\mathbf{Q}_i, \mathbf{X}_i, \mathbf{X}_i). \quad (5)$$

Here, $\mathbf{Q}_i$ are learnable parameters, but because they attend to turn-specific keys and values ($\mathbf{X}_i$), the resulting $\mathbf{Z}_i$ vectors dynamically capture the unique context of that specific turn. This architecture introduces an information bottleneck that compresses the audio of each prior turn into a fixed-length set of K latent tokens.

Compressed Multi-turn Prompting. With the compression module in place, each context turn $\mathcal{T}_i$ in Eq. 3 is replaced by its compressed counterpart $\mathcal{T}'_i$:

$$\mathcal{T}'_i = \langle \text{user} \rangle \mathbf{Z}_i \backslash n \mathcal{P} \langle \text{end} \rangle \langle \text{assistant} \rangle Y_i \langle \text{end} \rangle. \quad (6)$$

That is, the original audio-token sequence $\mathbf{X}_i$ is replaced by the compressed representation $\mathbf{Z}_i$, while the transcript Y_i is kept explicit. The resulting compressed multi-turn input sequence is

$$\mathcal{S}_{\text{comp}} = \mathcal{T}'_1 \mathcal{T}'_2 \dots \mathcal{T}'_{N-1} \langle \text{user} \rangle \mathbf{X}_N \backslash n \mathcal{P} \langle \text{end} \rangle \langle \text{assistant} \rangle. \quad (7)$$

Compression is applied only to prior context turns; the current-turn audio $\mathbf{X}_N$ remains uncompressed to preserve full acoustic detail.

B. Training Strategy

Training the model directly on compressed multi-turn inputs is challenging because it must both interpret the latent audio tokens $\mathbf{Z}_i$ and use them for current-turn transcription. We therefore adopt a two-stage training strategy following [27]. In their setting, Stage 1 trains a compression module to reconstruct text from compressed text representations, and Stage 2 performs curriculum-based training with progressively increasing context length. We adapt this procedure to ASR: Stage 1 aligns compressed audio representations to the LLM through single-turn ASR, and Stage 2 fine-tunes the model on compressed multi-turn inputs using a curriculum over the number of context turns.

Stage 1: Compressed-Audio-to-LLM Alignment. The goal of the first stage is to make the compressed audio representations compatible with the LLM input space. To do so, we repurpose the single-turn ASR task by replacing the raw audio embeddings $\mathbf{X}_1$ in the baseline prompt (Eq. 2) with the compressed tokens $\mathbf{Z}_1$. During this stage, the base model is frozen, and only the compression module, including the turn-specific queries $\mathbf{Q}_i$ and the cross-attention parameters, is optimized. The model is trained with the standard cross-entropy loss to predict the transcript Y_1 from the compressed audio input. We do not compress prior-turn text in this work. Unlike audio, compressed text did not admit a comparably effective alignment stage in our preliminary experiments, and retaining transcripts explicitly yielded a simpler and more reliable training setup.

Stage 2: Contextual Fine-tuning. In the second stage, we initialize the compression module from the Stage 1 checkpoint and fine-tune the model using the compressed multi-turn input $\mathcal{S}_{\text{comp}}$ in Eq. 7. At this stage, we jointly optimize the compression module and the audio LoRA [29] parameters of the LLM, allowing the model to combine compressed context turns with the full-resolution audio of the current turn. Following [27], we use a curriculum over context length: training starts with no context turns and progressively increases the maximum number of available turns until the target number of context turns is reached.

V. EXPERIMENTAL SETUP

A. Datasets

Our experiments use DefinedAI¹ as the main in-domain dataset, WoW as the out-of-domain evaluation set, and LibriSpeech 960h [30] only for Stage 1 compression training.

DefinedAI consists of scripted conversations between agents and customers. Unless specified, all models are fine-tuned on the DefinedAI train split and selected on the dev split. The dataset contains 17k/559/2k utterances in train/dev/test, corresponding to 40/2.25/4.5 hours and 359k/20k/42k total

¹<https://www.defined.ai>

words. The numbers of annotated contextual entities (measured as entity words) are 6.5k/379/727, with approximately 12.9k/700/1.4k entity words in train/dev/test.

WoW is an internal dataset of real call-center conversations between agents and customers, used only for out-of-domain evaluation. None of our models are trained on WoW. The test set contains 1,465 utterances (1.76 hours) and 20k total words, of which 3,434 are entity words, corresponding to an entity-word ratio of 16.9%. We include WoW because it is more realistic than the scripted DefinedAI data and substantially richer in contextual entities.

LibriSpeech 960h is a large read-speech corpus that we use for additional pretraining of the audio compression module in single-turn compressed-audio experiments (Stage 1 of training in Section IV-B). LibriSpeech is not used for Stage 2 contextual fine-tuning or for evaluation.

Evaluation protocol. We evaluate on the DefinedAI (in-domain) and WoW (out-of-domain) test splits; unless noted, all contextual models are trained on DefinedAI.

B. Evaluation Metrics

We report standard word error rate (WER) over the full reference transcript and bias-word error rate (Bias-WER) over annotated contextual-entity tokens, such as person names, locations, and product names. Bias-WER uses the same alignment and edit-distance procedure as standard WER, but restricts scoring to the annotated entity spans.

C. Implementation Details

We use PHI-4-MULTIMODAL as the backbone in all experiments. Single-turn and raw-context ASR models are fine-tuned on DefinedAI for 4 epochs using AdamW, batch size 16, learning rate 4×10^{-5} , weight decay 0.01, maximum gradient norm 1.0, and linear scheduler with 50 warmup steps. For raw-context ASR, the number of prior turns is sampled uniformly from 0 to 10 during training, teaching the model to use a variable number of context turns during fine-tuning.

For Abstract Compression, Stage 1 starts from the best DefinedAI single-turn checkpoint with the cross-attention compression module attached. The backbone is frozen, only the compression module is optimized, and the learning rate is increased to 10^{-3} . Stage 2 initializes from the best Stage 1 checkpoint and jointly fine-tunes the compression module and audio LoRA parameters, using the same optimization settings as standard ASR fine-tuning. Unlike raw-context training, Stage 2 uses a curriculum over context length: training starts with zero prior turns and increases the maximum available context by one every 10% of training until reaching 10 turns.

During decoding, we use a fixed number of prior turns for each evaluation setting. This provides a controlled comparison across models and makes the effect of context length easier to interpret. Prior-turn transcripts are provided as ground-truth text during both training and inference, and the model predicts only the current-turn transcript. Unless otherwise noted, checkpoints are selected on the DefinedAI dev split.

TABLE I
ASR PERFORMANCE ON THE DEFINEDAI (IN-DOMAIN) AND WoW (OUT-OF-DOMAIN) TEST SETS. ALL SYSTEMS USE PHI-4-MULTIMODAL (PHI-4-MM). OPEN-SOURCE PHI-4-MM DENOTES THE ORIGINAL NON-FINE-TUNED MODEL, WHILE THE REMAINING CONFIGURATIONS ARE FINE-TUNED VARIANTS. N_{TRAIN} AND N_{DECODE} DENOTE THE NUMBER OF CONTEXT TURNS USED FOR TRAINING AND DECODING.

Configuration	N_{train}	N_{decode}	WER	Bias-WER
DefinedAI Test Set (In-domain)				
Open-source PHI-4-MM	0	0	13.4	21.0
Single-turn	0	0	7.6	13.5
		5	9.1	15.1
Multi-turn, Raw	$\mathcal{R}_{[0,10]}$	5	7.5	13.3
		10	7.5	13.1
Multi-turn, Compressed	$0 \rightarrow 10$	10	8.0	13.3
WoW Test Set (Out-of-domain)				
Open-source PHI-4-MM	0	0	20.4	30.0
Single-turn	0	0	13.4	25.6
		10	14.5	27.2
Multi-turn, Raw	$\mathcal{R}_{[1,10]}$	5	12.9	24.2
		10	12.7	23.3
Multi-turn, Compressed	$0 \rightarrow 10$	10	13.2	24.5

VI. EXPERIMENTAL RESULTS

We organize the results around three questions. First, can the model benefit from conversational context at all? Second, can Abstract Compression retain those gains under a much smaller prior-turn audio budget? Third, what factors govern the quality-efficiency tradeoff?

A. Raw Context

Table I compares single-turn ASR, inference-time context prompting, and raw multi-turn fine-tuning. Fine-tuning PHI-4-MULTIMODAL for single-turn ASR substantially improves over the open-source model, establishing the single-turn row as the main baseline. Simply decoding this model with context, however, degrades performance, showing that inference-time prompting alone is insufficient for reliable cross-turn use. After supervised multi-turn fine-tuning, raw context improves over the single-turn baseline on both datasets, with larger gains on Bias-WER than overall WER. On DefinedAI, 10 context turns reduce Bias-WER from 13.5% to 13.1%; on the more entity-dense WoW set, WER improves from 13.4% to 12.7% and Bias-WER from 25.6% to 23.3%. This supports our hypothesis that conversational context is most useful for recovering contextual entities.

A context-modality ablation on WoW (Table II) shows that raw audio-only context performs similarly to audio+text context and significantly better than text-only context, suggesting that full-resolution audio already contains most contextual cues. This supports compressing prior-turn audio, rather than relying on standalone transcript conditioning, to preserve contextual information. We nevertheless retain explicit transcripts in Abstract Compression because text adds little prompt

TABLE II
CONTEXT-MODALITY ABLATION ON WoW. N_{DECODE} DENOTES THE NUMBER OF CONTEXT TURNS USED AT DECODING.

Context Modality	N_{decode}	WER	Bias-WER
Audio & Text	5	12.9	24.2
	10	12.7	23.3
Audio only	5	13.0	23.8
	10	12.8	23.2
Text only	5	13.5	28.9
	10	13.4	28.5

TABLE III
STAGE 1 ASR FROM COMPRESSED AUDIO (WITHOUT CONTEXT). K DENOTES THE NUMBER OF LATENT AUDIO TOKENS.

Compression Setting	Training Data	DefinedAI WER	WoW WER
Full audio	DefinedAI (40h)	7.6	13.4
$K = 4$	DefinedAI (40h)	51.2	72.3
$K = 8$	DefinedAI (40h)	46.3	66.4
$K = 16$	DefinedAI (40h)	37.1	54.2
$K = 16$	LibriSpeech (960h) + DefinedAI (40h)	21.2	29.1

overhead and preserves the standard chat-template structure of PHI-4-MULTIMODAL, ensuring stable instruction-following.

B. Abstract Compression

Stage 1. Table III evaluates whether compressed audio embeddings retain enough information for single-turn ASR. Increasing K improves WER, and adding LibriSpeech 960h to DefinedAI further improves the $K = 16$ setting. Although compressed audio remains worse than full-resolution audio, these results show that the latent tokens preserve useful acoustic information for downstream contextual modeling.

Stage 2. Table I shows that compressed multi-turn conditioning improves Bias-WER over the single-turn baseline while remaining behind raw-context conditioning. On DefinedAI, Abstract Compression reaches 13.3% Bias-WER, compared with 13.5% for single-turn and 13.1% for raw context. On WoW, it reaches 24.5% Bias-WER, improving over the 25.6% single-turn baseline while trailing the 23.3% raw-context result. Thus, compressed context recovers a meaningful fraction of the entity-recognition benefit of raw context under a smaller prior-turn audio budget.

C. Compression Rate Analysis

To evaluate efficiency of Abstract Compression, we measure how much of the context input is retained after compression. For a sample with n prior turns, we first define the audio-only compression rate as

$$\rho_n^{\text{audio}} = \frac{nK}{\sum_{i=1}^n |\mathbf{X}_i|}, \quad (8)$$

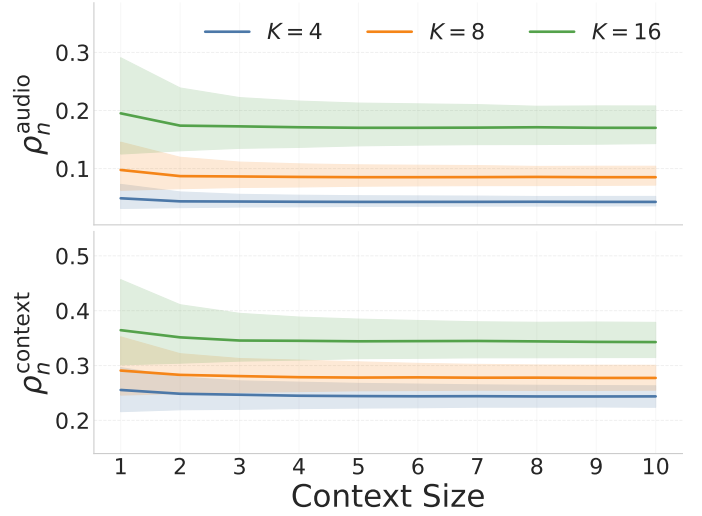


Fig. 2. Compression rates as a function of context size on DefinedAI test set. Top: audio compression rate ρ_n^{audio} (see Eq. 8). Bottom: overall context compression rate ρ_n^{context} (see Eq. 9). Lines show the median across examples and shaded regions denote interquartile range.

where $|\mathbf{X}_i|$ is the number of raw audio embeddings in context turn i , and K is the fixed number of latent audio embeddings per turn. This captures how aggressively the prior-turn audio stream is compressed. Second, we define the overall context compression rate as

$$\rho_n^{\text{context}} = \frac{\sum_{i=1}^n (K + |Y_i|)}{\sum_{i=1}^n (|\mathbf{X}_i| + |Y_i|)}, \quad (9)$$

where $|Y_i|$ is the number of transcript tokens in turn i . This metric reflects the retained fraction of the full context prompt, since our method compresses context audio only, while keeping prior-turn transcripts explicit. Smaller values of either ratio indicate stronger compression. The distinction between the two metrics is important. ρ_n^{audio} measures how aggressively the audio is compressed, whereas ρ_n^{context} captures the realized reduction in the full conversational context seen by the model at inference time. Because prior-turn transcripts are not compressed, ρ_n^{context} is naturally larger than ρ_n^{audio} .

Figure 2 shows two trends on the DefinedAI test set. First, both ρ_n^{audio} and ρ_n^{context} decrease and then stabilize as more context turns are included, since each additional turn contributes exactly K latent audio tokens regardless of its original duration. Second, the variance of both ratios narrows with more turns, as utterance-level length variation averages out across the context. Thus, Abstract Compression not only reduces the average context token budget, but also makes the effective context footprint more predictable across examples.

D. Ablation Studies

We next analyze which aspects of the compression setup affect performance the most. All ablations are reported on the WoW test set in Table IV.

Effect of bottleneck size. The number of latent tokens K directly controls the information capacity of the bottleneck.

TABLE IV
STAGE 2 COMPRESSION ABLATIONS ON WOW.

Latent Tokens (K)	Stage 1 Train Data	N_{decode}	WER	Bias-WER
4	DefinedAI (40h)	0	13.5	26.6
		1	13.2	26.0
		5	13.3	25.2
		10	13.4	25.3
8	DefinedAI (40h)	0	13.7	27.0
		1	13.5	26.6
		5	13.4	25.5
		10	13.5	25.6
16	DefinedAI (40h)	0	13.5	26.5
		1	13.3	25.3
		5	13.3	24.5
		10	13.2	24.5
16	LibriSpeech (960h) + DefinedAI (40h)	0	13.4	26.3
		1	13.3	25.1
		5	13.6	25.1
		10	13.4	25.1

The trend is consistent across both Stage 1 and Stage 2. In Stage 1 (Table III), increasing K improves single-turn transcription from compressed audio alone. In Stage 2 (Table IV), larger K also makes compressed context more useful. With $K = 4$, Bias-WER improves only modestly, from 26.6% without context to 25.2% with context. With $K = 8$, the best Bias-WER is 25.5%. The strongest results are obtained with $K = 16$, where Bias-WER drops from 26.5% without context to 24.5% with 5 or 10 context turns. This suggests that very aggressive compression removes information needed for cross-turn disambiguation, whereas a moderate bottleneck can preserve a useful portion of the contextual signal.

Effect of Stage 1 data. The Stage 1 results in Table III show that a larger Stage 1 training set clearly helps in the single-turn alignment setting: for $K = 16$, pretraining on LibriSpeech 960h + DefinedAI substantially improves Stage 1 WER relative to DefinedAI-only training. However, Table IV shows that this gain does not translate into better contextual ASR after Stage 2 fine-tuning. For $K = 16$, initializing Stage 2 with LibriSpeech 960h + DefinedAI does not outperform the DefinedAI-only initialization. This suggests that better single-turn compressed-audio reconstruction is not sufficient by itself to guarantee better downstream contextual use. One possible explanation is that Stage 2 depends not only on whether the latent tokens preserve acoustic content, but also on whether that content is aligned with the conversational cues that matter for entity recovery in the downstream domain. In our setting, in-domain Stage 1 training may therefore provide a better initialization for this task-specific alignment, even if it is weaker as a general-purpose audio compressor.

Effect of context length. Table IV shows how the compressed model behaves as context turns increase at inference. In nearly all settings, adding context improves over the no-context condition, confirming that the model uses the compressed representations. Most gains arise from the first few

turns and are largely saturated with 5 turns. For instance, with $K = 16$ and DefinedAI Stage 1 training, Bias-WER drops from 26.5% with 0 context turns (single-turn) to 25.3% with 1 turn and 24.5% with 5 turns, with no additional improvement with 10 turns. This pattern is favorable from an efficiency standpoint, since it suggests that a moderate amount of compressed context captures most of the contextual benefit. **System-Level Efficiency Implications.** We benchmark the LLM forward pass after input construction to isolate the cost induced by the input sequence on a single NVIDIA H100 80 GB GPU. Table V compares single-turn decoding with raw and compressed 10-turn decoding, showing raw-context substantially increases both forward time and peak GPU memory relative to the single-turn baseline, while Abstract Compression reduces this overhead significantly.

TABLE V
ABSTRACT COMPRESSION EFFICIENCY ON THE TEST SET. AVG. TIME IS WALL-CLOCK LLM-FORWARD TIME PER DECODED UTTERANCE, REPORTED IN SECONDS; PEAK MEMORY IS REPORTED IN GB, AND K DENOTES THE NUMBER OF LATENT AUDIO TOKENS PER CONTEXT TURN.

Configuration	K	N_{decode}	Avg. Time	Peak Mem.
Single-turn	–	0	0.440	11.07
Multi-turn, Raw	–	10	1.525	28.98
Multi-turn, Compressed	4	10	0.644	17.13
Multi-turn, Compressed	8	10	0.732	17.65
Multi-turn, Compressed	16	10	0.835	18.05

VII. ILLUSTRATIVE EXAMPLE

Figure 3 demonstrates how acoustic context aids recognition: when the entity “Buick Encore” appears in the history and is spoken rapidly in the target turn, the single-turn model predicts the phonetically similar phrase “VIEWER CONCOURSE,” whereas both raw and compressed context recover the correct entity. This suggests that compressed context can preserve acoustic cues useful for recovering the entity.

Compressed context recovers the entity	
History	HOW MUCH FOR THE TWENTY BUICK ENCORE
Target	THIRTY ONE FOR THE TWENTY BUICK ENCORE
Single-turn	THIRTY ONE FOR THE TWENTY VIEWER CONCOURSE
Raw context	THIRTY ONE FOR THE TWENTY BUICK ENCORE
Compressed	THIRTY ONE FOR THE TWENTY BUICK ENCORE

Fig. 3. Qualitative entity-recovery example.

VIII. CONCLUSION

We show that conversational multimodal context can improve LLM-based ASR, but only after supervised multi-turn training, with the largest gains accruing on contextual entities. To reduce the cost of raw-context conditioning, we introduce Abstract Compression, which compresses prior-turn audio into a fixed latent budget and recovers part of the benefit of full raw context. Overall, our results suggest that compact representations of conversational audio context offer a practical quality-efficiency tradeoff for context-aware ASR.

ACKNOWLEDGEMENTS

This work was supported by Idiap Research Institute and Uniphore collaboration project. Part of this work was also supported by EU Horizon 2020 project ELOQUENCE (grant number 101070558).

The authors used ChatGPT (OpenAI) and Claude (Anthropic) for language editing and improving the clarity of the manuscript. All experiments, results, and scientific claims are the authors' own.

REFERENCES

- [1] S. Kim and F. Metze, "Dialog-context aware end-to-end speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 434–440.
- [2] S. Kim, S. Dalmia, and F. Metze, "Gated embeddings in end-to-end speech recognition for conversational-context fusion," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1131–1141.
- [3] T. Hori, N. Moritz, C. Hori, and J. L. Roux, "Advanced long-context end-to-end speech recognition using context-expanded transformers," in *Proc. Interspeech 2021*, 2021, pp. 2097–2101.
- [4] W. Lee, S. Kim, and G. G. Lee, "Enhancing dialogue speech recognition with robust contextual awareness via noise representation learning," in *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2024, pp. 333–343.
- [5] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: end-to-end contextual speech recognition," in *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018, pp. 418–425.
- [6] I. Williams, A. Kannan, P. S. Aleksic, D. Rybach, and T. N. Sainath, "Contextual speech recognition in end-to-end neural network systems using beam search," in *Interspeech*, 2018, pp. 2227–2231.
- [7] M. Jain, G. Keren, J. Mahadeokar, G. Zweig, F. Metze, and Y. Saraf, "Contextual RNN-T for open domain ASR," in *Proc. Interspeech 2020*, 2020, pp. 11–15.
- [8] S. Tong, P. Harding, and S. Wiesler, "Hierarchical attention-based contextual biasing for personalized speech recognition using neural transducers," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.
- [9] S. Zhou, Z. Li, Y. Hong, M. Zhang, Z. Wang, and B. Huai, "CopyNE: Better contextual ASR by copying named entities," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 2675–2686.
- [10] Y. Li, Y. Li, M. Zhang, C. Su, J. Yu, M. Piao, X. Qiao, M. Ma, Y. Zhao, and H. Yang, "CB-Whisper: Contextual biasing Whisper using open-vocabulary keyword-spotting," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 2941–2946.
- [11] H. Liu, Y. Wang, and Y. Wang, "Post-decoder biasing for end-to-end speech recognition of multi-turn medical interview," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 12917–12926.
- [12] W. Xiong, L. Wu, J. Zhang, and A. Stolcke, "Session-level language modeling for conversational speech," in *Proc. EMNLP*, 2018, pp. 2764–2768.
- [13] C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, M. Zejun, and C. Zhang, "SALMONN: Towards Generic Hearing Abilities for Large Language Models," in *Proc. 12th Intl. Conf. on Learning Representations*, 2024.
- [14] J. Wu, Y. Gaur, Z. Chen, L. Zhou, Y. Zhu, T. Wang, J. Li, S. Liu, B. Ren, L. Liu *et al.*, "On decoder-only architecture for speech-to-text and large language model integration," in *Proc. IEEE ASRU*, 2023.
- [15] Z. Ma, G. Yang, Y. Yang, Z. Gao, J. Wang, Z. Du, F. Yu, Q. Chen, S. Zheng, S. Zhang *et al.*, "An embarrassingly simple approach for LLM with strong ASR capacity," preprint arXiv:2402.08846, 2024.
- [16] S. Kumar, I. Thorbecke, S. Burdisso, E. Villatoro-Tello, M. K E, K. Hacıoğlu, P. Rangappa, P. Motlicek, A. Ganapathiraju, and A. Stolcke, "Performance evaluation of SLAM-ASR: The good, the bad, the ugly, and the way forward," in *Proc. IEEE ICASSP Workshop*, 2025.
- [17] A. Abouelenin, A. Ashfaq, A. Atkinson, H. Awadalla, N. Bach, J. Bao, A. Benhaim, M. Cai, V. Chaudhary, C. Chen *et al.*, "Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras," *arXiv preprint arXiv:2503.01743*, 2025.
- [18] S. Burdisso, E. Villatoro-Tello, T. Bañeras-Roux, S. Kumar, S. Madikeri, M. K E, P. Motlicek, and A. Stolcke, "Avoiding Catastrophic Forgetting in Text-Only Adaptation of LLM-based ASR via Multi-View Text Denoising," in *Proc. Interspeech*, 2026.
- [19] E. Lakomkin, C. Wu, Y. Fathullah, O. Kalinli, M. L. Seltzer, and C. Fuegen, "End-to-end speech recognition contextualization with large language models," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 12 406–12 410.
- [20] E. Villatoro-Tello, S. Burdisso, S. Kumar, H. Watawana, S. Madikeri, M. K E, J. Prakash, T. Bañeras-Roux, K. Hacıoğlu, P. Motlicek, and A. Stolcke, "Context Projector: Complementary Keyword and Dialogue Context Embeddings for LLM-based ASR," in *Proc. Interspeech*, 2026.
- [21] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General perception with iterative attention," in *International conference on machine learning*. PMLR, 2021, pp. 4651–4664.
- [22] J. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. L. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, "Flamingo: a visual language model for few-shot learning," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022.
- [23] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 2023, pp. 19 730–19 742. [Online]. Available: <https://proceedings.mlr.press/v202/li23q.html>
- [24] J. W. Rae, A. Potapenko, S. M. Jayakumar, C. Hillier, and T. P. Lillicrap, "Compressive transformers for long-range sequence modelling," in *International Conference on Learning Representations*, 2020. [Online]. Available: <https://openreview.net/forum?id=SylKikSYDH>
- [25] F. Xu, W. Shi, and E. Choi, "Recomp: Improving retrieval-augmented lms with context compression and selective augmentation," in *The Twelfth International Conference on Learning Representations*, 2023.
- [26] X. Li, H. Li, Y. Zhou, Q. Ai, and Y. Liu, "Atacompressor: Adaptive task-aware compression for efficient long-context processing in LLMs," in *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, 2025, pp. 343–352.
- [27] X. Lin, A. Ghosh, B. K. H. Low, A. Shrivastava, and V. Mohan, "REFRAG: Rethinking RAG based decoding," 2025. [Online]. Available: <https://arxiv.org/abs/2509.01092>
- [28] E. Villatoro-Tello, S. Burdisso *et al.*, "Context projector: Complementary keyword and dialogue context embeddings for LLM-based ASR," in *Proceedings Interspeech*, 2027, to appear.
- [29] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "LoRA: Low-rank adaptation of large language models." *Iclr*, vol. 1, no. 2, p. 3, 2022.
- [30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: an ASR corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.