

---

# Streaming Sequence-to-Sequence Learning with Delayed Streams Modeling

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 We introduce Delayed Streams Modeling (DSM), a flexible formulation for  
2 streaming, multimodal sequence-to-sequence learning. Sequence-to-sequence  
3 generation is typically cast in an offline manner: the model consumes the com-  
4 plete input sequence before generating the first output timestep. DSM instead  
5 models time-aligned streams with a decoder-only language model. By further-  
6 more introducing delays between streams, and selectively feeding or sampling  
7 them, DSM provides streaming inference of arbitrary output sequences, from  
8 any input combination, making it applicable to many sequence-to-sequence prob-  
9 lems. In particular, given a text and audio stream, automatic speech recogni-  
10 tion (ASR) corresponds to the text stream being delayed, while the opposite  
11 gives a text-to-speech (TTS) model. We perform extensive experiments for these  
12 two major sequence-to-sequence tasks, showing that DSM provides state-of-the-  
13 art performance and latency while supporting arbitrary long sequences, being  
14 even competitive with offline baselines. We demonstrate DSM applications on  
15 <https://delayed-stream-modeling.github.io/>.

## 16 1 Introduction

17 We are interested in streaming sequence-to-sequence (seq2seq) learning, i.e. predicting an output  
18 sequence as we process an input sequence synchronously, as opposed to offline seq2seq where  
19 inputs are recorded entirely before producing the output sequence. The latter class of offline  
20 models was introduced for a diverse set of tasks such as handwriting recognition (Graves et al.,  
21 2013), automatic speech recognition (ASR) (Graves et al., 2013) or machine translation (Bahdanau  
22 et al., 2015; Sutskever et al., 2014), by designing modality-dependent input encoders, typically  
23 coupled with a text decoder (Hochreiter & Schmidhuber, 1997). Although this asymmetry between  
24 input processing and output generation facilitated the adoption of this framework in many tasks,  
25 it also led to a divergence of model architectures across modalities. As an example, a Tacotron  
26 text-to-speech (TTS) model (Wang et al., 2017) would differ from an ASR model such as LAS (Chan  
27 et al., 2016). The advent of decoder-only Transformers (Vaswani et al., 2017) for text language  
28 modeling reduced the gap between input and output processing by allowing a single model to process  
29 a simple concatenation of tokens. In parallel, neural compression algorithms that can transform  
30 images (Razavi et al., 2019; Esser et al., 2020) and audio (Zeghidour et al., 2022; Défossez et al.,  
31 2023) into discrete tokens analogous to text allowed integrating these modalities along text sequences.  
32 Thus, a decoder-only model can be used for seq2seq tasks such as ASR (Rubenstein et al., 2023),  
33 TTS (Wang et al., 2023), spoken dialogue (Défossez et al., 2024), visual understanding (Beyer  
34 et al., 2024) or image generation (Ramesh et al., 2021). Furthermore, inputs and outputs are  
35 interchangeable in this framework, meaning a single model can be trained for generation in both  
36 directions: AudioPALM (Rubenstein et al., 2023) performs TTS and ASR, while CM3Leon (Yu et al.,  
37 2023) provides both image captioning and generation. Yet, a major limitation of these decoder-only

approaches is their incompatibility with streaming. First, their prefix-based formulation requires access to the full input sequence before generation, which prevents real-time inference and inherently limits the maximum input length. Second, modalities operate at differing framerates: audio or video tokens are typically sampled regularly, while text tokens represent linguistic units pronounced over varying durations. This prevents applications such as meeting transcription or continuous translation.

In this work, we present Delayed Streams Modeling (DSM), a framework for streaming sequence-to-sequence learning across modalities. DSM uses a decoder-only model to process as many parallel token streams as there are I/O sequences. This multistream architecture, introduced by Défossez et al. (2024), allows for a synchronous autoregressive modeling of aligned sequences which—when coupled with a finite context—provides real-time, streaming generation over infinite input sequences. Moreover, by operating at a constant framerate, DSM allows for batching, a feature rarely provided by streaming models. The second key component of DSM is the introduction of a delay between streams to control the quality/latency trade-off: shifting a sequence B such that it is delayed w.r.t. sequence A allows for a better prediction of the former based on the latter. With an appropriate masking, a DSM model can be trained to continuously predict any combination of output sequences from any combination of input sequences. To illustrate the abilities of the DSM framework, we train speech-text models for ASR and TTS. We show how DSM provides a state-of-the-art tradeoff between latency—as low as a few hundred milliseconds—and quality, while providing long-form synthesis and transcription, along with precise word timestamps that locate where they are pronounced. We furthermore introduce *delay conditioning* which allows controlling the quality/latency trade-off at inference time, without retraining a model. We will release our code and models, along with an evaluation dataset for long-form dialog TTS.

## 2 Related Work

**Streaming Sequence-to-Sequence Learning.** Most streaming seq2seq literature has focused on speech-to-text tasks, in particular ASR (Li et al., 2021) and translation (Zhang et al., 2024; Barrault et al., 2023). Monotonic (Raffel et al., 2017; Chiu & Raffel, 2018) and local (Chiu et al., 2019) attention respectively allow for causal attention of outputs with respect to inputs along and handling arbitrarily long sequences. A common limitation of streaming models is their incompatibility with batching when using an inference policy (Barrault et al., 2023), or the lack of symmetry meaning that specific models must be used for speech-to-text (Li et al., 2021) and text-to-speech (Wang et al., 2017). In contrast, DSM allows for batching and accelerated inference. In the context of this paper, this allows DSM to be trained for state-of-the-art ASR or TTS (see Figure 1), as shown in Section 4, with its performance being even competitive with offline approaches.

**Multimodal language models.** Transformer-based autoregressive models are the current main approach to sequence-to-sequence problems. They were introduced by Vaswani et al. (2017) for machine translation, and were soon extended to multimodal tasks, such as ASR (Radford et al., 2023) or visual understanding (Alayrac et al., 2022), by designing modality-specific encoders. More recently, neural codecs have provided compact, discrete representations of images (Esser et al., 2020) and audio (Zeghidour et al., 2022) that remove the need for modality-specific encoders inside the generative model, while providing a symmetrical processing of inputs and outputs which allows performing bidirectional tasks (e.g. speech-to-text and text-to-speech (Rubenstein et al., 2023)) with a single architecture. Défossez et al. (2024) introduce a multistream decoder architecture for spoken dialogue, which predicts text and audio tokens in a streaming fashion, later applied by Labiausse et al. (2025) to real-time speech translation. In this work we extend the approach of Défossez et al. (2024), in order to reach state-of-the-art performance on the two most competitive speech-text tasks, namely ASR and TTS. Moreover, while (Défossez et al., 2024) and (Labiausse et al., 2025) operate with a delay specified before training, we propose delay conditioning for inference-time latency control without retraining.

## 3 Method

**Notation.** We wish to solve a sequence-to-sequence task between two domains  $\mathcal{X}$  and  $\mathcal{Y}$ . Each domain consists of sequences of vectors of all possible lengths, e.g.

$$\mathcal{X} = \bigcup_{T \in \mathbb{N}} \{(X_t) \in \mathbb{R}^{T \times d}\}, \quad \mathcal{Y} = \bigcup_{T' \in \mathbb{N}} \{(Y_{t'}) \in \mathbb{R}^{T' \times d'}\}. \quad (1)$$

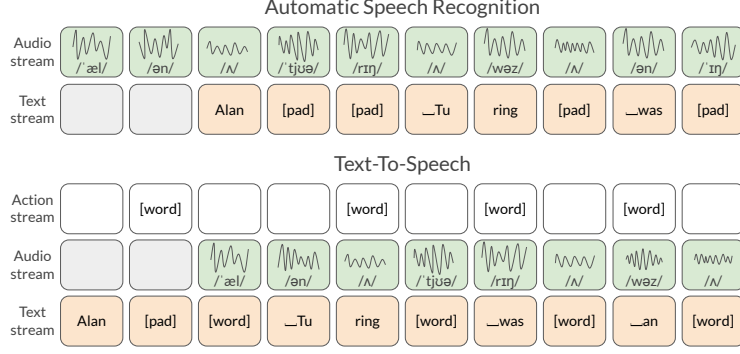


Figure 1: **Delayed stream modeling for speech-text tasks.** Depending on which stream is delayed with respect to the other, we solve either an ASR or a TTS task. For TTS, we further need an action stream for the model to let us know when it is ready to receive a new word.

88 In the case where either  $X_t$  or  $Y_t$  is discrete-valued, we can use a one-hot representation for it in  
 89 Eq. (1). We assume that we are given a joint probability distribution over the outer product domain  
 90  $\mathcal{X} \times \mathcal{Y}$ , and that we have the random variables  $X \in \mathcal{X}$  and  $Y \in \mathcal{Y}$ , along with the joint distribution

$$\mathbb{P}[X, Y] = p(X, Y). \quad (2)$$

91 We also introduce  $T \in \mathbb{N}$  (resp.  $T'$ ) the random variable indicating the length of  $X$  (resp.  $Y$ ),  
 92 along with the marginals  $p(X)$  and  $p(Y)$ . For any sequence  $Z$ , and index  $t$ , we denote  $Z_{<t} =$   
 93  $(Z_1, \dots, Z_{t-1})$ , potentially empty if  $t \leq 0$ . We similarly define  $Z_{\leq t}$ ,  $Z_{\geq t}$ , and  $Z_{>t}$ .

94 **Sequence-to-sequence as joint modeling.** Let's assume for this paragraph that  $\mathcal{X}$  is the set of all  
 95 possible monophonic waveforms sampled at 24 kHz, and  $\mathcal{Y}$  is made of sequences of one-hot encoded  
 96 vectors over a set of words. Intuitively, we assume there exists a coupling  $p(X, Y)$  such that  $p(X, Y)$   
 97 is high if  $Y$  represents the transcription of  $X$ , or conversely, if  $X$  represents a speech utterance of  
 98 the text given by  $Y$ . Formally, the task of ASR corresponds to sampling from the distribution  $\mathbb{P}[Y|X]$ ,  
 99 while the task of TTS corresponds to sampling from the distribution  $\mathbb{P}[X|Y]$ . Thus, each task can be  
 100 solved by accurately estimating both probability distributions,

$$q(X, Y) \approx \mathbb{P}[Y|X], \quad q'(Y, X) \approx \mathbb{P}[X|Y]. \quad (3)$$

101 For simplicity, we now only focus on estimating  $\mathbb{P}[Y|X]$ , the inverse task being obtained by exchanging  
 102 the definition of  $X$  and  $Y$ . We thus call  $\mathcal{X}$  the input domain, and  $\mathcal{Y}$  the output domain.

103 **Auto-regressive modeling of  $Y$ .** A good candidate for estimating  $\mathbb{P}[Y|X]$  is auto-regressive model-  
 104 ing, with a Transformer model (Vaswani et al., 2017), under the extra assumptions that the output  
 105 domain  $\mathcal{Y}$  can be discretized. Thus, one would estimate

$$q(y|X, Y_{<t}) \approx \mathbb{P}[Y_t = y|X, Y_{<t}]. \quad (4)$$

106 One can then sample  $Y$  auto-regressively, knowing  $X$ . Due to the lack of explicit structure between  
 107 the time grid  $t$  of  $X$  and  $t'$  of  $Y$ , one would usually condition on the entirety of  $X$ , e.g. when using  
 108 Transformer based models, either by prefixing the entire sequence  $X$  before the generation  $Y$ , or by  
 109 providing  $X$  through cross-attention layers, which is mathematically equivalent. This forbids the use  
 110 of the model in a streaming fashion, as the entire input signal  $X$  must be known ahead of time, and  
 111 cannot be extended once the generation of  $Y$  has started. Such methods often require explicit and  
 112 manual chunking and stitching operations, which also reduces their ability to be efficiently batched.  
 113 Conversely, aligning  $X$  and  $Y$  to the same frame rate allows for batched streaming inference.

114 **Aligning sequences for streaming prediction.** We assume that both domains  $\mathcal{X}$  and  $\mathcal{Y}$  can share the  
 115 same time grid, e.g.  $(X_t) \in \mathbb{R}^{T \times d}$  and  $(Y_t) \in \mathbb{R}^{T \times d'}$ . We call two such aligned sequences *streams*.  
 116 Then one can simply model

$$q_{\text{aligned}}(y|X_{\leq t}, Y_{<t}) \approx \mathbb{P}[Y_t = y|X_{\leq t}, Y_{<t}]. \quad (5)$$

117 Given  $X \sim p(X)$ , we sample auto-regressively from Eq. (5), with a streaming context  $X$ ,

$$\tilde{Y}_1 \sim q_{\text{aligned}}(\tilde{Y}_1|X_1), \quad \tilde{Y}_t \sim q_{\text{aligned}}(\tilde{Y}_t|X_{\leq t}, \tilde{Y}_{<t}). \quad (6)$$

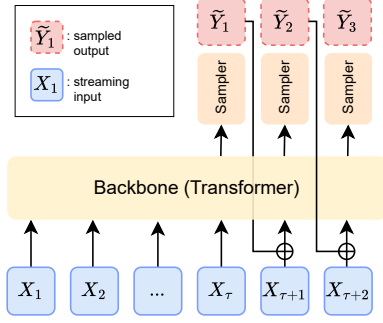


Figure 2: **DSM Architecture.** Transformer is fed with the streaming input  $X_t$ . After a delay  $\tau$ , a sampler is fed with the output of the backbone samples  $\tilde{Y}_t$ . At the next step, the backbone receives both the sampled value and next streaming input, whose embeddings are summed.

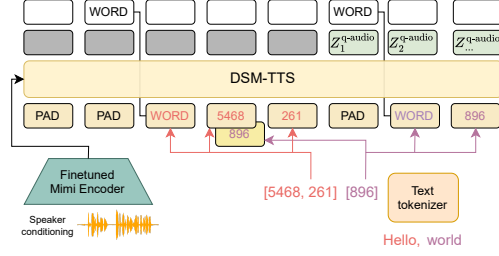


Figure 3: **DSM-TTS inference.** The input words “Hello, world” are tokenized. Until the model action stream outputs a WORD, it is fed with PAD. Then the first word’s tokens are fed, including a look-ahead text stream. Once a delay  $\tau = 5$  has accumulated, the model also outputs the audio.

118 We would want that given  $X \sim p(X)$ , then  $(X, \tilde{Y}) \sim (X, Y)$ , so that in particular  $\mathbb{P}[\tilde{Y}|X] \approx$   
 119  $\mathbb{P}[Y|X]$ . However this needs not be the case unless certain conditions are met.

120 **The importance of causality.** In particular, for  $(X, \tilde{Y}) \sim (X, Y)$  to be true,  $Y_{>t}$  must be independent  
 121 of  $X_{>t}$ , knowing  $X_{\leq t}$ . To realize that, one can look at a simple counter-example taking  $X_t \sim$   
 122  $\mathcal{B}(0.5)$  independent Bernoulli variables, and  $Y_t = X_t \oplus X_{t+1}$  the XOR of  $X_t$  and  $X_{t+1}$ . Clearly  
 123  $\mathbb{P}[Y_t|X_{\leq t}, Y_{<t}] \sim \mathcal{B}(0.5)$  for all  $t$ , yet, given  $X = (0, 1)$ , one would have

$$Y_1 = 1 \quad \text{a.s.}, \quad \tilde{Y}_1 \sim \mathcal{B}(0.5).$$

124 Thus  $Y|X$  and  $\tilde{Y}|X$  have different distributions. Intuitively, given that we do not sample  $X$  but  
 125 teacher-force real-world data, we must ensure that when sampling  $\tilde{Y}_t$ , no future value of  $X_{>t}$  might  
 126 end up in “contradiction” with the value we sampled.

127 **Delaying the output stream.** In practice, this is achieved by delaying the output stream  $Y_t$  by a  
 128 number of steps  $\tau > 0$ . Thus, we replace Eq. (5) by

$$q_\tau(y|X_{\leq t+\tau}, Y_{<t}) \approx \mathbb{P}[Y_t = y|X_{\leq t+\tau}, Y_{<t}], \quad (7)$$

129 and define  $\tilde{Y}^\tau$ , similarly to the procedure described in Eq. 6. Perfect independence is hard to achieve:  
 130 in the case of ASR, a named entity might be ambiguous without context, and only future development  
 131 in a discussion would resolve this ambiguity. Taking  $\tau = T$  recovers the prefixing or cross-attention  
 132 approaches presented earlier. In practice, there is a trade-off between the level of independence of  $Y_t$   
 133 with  $X_{>t+\tau}$ , and the latency of the method.

134 **Architecture of a DSM.** In practice, a DSM, depicted in Figure 2, involves three components: (i) an  
 135 auto-regressive backbone, (ii) an input embedder for  $X$  and  $Y$  into the backbone, and (iii) a sampler  
 136 for  $\tilde{Y}^\tau$  conditioned on the output of the backbone. The backbone can be a Transformer architecture,  
 137 optionally equipped with cross-attention layers to provide further non-streaming contextual  
 138 information. The embedder for  $X$  and  $Y$  can be learnt embedding tables in the case where both  
 139 domains are discrete, which are summed before going into the backbone. On the output side, we mask  
 140 the loss on the tokens of  $X$  and only compute cross-entropy on  $Y$ . Finally, the conditional sampler  
 141 can be a linear layer applied to the output of the backbone to derive logits if  $Y$  is discrete. It could  
 142 also be a flow or diffusion model conditioned on the output of the backbone for the continuous case.

### 143 3.1 Representations of the speech and text domains

144 We demonstrate the DSM framework on ASR and TTS, where the two domains are text and audio.

145 **Audio.** Given a waveform  $w \in \mathbb{R}^{d_s \cdot f_s}$  with the duration in seconds  $d_s$  and the sample rate  
 146  $f_s = 24$  kHz, we turn it into a more compact latent space using the Mimi codec (Défossez et al.,

2024), giving us a sequence of tensors  $Z^{\text{audio}} \in \mathbb{R}^{d_s \cdot f_r \times d_{\text{audio}}}$ , with a frame rate of  $f_r = 12.5$  Hz. This latent space is discretized with Residual Vector Quantization (Zeghidour et al., 2022) (RVQ), giving us a set of  $Q \in \llbracket 1, 32 \rrbracket$  coarse-to-fine discrete values per time step with cardinality  $N_a=2048$ , each coming from one codebook in the RVQ, giving a quantized representation  $Z^{\text{q-audio}} \in \{1, \dots, N_a\}^{d_s \cdot f_r \times Q}$ .

**Text.** We tokenize text using a vocabulary of  $N_t$ , specifically trained on speech data transcriptions. Two tokens have a special meaning: PAD (indicating the absence of words at this time) and WORD (indicating the start of a new word) following Défossez et al. (2024). Given a transcript, with word-level timestamps, of a waveform of duration  $d_s$ , its aligned text representation is  $Z^{\text{text}} \in \{1, \dots, N_t\}^{d_s \cdot f_r}$ . For each word in the transcript represented by tokens  $(x_1, \dots, x_n) \in \{1, \dots, N_t\}^n$  and starting at  $s \in \mathbb{R}^+$  seconds, we define its start index  $i = \text{floor}(s \cdot f_r)$ , and store it as  $Z_i^{\text{text}} \leftarrow \text{WORD}$ ,  $Z_{i+1}^{\text{text}} \leftarrow x_1$ ,  $Z_{i+2}^{\text{text}} \leftarrow x_2$ , etc. Any step in  $Z^{\text{text}}$  not assigned by a word token is given the special value PAD.

### 3.2 DSM for automatic speech recognition: DSM-ASR

For ASR, we consider  $X = Z^{\text{q-audio}}$  and  $Y = Z^{\text{text}}$ . By predicting the word tokens of  $Y$ , we learn to transcribe audio, while computing the loss on PAD and WORD tokens trains the model to predict the precise boundaries of each word. At inference time, we teacher-force the audio tokens of  $X$  and sample the full sequence  $Z^{\text{text}}$  to obtain a transcription along with timestamps with a precision of 80ms (frame size). This is allowed by the fact that we apply a constant delay to all words in the sequence, meaning we only need to shift the output timestamps back by the same value to recover the true timestamps.

**Deriving aligned speech-text data.** We are looking for fine-grained alignment between speech and text, however speech datasets are typically aligned at the level of the sentence (Panayotov et al., 2015). Conveniently, *whisper-timestamped* (Louradour, 2023) provides automatic transcriptions with word-level timestamps. We rely on these pseudo-labels for the pretraining phase of DSM-ASR. We then finetune on a mixture of public datasets with ground-truth transcripts (see details in Section 4.2), which pose two challenges. First, the automatic transcriptions extracted by Whisper in pretraining are formatted with capitalization and punctuation, but the level of formatting varies a lot between datasets. To address this, we train a 300M prefix-LM for automatic formatting, on a dataset of formatted Whisper transcripts. A second challenge is that these ground-truth transcripts do not have word-level alignment. We derive those by producing pseudo-transcripts with Whisper, and reconciling them with the formatted transcript using a Dynamic Time Warping algorithm (Giorgino, 2009).

**Delay conditioning for inference-time control.** As shown in Section 4.3.1, transcription quality is heavily dependent on the delay between audio and text. Thus, training DSM-ASR with a fixed delay requires choosing a latency/quality trade-off beforehand, and retraining a new model for each delay, despite the training task remaining fundamentally the same. To instead control this trade-off at inference, we train DSM-ASR over random delays, sampled for each sequence. The model is additionally conditioned on a cosine embedding (Vaswani et al., 2017) of the delay (expressed in milliseconds), added to the inputs. Experiments in Section 4.3.1 show that this delay conditioning performs at least as well as models with a fixed delay, and that the effective delay precisely respects the conditioning value.

### 3.3 DSM for text-to-speech

We further apply DSM to TTS, taking  $X = Z^{\text{text}}$ ,  $Y = Z^{\text{q-audio}}$ . We use a stream delay of 2s (or 25 steps) on the output audio. For sampling along the  $Q$  dimension in  $Z^{\text{q-audio}}$ , we use a RQ-Transformer as a sampler (Lee et al., 2022; Défossez et al., 2024), i.e. a smaller Transformer conditioned on the output of the backbone at each timestep and performing autoregressive modeling along the  $Q$  dimension. All the backbone inputs (generated audio tokens and next word token input) are fed through learnt embeddings and summed. We are confronted with the problem that the input domain is no longer plain text, but text properly padded for time alignment. While at train time we can teacher-force the ground-truth padded text, this is not the case for a novel text to synthesize at inference time.

**Action output stream.** We add an extra stream to the TTS outputs, whose goal is to predict whether the next input text token will be a WORD token or not. This special input token indicates that a new word is starting, and that its tokens are going to follow as inputs. This extra stream controls an inference-time *action*: when predicted by the model, we will feed as input the text tokens for the next word over the next time steps. While these are being fed, the model is not allowed to output another WORD action. The action output stream is not fed back into the model as it is redundant with the text stream input.

Table 1: **Short-form ASR performance.** We report Word Error Rates (WER, %) for DSM-ASR and selected non-streaming baselines from the OpenASR leaderboard, along with streaming baselines.

MODEL	AVG.	AMI	EARNINGS22	GIGASPEECH	LS CLEAN	LS OTHER	SPGISPEECH	TED-LIUM	VOXPOPULI
<i>Non-streaming</i>									
WHISPER MEDIUM.EN	8.1	16.7	12.6	11.0	3.0	5.9	3.3	4.1	9.6
WHISPER LARGE-V3	7.5	16.0	11.3	10.0	2.0	3.9	2.9	3.9	9.5
CRISPERWHISPER	6.7	<b>8.7</b>	12.9	10.2	1.8	4.0	2.7	3.2	9.8
CANARY-FLASH	6.4	13.1	12.8	9.9	<b>1.5</b>	<b>2.9</b>	<b>2.0</b>	3.1	<b>5.6</b>
PHI-4 MULTIMODAL	<b>6.1</b>	11.5	<b>10.5</b>	9.8	1.7	3.8	3.1	<b>2.9</b>	5.9
PARAKEET-TDT-V2	<b>6.1</b>	11.2	11.2	<b>9.7</b>	1.7	3.2	2.2	3.4	6.0
<i>Streaming</i>									
WHISPER MEDIUM.EN	9.0	22.1	13.4	10.4	3.0	6.2	3.7	4.7	8.6
WHISPER LARGE-V3	9.4	18.4	11.0	10.0	8.4	12.6	3.2	3.8	7.9
DSM-ASR	<b>6.3</b>	<b>11.7</b>	<b>10.6</b>	<b>9.7</b>	<b>1.7</b>	<b>4.3</b>	<b>2.0</b>	<b>3.4</b>	<b>6.8</b>

**Lookahead second text stream.** The action stream allows the model to predict the next word position, although the model has no knowledge of its content for making that decision. The delay between text and audio only provides context for the audio generation, however, the decision on where to insert pauses and words has no such context. Given a sequence of words  $m_1, m_2, \dots$ , the lookahead text stream feeds the tokens of the words  $m_{i+l}$  to the backbone while the primary text feed contains the tokens of words  $m_i$ .

**Speaker conditioning.** We provide speaker embeddings for up to 5 speakers. Each speaker is represented by a 10s audio extract of the same speaker outside of the training segment. Speakers are identified using the diarization tool Pyannote (Bredin, 2023). One speaker is elected the main speaker. When a turn of the main speaker starts, its first word is prefixed with a special MAIN token, while when any other speaker turn starts, it is prefixed with OTHER. This allows us to generate dialogs with control over change of turns and speaker voices. Each speaker audio extract is encoded with a copy of the Mimi encoder, whose Transformer is fine tuned along with the main TTS model, while convolution layers are kept frozen for stability. We concatenate all the speaker embeddings, sum them with an absolute positional embedding (Vaswani et al., 2017), and feed them through cross-attention layers to the backbone. An overview of the whole DSM-TTS inference process is shown in Figure 3.

## 4 Experiments

### 4.1 Architectural hyperparameters

We use a Transformer (Vaswani et al., 2017) backbone with RoPE positional encoding (Su et al., 2024). For the DSM-TTS experiments, we use a 1B parameters backbone with a 2048 dimension latent space, GLU feed-forward units, 16 layers, and 16 heads of attention. The DSM-ASR uses a 3B parameters backbone, with 2048 dimensions, 48 layers, and 32 attention heads, and a linear to predict the logits over the text vocabulary, with a cardinality  $N_t^{\text{asr}} = 4000$ , trained for English only. The model uses a variable delay  $\tau$  which is sampled per batch item in a range of  $[0.25, 4]$ s (Section 3.2). The TTS model also receives the speaker embedding through cross-attention. The sampler is a Transformer along the codebook  $Q$  dimension described in Section 3.1, with no context over the time axis, with a dimension of 1024, 6 layers for each codebook, with a linear layer to estimate the logits. The text tokenizer is trained on bilingual French/English data, with a cardinality  $N_t^{\text{tts}} = 8000$ . The model uses a delay  $\tau$  of 2s, or 25 steps, and a lookahead stream with  $l = 2$  (Section 3.3). We use AdamW (Loshchilov & Hutter, 2019), a cosine learning rate schedule with linear warmup, with an initial rate of  $2 \cdot 10^{-4}$  for the TTS, and  $4 \cdot 10^{-4}$  for the ASR, and a weight decay of 0.1.

### 4.2 Training protocol

**Pretraining.** We use an audio collection of 2.5 million hours of publicly available audio content in English, and French transcribed with whisper-timestamped. Given the synthetic nature of text transcripts, this phase amounts to hard distillation of whisper-timestamped. We train DSM-ASR on random 90s segments for 1.6M steps, on 48 H100s. DSM-TTS is trained on 120s audio extracts, on 16 H100s, 250k updates with batch size 64.

Table 2: **Long-form ASR performance.** We report Word Error Rates (WER, %) across four long-form datasets for DSM-ASR and a set of streaming and non-streaming baselines.

MODEL	AVG.	TED-LIUM	MEANWHILE	REV16	EARNINGS21
<i>Non-streaming</i>					
DISTIL-LARGE V2	8.7	3.7	7.8	12.2	11.2
WHISPER-LARGE-V2	9.0	4.4	6.3	13.6	11.8
<i>Streaming</i>					
WHISPER MEDIUM.EN	9.0	3.9	6.7	13.0	12.5
WHISPER LARGE-V3	8.1	3.4	6.1	<b>11.4</b>	11.4
DSM-ASR	<b>7.9</b>	<b>2.9</b>	<b>5.7</b>	12.3	<b>10.6</b>

**Finetuning (DSM-ASR).** We then finetune the model on a collection of public datasets with ground-truth transcripts, described in Appendix A.1 and totaling 24 hours. This training stage lasts for 100k updates with batches of 128 examples, using 16 H100s.

We then adapt the model to long-form inputs, which most public datasets lack, by constructing a long-form mixture described in Appendix A.2. We run this stage for DSM-ASR for 25k updates with batch size 32, using 16 H100s.

### 4.3 Automatic Speech Recognition

We evaluate DSM-ASR (with a default delay of 2.5s) in terms of transcription quality, latency, and timestamps precision. We consider short-form transcription (shorter than 30s), as it is the focus of the OpenASR Leaderboard (Srivastav et al., 2023). We also look at streaming inference for long-form transcription (up to 2 hours).

**Baselines.** We benchmark DSM-ASR against leading models of the OpenASR Leaderboard, including Whisper (Radford et al., 2023), Canary-Flash (Zelasko et al., 2025), Phi-4 Multimodal Instruct (Abouelenin et al., 2025) and Parakeet (Xu et al., 2023). Notably, all these models perform non-streaming ASR, as they require access to the full input sequence. We thus also include a streaming variant of Whisper (Macháček et al., 2023), with a delay of 2.5s for a fair comparison. For long-form transcription, we add the Distil-Whisper (Gandhi et al., 2023) variant.

**Transcription quality.** We report micro-averaged Word Error Rate (WER), which avoids over-emphasizing short sequences, and is the standard computation used in the OpenASR Leaderboard, of which we use the official evaluation codebase.<sup>1</sup> Throughout this Section, we use Whisper normalizer for English (Radford et al., 2023).

**Latency.** We evaluate the latency of streaming models as the average delay between the real timestamp of a word, and the time when this word is transcribed in the output. In the absence of ground-truth timestamps, we use pseudo-timestamps on Librispeech test-clean (Panayotov et al., 2015) provided by Lugosch et al. (2019). These timestamps were obtained by Montreal Forced Aligner (McAuliffe et al., 2025), and use them as reference.

**Timestamps.** See Appendix C for a complete description.

#### 4.3.1 ASR Results

**Short-form transcription.** Table 1 shows that DSM-ASR is significantly better than streaming baselines, and even competitive with the best, non-streaming models of the OpenASR Leaderboard. With an average WER of 6.3%—while 6.1% being the current best score of the leaderboard—DSM-ASR is remarkably the only streaming model among top ASR systems. In Appendix C, we see that, in terms of timestamp precision, DSM-ASR performs significantly better than Whisper Large-v3 while somewhat underperforming CrisperWhisper, though with a better WER.

**Long-form transcription.** Table 2 reports WER values across 4 long-form datasets with sequences up to 2 hours: TED-LIUM (Hernandez et al., 2018), Meanwhile (Radford et al., 2023), Rev16 (Radford et al., 2023), and Earnings21 (Rio et al., 2021). We see that DSM-ASR outperforms both streaming and non-streaming baselines.

<sup>1</sup>[https://github.com/huggingface/open\\_asr\\_leaderboard](https://github.com/huggingface/open_asr_leaderboard)

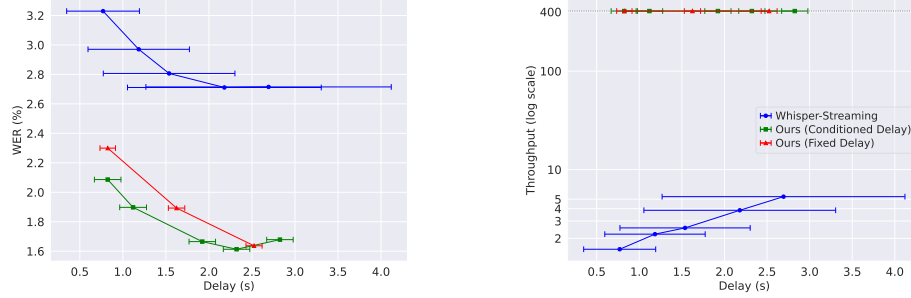


Figure 4: **ASR WER and throughput vs. delay.** Word Error Rate (WER, %) (left) and throughput (right) in function of the delay. Throughput is the product between Real-Time Factor and batch size.

Table 3: **Long-form TTS WER.** We compare open-source and closed-source baselines. Short-form inference is ran with a short context. ElevenLabs is evaluated on a limited subset due to its cost.

MODEL	WER ENGLISH (%) (↓)			WER FRENCH (%) (↓)		
	AVG.	DIALOGS	MONOLOGUES	AVG.	DIALOGS	MONOLOGUES
Short-form inference						
DIA	6.7	7.8	5.7	24.9	22.4	27.3
CSM	4.7	3.9	5.4	-	-	-
ORPHEUS	7.1	3.8	10.5	-	-	-
Long-form inference						
CSM	15.5	15.7	15.4	-	-	-
DSM-TTS (OURS)	3.6	2.8	4.3	6.4	3.7	9.1
Long-form inference, small subset						
DSM-TTS (OURS)	7.5	10.4	4.5	8.1	8.9	7.4
11LABS FLASH v2.5	2.5	1.0	4.0	4.5	4.1	4.9
11LABS MULTILINGUAL v2	2.1	1.0	3.1	2.9	2.8	3.0

**Delay conditioning and latency.** Figure 4 (left) compares the WER obtained for DSM-ASR with and without delay conditioning, along with Whisper-Streaming. We observe that the delay of Whisper-Streaming has a large variance, while DSM-ASR has a precision of  $\sim 300$ ms around its target delay. Interestingly, training a single DSM-ASR model with delay conditioning outperforms fixed delay variants. Figure 4 (right) shows the throughput on an H100 GPU: DSM-ASR can process 400 sequences simultaneously while being real-time and its throughput is independent of the delay. This is unlike Whisper-Streaming which reduces its delay by re-evaluating the partial input sequence more frequently, increasing the computational cost. Combined with the fact that Whisper-Streaming does not allow for batching, this results in a 100x lower throughput than that of DSM-ASR.

#### 4.4 Text-To-Speech experiments

**Evaluation datasets.** We collect a novel dataset for long-form TTS evaluation in English and French. We first use news articles from the NTREX-128 (Federmann et al., 2022) text dataset, given 123 monologues per language. To evaluate controllable dialog capabilities, we use 110 synthetic scripts per language generated by a LLM, spanning three categories: daily life, technical, and number-heavy discussions. For voice conditioning, we use samples from the test set of VCTK (Yamagishi et al., 2019) for English, and from the test and valid sets of CML (Oliveira et al., 2023) for French. We provide examples and more details in the Appendix B, and we will release this dataset.

**Metrics.** We evaluate the per-document WER, using text normalization from Whisper (Radford et al., 2023). We collect subjective metrics covering both the speaker similarity to the conditioning and overall speech quality, see Appendix A for more details.

**Baselines.** We compare to open-source models Dia, Orpheus, and CSM, as well as the closed-source ElevenLabs API.<sup>2</sup> Dia and ElevenLabs support French and English, while Orpheus and CSM only support English. Dia, Orpheus and CSM can be speaker-conditioned through prefixing, with Dia

<sup>2</sup>Available at nari-labs/dia, canopyai/Orpheus-TTS, SesameAILabs/csm and elevenlabs.io.



Table 4: **Subjective evaluations on TTS.** We compare with baselines over two axes through human evaluations: speech quality, measured as MUSHRA scores (1–100, along with std. of the mean), and speaker-similarity win-rates, summarized as ELO scores (with 95% confidence intervals).

MODEL	ENGLISH		FRENCH	
	QUALITY (↑)	SPK. SIM. (↑)	QUALITY(↑)	SPK. SIM.(↑)
DIA	52.9 ± 2.0	1930.0 ± 22.6	30.0 ± 2.0	1578.7 ± 53.8
CSM	43.0 ± 1.4	2056.0 ± 18.6	-	-
ORPHEUS	33.7 ± 1.8	1820.5 ± 32.7	-	-
DSM-TTS (OURS)	50.5 ± 1.8	<b>2066.9</b> ± 22.8	52.1 ± 1.6	2078.5 ± 18.0
11LABS FLASH V2.5	<b>64.1</b> ± 1.7	1977.8 ± 22.6	65.7 ± 1.8	1926.0 ± 15.7
11LABS MULTILINGUAL V2	61.3 ± 1.8	<b>2067.4</b> ± 23.5	<b>68.7</b> ± 1.6	<b>2099.3</b> ± 17.8

Table 5: **TTS: Latency and throughput.** We compare the inference performance of DSM-TTS and selected baselines. Real-Time Factor (RTF) is higher than 1 if the model can produce audio in real time. Throughput is the product between Real-Time-factor and batch size.

MODEL	MODEL SIZE	LATENCY (MS)(↓)	RTF(↑)	THROUGHPUT(↑)
DIA	1.6B	-	0.7	0.7
CSM	1.5B	-	1.0	1.0
ORPHEUS	3.8B	-	0.7	0.7
DSM-TTS B.S.=1	3.7B	185	2.7	2.7
DSM-TTS B.S.=32	3.7B	560	2.1	67.4
DSM-TTS B.S.=64	3.7B	708	1.7	111.3

and CSM supporting dialogs. For Orpheus and ElevenLabs, dialogs are emulated by concatenating single-speaker turns. Details of how baselines are evaluated are provided in Appendix E.

#### 4.4.1 TTS Results

**Main results.** As seen in Table 3, our approach provides the lowest WER across all languages for both monologues and dialogs. Our method is the only one to run long-form inference across all cases, CSM showing strong degradation when running with longer sequences, Dia only being trained for 20s output, and ElevenLab requiring per-turn generation for dialogs. In terms of subjective results, our model is on par with the commercial ElevenLab models for speaker similarity, outperforming all existing methods. In quality, it is equivalent to the best open-source baseline for English and much better for French, while lagging behind commercial models. Note that we kept all methods with their original sample-rate (e.g. 44.1kHz for ElevenLab) which contributes to the difference.

**Throughput and latency.** Our method is easily batchable, leading to gains in throughput while staying compatible with real-time generation. As shown in Table 5, on a single H100 the amount of audio generated is 100x real-time, more details are provided in Appendix G.

**DSM-ASR and DSM-TTS as a speech interface for LLMs.** We combine DSM-ASR, DSM-TTS, and Gemma 3 (Gemma Team et al., 2025) into an LLM voice chat application with sub-second latency. We provide conversation samples at <https://delayed-stream-modeling.github.io/>.

## 5 Conclusion

We introduce Delayed Streams Modeling, a flexible framework for streaming sequence-to-sequence learning. DSM provides a remarkable trade-off between quality and latency, and an unprecedented throughput among streaming models. Focusing on speech-text tasks, DSM-ASR is the first streaming ASR model to provide timestamped, formatted transcripts that competes with the top offline models, while DSM-TTS is competitive with non-streaming baselines while being the only model providing arbitrarily long synthesis. In future work, we will extend DSM to more sequential multimodal tasks.

**Limitations.** We acknowledge that streaming naturalistic speech with voice conditioning opens up both opportunities in inclusive human-machine interactions and risks of fraudulent impersonation. Addressing the latter requires that public access to such technologies is accompanied by proper user terms, voice verification mechanisms, and watermarking of generated content. Finally, the need for aligned domains reduces the amount of gold-standard ground-truth data that can be used for training.

## References

- Abouelenin, A., Ashfaq, A., Atkinson, A., Awadalla, H., Bach, N., Bao, J., Benhaim, A., Cai, M., Chaudhary, V., Chen, C., Chen, D., Chen, D., Chen, J., Chen, W., Chen, Y., Chen, Y., Dai, Q., Dai, X., Fan, R., Gao, M., Gao, M., Garg, A., Goswami, A., Hao, J., Hendy, A., Hu, Y., Jin, X., Khademi, M., Kim, D., Kim, Y. J., Lee, G., Li, J., Li, Y., Liang, C., Lin, X., Lin, Z., Liu, M., Liu, Y., Lopez, G., Luo, C., Madan, P., Mazalov, V., Mitra, A., Mousavi, A., Nguyen, A., Pan, J., Perez-Becker, D., Platin, J., Portet, T., Qiu, K., Ren, B., Ren, L., Roy, S., Shang, N., Shen, Y., Singhal, S., Som, S., Song, X., Sych, T., Vaddamanu, P., Wang, S., Wang, Y., Wang, Z., Wu, H., Xu, H., Xu, W., Yang, Y., Yang, Z., Yu, D., Zabir, I., Zhang, J., Zhang, L. L., Zhang, Y., and Zhou, X. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *CoRR*, abs/2503.01743, 2025. doi: 10.48550/ARXIV.2503.01743.
- Alayrac, J., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J. L., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *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*, 2022.
- Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- Barraut, L., Chung, Y., Meglioli, M. C., Dale, D., Dong, N., Duppenhaler, M., Duquenne, P., Ellis, B., Elsahar, H., Haaheim, J., Hoffman, J., Hwang, M., Inaguma, H., Klaiber, C., Kulikov, I., Li, P., Licht, D., Maillard, J., Mavlyutov, R., Rakotoarison, A., Sadagopan, K. R., Ramakrishnan, A., Tran, T., Wenzek, G., Yang, Y., Ye, E., Evtimov, I., Fernandez, P., Gao, C., Hansanti, P., Kalbassi, E., Kallet, A., Kozhevnikov, A., Gonzalez, G. M., Roman, R. S., Touret, C., Wong, C., Wood, C., Yu, B., Andrews, P., Balioglu, C., Chen, P., Costa-jussà, M. R., Elbayad, M., Gong, H., Guzmán, F., Heffernan, K., Jain, S., Kao, J., Lee, A., Ma, X., Mourachko, A., Peloquin, B. N., Pino, J., Popuri, S., Ropers, C., Saleem, S., Schwenk, H., Sun, A. Y., Tomasello, P., Wang, C., Wang, J., Wang, S., and Williamson, M. Seamless: Multilingual expressive and streaming speech translation. *CoRR*, abs/2312.05187, 2023. doi: 10.48550/ARXIV.2312.05187.
- Beyer, L., Steiner, A., Pinto, A. S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., Unterthiner, T., Keysers, D., Koppula, S., Liu, F., Grycner, A., Gritsenko, A. A., Houlsby, N., Kumar, M., Rong, K., Eisenschlos, J., Kabra, R., Bauer, M., Bosnjak, M., Chen, X., Minderer, M., Voigtlaender, P., Bica, I., Balazevic, I., Puigcerver, J., Papalampidi, P., Hénaff, O. J., Xiong, X., Soricut, R., Harmsen, J., and Zhai, X. Paligemma: A versatile 3b VLM for transfer. *CoRR*, abs/2407.07726, 2024. doi: 10.48550/ARXIV.2407.07726.
- Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: The method of paired comparisons. *Biometrika*, 39(3-4):324–345, 12 1952. ISSN 0006-3444. doi: 10.1093/biomet/39.3-4.324.
- Bredin, H. pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe. In *Proc. INTERSPEECH 2023*, 2023.
- Carletta, J. Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus. *Language Resources and Evaluation*, 41:181–190, 2007.
- Caron, F. and Doucet, A. Efficient bayesian inference for generalized bradley-terry models, 2010.
- Chan, W., Jaitly, N., Le, Q., and Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4960–4964. IEEE, 2016.

376 Chen, G., Chai, S., Wang, G., Du, J., Zhang, W.-Q., Weng, C., Su, D., Povey, D., Trmal, J., Zhang, J.,  
377 et al. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio.  
378 *arXiv preprint arXiv:2106.06909*, 2021.

379 Chen, Y., Niu, Z., Ma, Z., Deng, K., Wang, C., Zhao, J., Yu, K., and Chen, X. F5-tts: A fairytaler that  
380 fakes fluent and faithful speech with flow matching. *arXiv preprint arXiv:2410.06885*, 2024.

381 Chiu, C. and Raffel, C. Monotonic chunkwise attention. In *6th International Conference on Learning*  
382 *Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track*  
383 *Proceedings*. OpenReview.net, 2018.

384 Chiu, C., Kannan, A., Prabhavalkar, R., Chen, Z., Sainath, T. N., Wu, Y., Han, W., Zhang, Y.,  
385 Pang, R., Kishchenko, S., Nguyen, P., Narayanan, A., Liao, H., and Zhang, S. A comparison of  
386 end-to-end models for long-form speech recognition. In *IEEE Automatic Speech Recognition and*  
387 *Understanding Workshop, ASRU 2019, Singapore, December 14-18, 2019*, pp. 889–896. IEEE,  
388 2019. doi: 10.1109/ASRU46091.2019.9003854.

389 Défossez, A., Copet, J., Synnaeve, G., and Adi, Y. High fidelity neural audio compression. *Transac-*  
390 *tions on Machine Learning Research*, 2023.

391 Défossez, A., Mazaré, L., Orsini, M., Royer, A., Pérez, P., Jégou, H., Grave, E., and Zeghidour, N.  
392 Moshi: a speech-text foundation model for real-time dialogue. *CoRR*, abs/2410.00037, 2024. doi:  
393 10.48550/ARXIV.2410.00037.

394 Del Rio, M., Ha, P., McNamara, Q., Miller, C., and Chandra, S. Earnings-22: A practical benchmark  
395 for accents in the wild. *arXiv preprint arXiv:2203.15591*, 2022.

396 Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis.  
397 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12868–  
398 12878, 2020.

399 Federmann, C., Kocmi, T., and Xin, Y. NTREX-128 – news test references for MT evaluation of  
400 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pp.  
401 21–24, Online, nov 2022. Association for Computational Linguistics.

402 Gandhi, S., von Platen, P., and Rush, A. M. Distil-whisper: Robust knowledge distillation via  
403 large-scale pseudo labelling. *CoRR*, abs/2311.00430, 2023. doi: 10.48550/ARXIV.2311.00430.

404 Gemma Team, Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova,  
405 T., Ramé, A., Rivière, M., Rouillard, L., Mesnard, T., Cideron, G., bastien Grill, J., Ramos, S.,  
406 Yvinec, E., Casbon, M., Pot, E., Penchev, I., Liu, G., Visin, F., Kenealy, K., Beyer, L., Zhai, X.,  
407 Tsitsulin, A., Busa-Fekete, R., Feng, A., Sachdeva, N., Coleman, B., Gao, Y., Mustafa, B., Barr,  
408 I., Parisotto, E., Tian, D., Eyal, M., Cherry, C., Peter, J.-T., Sinopalnikov, D., Bhupatiraju, S.,  
409 Agarwal, R., Kazemi, M., Malkin, D., Kumar, R., Vilar, D., Brusilovsky, I., Luo, J., Steiner, A.,  
410 Friesen, A., Sharma, A., Sharma, A., Gilady, A. M., Goedeckemeyer, A., Saade, A., Feng, A.,  
411 Kolesnikov, A., Bendebury, A., Abdagic, A., Vadi, A., György, A., Pinto, A. S., Das, A., Bapna,  
412 A., Miech, A., Yang, A., Paterson, A., Shenoy, A., Chakrabarti, A., Piot, B., Wu, B., Shahriari,  
413 B., Petrini, B., Chen, C., Lan, C. L., Choquette-Choo, C. A., Carey, C., Brick, C., Deutsch, D.,  
414 Eisenbud, D., Cattle, D., Cheng, D., Paparas, D., Sreepathihalli, D. S., Reid, D., Tran, D., Zelle, D.,  
415 Noland, E., Huizenga, E., Kharitonov, E., Liu, F., Amirkhanyan, G., Cameron, G., Hashemi, H.,  
416 Klimczak-Plucińska, H., Singh, H., Mehta, H., Lehri, H. T., Hazimeh, H., Ballantyne, I., Szpektor,  
417 I., Nardini, I., Pouget-Abadie, J., Chan, J., Stanton, J., Wieting, J., Lai, J., Orbay, J., Fernandez,  
418 J., Newlan, J., yeong Ji, J., Singh, J., Black, K., Yu, K., Hui, K., Vodrahalli, K., Greff, K., Qiu,  
419 L., Valentine, M., Coelho, M., Ritter, M., Hoffman, M., Watson, M., Chaturvedi, M., Moynihan,  
420 M., Ma, M., Babar, N., Noy, N., Byrd, N., Roy, N., Momchev, N., Chauhan, N., Sachdeva, N.,  
421 Bunyan, O., Botarda, P., Caron, P., Rubenstein, P. K., Culliton, P., Schmid, P., Sessa, P. G., Xu, P.,  
422 Stanczyk, P., Tafti, P., Shivanna, R., Wu, R., Pan, R., Rokni, R., Willoughby, R., Vallu, R., Mullins,  
423 R., Jerome, S., Smoot, S., Girgin, S., Iqbal, S., Reddy, S., Sheth, S., Pöder, S., Bhatnagar, S.,  
424 Panyam, S. R., Eiger, S., Zhang, S., Liu, T., Yacovone, T., Liechty, T., Kalra, U., Evci, U., Misra,  
425 V., Roseberry, V., Feinberg, V., Kolesnikov, V., Han, W., Kwon, W., Chen, X., Chow, Y., Zhu, Y.,  
426 Wei, Z., Egyed, Z., Cotruta, V., Giang, M., Kirk, P., Rao, A., Black, K., Babar, N., Lo, J., Moreira,  
427 E., Martins, L. G., Sanseviero, O., Gonzalez, L., Gleicher, Z., Warkentin, T., Mirrokni, V., Senter,

428 E., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Matias, Y., Sculley, D., Petrov, S., Fiedel,  
429 N., Shazeer, N., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya,  
430 E., Alayrac, J.-B., Anil, R., Dmitry, Lepikhin, Borgeaud, S., Bachem, O., Joulin, A., Andreev, A.,  
431 Hardin, C., Dadashi, R., and Hussenot, L. Gemma 3 technical report, 2025.

432 Giorgino, T. Computing and visualizing dynamic time warping alignments in r: the dtw package.  
433 *Journal of statistical Software*, 31:1–24, 2009.

434 Graves, A., Mohamed, A., and Hinton, G. E. Speech recognition with deep recurrent neural networks.  
435 In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013,*  
436 *Vancouver, BC, Canada, May 26-31, 2013*, pp. 6645–6649. IEEE, 2013. doi: 10.1109/ICASSP.  
437 2013.6638947.

438 Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., and Esteve, Y. Ted-lium 3: Twice as much  
439 data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer: 20th*  
440 *International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings*  
441 *20*, pp. 198–208. Springer, 2018.

442 Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

443 Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780,  
444 1997. doi: 10.1162/neco.1997.9.8.1735.

445 Kreuk, F., Synnaeve, G., Polyak, A., Singer, U., Défossez, A., Copet, J., Parikh, D., Taigman, Y., and  
446 Adi, Y. Audiogen: Textually guided audio generation. In *The Eleventh International Conference*  
447 *on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

448 Labiausse, T., Mazaré, L., Grave, E., Pérez, P., Défossez, A., and Zeghidour, N. High-fidelity  
449 simultaneous speech-to-speech translation, 2025.

450 Lee, D., Kim, C., Kim, S., Cho, M., and Han, W. Autoregressive image generation using residual  
451 quantization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*  
452 *2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11513–11522. IEEE, 2022. doi: 10.1109/  
453 CVPR52688.2022.01123.

454 Li, B., Gulati, A., Yu, J., Sainath, T. N., Chiu, C., Narayanan, A., Chang, S., Pang, R., He, Y.,  
455 Qin, J., Han, W., Liang, Q., Zhang, Y., Strohman, T., and Wu, Y. A better and faster end-to-end  
456 model for streaming ASR. In *IEEE International Conference on Acoustics, Speech and Signal*  
457 *Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pp. 5634–5638. IEEE, 2021.  
458 doi: 10.1109/ICASSP39728.2021.9413899.

459 Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *7th International Conference*  
460 *on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

461 Louradour, J. whisper-timestamped. <https://github.com/linto-ai/whisper-timestamped>,  
462 2023.

463 Lugosch, L., Ravanelli, M., Ignoto, P., Tomar, V. S., and Bengio, Y. Speech model pre-training for  
464 end-to-end spoken language understanding. *arXiv preprint arXiv:1904.03670*, 2019.

465 Macháček, D., Dabre, R., and Bojar, O. Turning whisper into real-time transcription system. In  
466 Saha, S. and Sujaini, H. (eds.), *Proceedings of the 13th International Joint Conference on Natural*  
467 *Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for*  
468 *Computational Linguistics: System Demonstrations*, Bali, Indonesia, November 2023. Association  
469 for Computational Linguistics.

470 McAuliffe, M., Fatchurrahman, M. R., Feiteng, GalaxieT, NTT123, Gulati, A., Coles, A., Kong, C.,  
471 Veaux, C., Eren, E., Gritskevich, E., Thor, G., Mishra, H., Fruehwald, J., Potrykus, P., Sereda, T.,  
472 Mestrou, T., michaelasocolof, and vannawillerton. Montrealcorpustools/montreal-forced-aligner:  
473 Version 3.2.2, March 2025.

474 Oliveira, F. S., Casanova, E., Junior, A. C., Soares, A. S., and Galvão Filho, A. R. Cml-tts: A  
475 multilingual dataset for speech synthesis in low-resource languages. In Ekšteín, K., Pártl, F.,  
476 and Konopík, M. (eds.), *Text, Speech, and Dialogue*, pp. 188–199, Cham, 2023. Springer Nature  
477 Switzerland. ISBN 978-3-031-40498-6.

478 O’Neill, P. K., Lavrukhin, V., Majumdar, S., Noroozi, V., Zhang, Y., Kuchaiev, O., Balam, J.,  
479 Dovzhenko, Y., Freyberg, K., Shulman, M. D., et al. Spgispeech: 5,000 hours of transcribed  
480 financial audio for fully formatted end-to-end speech recognition. *arXiv preprint arXiv:2104.02014*,  
481 2021.

482 Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: An ASR corpus based on  
483 public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and*  
484 *Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pp.  
485 5206–5210. IEEE, 2015. doi: 10.1109/ICASSP.2015.7178964.

486 Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech  
487 recognition via large-scale weak supervision. In Krause, A., Brunskill, E., Cho, K., Engelhardt,  
488 B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML*  
489 *2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning*  
490 *Research*, pp. 28492–28518. PMLR, 2023.

491 Raffel, C., Luong, M., Liu, P. J., Weiss, R. J., and Eck, D. Online and linear-time attention by  
492 enforcing monotonic alignments. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th*  
493 *International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August*  
494 *2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2837–2846. PMLR, 2017.

495 Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I.  
496 Zero-shot text-to-image generation. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th*  
497 *International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*,  
498 volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 2021.

499 Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2.  
500 In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., and Garnett, R. (eds.),  
501 *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

502 Renals, S., Hain, T., and Boulard, H. Recognition and understanding of meetings the ami and amida  
503 projects. In *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pp.  
504 238–247. IEEE, 2007.

505 Rio, M. D., Delworth, N., Westerman, R., Huang, M., Bhandari, N., Palakapilly, J., McNamara, Q.,  
506 Dong, J., Zelasko, P., and Jette, M. Earnings-21: A practical benchmark for ASR in the wild.  
507 *CoRR*, abs/2104.11348, 2021.

508 Rubenstein, P. K., Asawaroengchai, C., Nguyen, D. D., Bapna, A., Borsos, Z., de Chaumont Quirry,  
509 F., Chen, P., Badawy, D. E., Han, W., Kharitonov, E., Muckenhirn, H., Padfield, D., Qin, J.,  
510 Rozenberg, D., Sainath, T. N., Schalkwyk, J., Sharifi, M., Ramanovich, M. T., Tagliasacchi, M.,  
511 Tudor, A., Velimirovic, M., Vincent, D., Yu, J., Wang, Y., Zayats, V., Zeghidour, N., Zhang, Y.,  
512 Zhang, Z., Zilka, L., and Frank, C. H. Audiopalm: A large language model that can speak and  
513 listen. *CoRR*, abs/2306.12925, 2023. doi: 10.48550/ARXIV.2306.12925.

514 Srivastav, V., Majumdar, S., Koluguri, N., Moumen, A., Gandhi, S., et al. Open automatic  
515 speech recognition leaderboard. [https://huggingface.co/spaces/hf-audio/open\\_asr\\_](https://huggingface.co/spaces/hf-audio/open_asr_leaderboard)  
516 [leaderboard](https://huggingface.co/spaces/hf-audio/open_asr_leaderboard), 2023.

517 Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary  
518 position embedding. *Neurocomputing*, 568:127063, 2024.

519 Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks.  
520 *Advances in neural information processing systems*, 27, 2014.

521 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., and and, L. K. Attention  
522 is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008,  
523 2017.

524 Wagner, L., Thallinger, B., and Zusag, M. Crisperwhisper: Accurate timestamps on verbatim speech  
525 transcriptions. *CoRR*, abs/2408.16589, 2024. doi: 10.48550/ARXIV.2408.16589.

526 Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., and  
527 Dupoux, E. Voxpopuli: A large-scale multilingual speech corpus for representation learning,  
528 semi-supervised learning and interpretation. *arXiv preprint arXiv:2101.00390*, 2021.

529 Wang, C., Chen, S., Wu, Y., Zhang, Z., Zhou, L., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., He,  
530 L., Zhao, S., and Wei, F. Neural codec language models are zero-shot text to speech synthesizers.  
531 *CoRR*, abs/2301.02111, 2023. doi: 10.48550/ARXIV.2301.02111.

532 Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen,  
533 Z., Bengio, S., Le, Q. V., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. Tacotron: Towards  
534 end-to-end speech synthesis. In Lacerda, F. (ed.), *18th Annual Conference of the International  
535 Speech Communication Association, Interspeech 2017, Stockholm, Sweden, August 20-24, 2017*,  
536 pp. 4006–4010. ISCA, 2017. doi: 10.21437/INTERSPEECH.2017-1452.

537 Xu, H., Jia, F., Majumdar, S., Huang, H., Watanabe, S., and Ginsburg, B. Efficient sequence  
538 transduction by jointly predicting tokens and durations. In Krause, A., Brunskill, E., Cho, K.,  
539 Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning,  
540 ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine  
541 Learning Research*, pp. 38462–38484. PMLR, 2023.

542 Yamagishi, J., Veaux, C., and MacDonald, K. Cstr vctk corpus: English multi-speaker corpus for cstr  
543 voice cloning toolkit (version 0.92), 2019.

544 Yu, L., Shi, B., Pasunuru, R., Muller, B., Golovneva, O. Y., Wang, T., Babu, A., Tang, B., Karrer, B.,  
545 Sheynin, S., Ross, C., Polyak, A., Howes, R., Sharma, V., Xu, P., Tamoyan, H., Ashual, O., Singer,  
546 U., Li, S.-W., Zhang, S., James, R., Ghosh, G., Taigman, Y., Fazel-Zarandi, M., Celikyilmaz, A.,  
547 Zettlemoyer, L., and Aghajanyan, A. Scaling autoregressive multi-modal models: Pretraining and  
548 instruction tuning. *ArXiv*, abs/2309.02591, 2023.

549 Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., and Tagliasacchi, M. Soundstream: An end-to-end  
550 neural audio codec. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30:495–507, 2022. doi:  
551 10.1109/TASLP.2021.3129994.

552 Zelasko, P., Dhawan, K., Galvez, D., Puvvada, K. C., Pasad, A., Koluguri, N. R., Hu, K., Lavrukhin,  
553 V., Balam, J., and Ginsburg, B. Training and inference efficiency of encoder-decoder speech  
554 models. *CoRR*, abs/2503.05931, 2025. doi: 10.48550/ARXIV.2503.05931.

555 Zhang, S., Fang, Q., Guo, S., Ma, Z., Zhang, M., and Feng, Y. Streamspeech: Simultaneous speech-  
556 to-speech translation with multi-task learning. In Ku, L., Martins, A., and Srikumar, V. (eds.),  
557 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume  
558 1: Long Papers)*, *ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 8964–8986. Association  
559 for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.485.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction describe the nature of the technical contribution and the application results (ASR and TTS).

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Due to the limited space, we briefly address limitations in the conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[No\]](#)

Justification: We do not claim any theorems. However, when introducing the DSM framework in Section 3, we do claim that the conditional independence of  $Y_t$  with future inputs  $X_{>t}$  is necessary and sufficient for  $\tilde{Y}$  to follow the right distribution. We only show the necessary direction through a counter example.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We describe our training datasets and hyperparameters, and will release the exact models used for producing experimental results, along with the codebase to reproduce our quantitative results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.



## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Although we will release inference code along with our new TTS evaluation dataset, neither is ready at time of submission, hence the no answer.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide an extensive description of training datasets and either perform evaluation on public benchmarks, or on new datasets that we will release.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report error bars for human evaluations of TTS, as well as error bars for the latency of ASR models. We do not provide error bars for WERs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We specify the number and type of GPU along with the number of training steps and batch size.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: In particular, the evaluation dataset that we collected is made of textual data either original or in the public domain, along with voice samples from public academic datasets.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the risk associated with voice cloning.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [\[Yes\]](#)

Justification: We will release a version of TTS without voice cloning.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: We provide references for all existing datasets that we use, and credit all the packages (whisper-timestamped, DTW, OpenASRLeaderboard) used in our experiments.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We describe the evaluation dataset in appendix.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[Yes\]](#)

Justification: We work with human raters for subjective evaluation of TTS using standardized protocols such as MUSHRA. Details are provided in the Appendix F.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[No\]](#)

Justification: We only ran subjective audio evaluations on generation that are low risk, e.g. it contains no unsafe or shocking content. Thus we did not consider it necessary to have overview for those studies.

Guidelines:

- 872 • The answer NA means that the paper does not involve crowdsourcing nor research with  
873 human subjects.
- 874 • Depending on the country in which research is conducted, IRB approval (or equivalent)  
875 may be required for any human subjects research. If you obtained IRB approval, you  
876 should clearly state this in the paper.
- 877 • We recognize that the procedures for this may vary significantly between institutions  
878 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the  
879 guidelines for their institution.
- 880 • For initial submissions, do not include any information that would break anonymity (if  
881 applicable), such as the institution conducting the review.

## 882 16. Declaration of LLM usage

883 Question: Does the paper describe the usage of LLMs if it is an important, original, or  
884 non-standard component of the core methods in this research? Note that if the LLM is used  
885 only for writing, editing, or formatting purposes and does not impact the core methodology,  
886 scientific rigorousness, or originality of the research, declaration is not required.

887 Answer: [Yes]

888 Justification: we describe how a LLM is used to generate synthetic dialogs for evaluation of  
889 our models in Section 4.4, and Appendix B.

890 Guidelines:

- 891 • The answer NA means that the core method development in this research does not  
892 involve LLMs as any important, original, or non-standard components.
- 893 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)  
894 for what should or should not be described.

## A Training Data

### A.1 Short-form finetuning data

This collection includes: LibriSpeech (Panayotov et al., 2015), VoxPopuli (Wang et al., 2021), GigaSpeech (Chen et al., 2021), AMI Carletta (2007); Renals et al. (2007), SPGISpeech (O’Neill et al., 2021), TED-LIUM (Hernandez et al., 2018), and Earnings22 (Del Rio et al., 2022). We filter out examples where Whisper had a high character error rate. With LibriSpeech (LS), we use timestamps provided by Montreal Forced Aligner (McAuliffe et al., 2025), as prepared by Lugosch et al. (2019). For the rest of the datasets, we use timestamps provided by Whisper (see Section 3.2). Importantly, at this stage, examples are short: only 0.5% of examples are longer than 30.5s. This data mixture totals 28k hours.

Table 6 summarizes licenses for the datasets used at this stage.

Dataset	License
AMI	CC-BY-4.0
EARNINGS22	CC-ShareAlike-4.0
GigaSpeech	Apache 2.0
LibriSpeech	CC-BY-4.0
SPGISpeech	Non-commercial use
TED-LIUM	BY-NC-ND 3.0
VoxPopuli	Creative Commons Zero

Table 6: Licenses for the public datasets we used for DSM-ASR finetuning.

### A.2 Long-form finetuning data

First, we concatenate utterances in LibriSpeech (Panayotov et al., 2015) to form segments of up to 6 minutes long ( $\approx$  1k hours). Second, we use a collection of synthesized dialogs, each dialog lasting for 5 minutes ( $\approx$  22k hours). Those examples are produced by a preliminary version of DSM-TTS. We combine these datasets with the short-form finetuning dataset in a weighted mixture, with respective weights 8:1:1 (i.e., a sample from LibriSpeech is 8x more likely to be included in a batch than from the short-form data mixture).

## B Evaluation TTS Data

For evaluating our data, we use both monologue and dialog scripts. We do not have reference audio to compare to, although we can still compare models to one another. We will release this dataset along with the inference and evaluation code used for all baselines and our own model.

**Monologues.** Those are taken from news articles of the NTREX-128 dataset (Federmann et al., 2022), giving us 123 articles per language. Note that those articles are already split in sentences, which we leverage when evaluating models that do not support long form generation.

**Dialogs.** Dialogs are generated using the `mistral-large-latest` model through their API<sup>3</sup>. The scripts fall in 3 categories: daily life, technical, and number heavy. For daily life scripts, the LLM API is first asked to come up with a number of situations that could arise in daily interactions. For each one, it is then tasked with generating a script in a given language and with a target number of turns. We try to generate 50 daily dialogs per language (due to failures in the generation, only 97 in total are available). For technical topics, we follow a similar approach, except the LLM API is given a specific topic (technical, but also person, piece of art, movie, etc.) to discuss instead. We again have 50 technical dialogs per language. Finally, for number heavy dialogs, we use 12 hand written prompts, such as “*PERSON\_A and PERSON\_B discuss the chronology of various events during the middle age*”, which we use for generating scripts in both English and French. In total we thus have nearly 112 dialogs per language. Examples are provided in Figures 5 and 6.

<sup>3</sup><https://docs.mistral.ai/api/>

Table 7: **Precision of predicted timestamps.** F1 and mIoU metrics, calculated on LS test-clean.

METRIC	CRISPERWHISPER	WHISPER LARGE-V3	DSM-ASR
F1	0.85	0.22	0.73
mIoU	0.65	0.30	0.54

**Voices.** For voice conditioning, we use samples from the test set of VCTK (Yamagishi et al., 2019) (ODC-BY license) for English, keeping only one utterance per speaker whose duration is longer than 7 seconds, and less than 10 seconds. We kept 50 voices. For French, we combine speakers from the test and valid sets of CML (Oliveira et al., 2023) (CC-BY 4.0), applying the same criterion. This gives us 35 voices. Pairs of voices are randomly assigned to each script, although the mapping is fixed across evaluation runs.

## C Timestamp predictions

We adopt evaluation metrics from Wagner et al. (2024). Firstly, we compute an F1 score by defining true positives as words that match a reference word and overlap with it temporarily; false positive words are predicted but either do not match or do not overlap with a reference word; and false negatives are the reference words that do not have a match in content or have no temporal overlap with predicted words. The temporal overlap happens if the predicted timestamps for the word start and word end fall within a collar distance from the true events. Since this F1 metric considers overlap in a binary way, Wagner et al. (2024) complement it with a more nuanced measure, mean Intersection over Union (mIoU), which additionally accounts for the relative duration of the temporal overlap. As the evaluation dataset, use the same time-aligned LibriSpeech test-clean as for latency measurement. Results are shown in Table 7.

## D DSM-ASR: speech representation

One natural question is whether DSM-ASR is hindered by using discretized speech representation. In order to answer this question, we use the fact that the models are trained using quantizer dropout: at training time, after a randomly selected cut-off level, all quantization levels are zeroed out (Zeghidour et al., 2022; Défossez et al., 2024). We exploit this by running a series of evaluations of the same model on LibriSpeech test-clean (Panayotov et al., 2015), while systematically dropping quantizers at different cut-off levels. We observe that having less than 24 quantizers comes at a cost on WER; however, having more than that is not beneficial. Thus we conclude that having even more nuanced representation (e.g., continuous embeddings), might not lead to extra gains.

## E TTS Baseline evaluations

We present how the baselines presented in Section 4.4 were evaluated. All baselines are given the same audio conditioning extracts of no more than 10 sec. for each speaker.

**ElevenLab.** For monologues, we directly feed the entire script. For dialogs, we feed turns one by one, with no context (except for the speaker conditioning).

**Dia.** Dia was only trained to generate short audio extracts. For dialogs, we provide the turns used for speaker conditioning, and the last turn that was generated (e.g. from the other speaker than we are currently generating). We take care to provide them in such an order that speaker alternate. Even with limited context, generation sometimes failed, in which case we revert to generating the segment with no context (always keeping the speaker conditioning).

**Orpheus.** For monologues, we generate the sentences one by one with a context given by the speaker conditioning audio sample, along with one sentence of context. When that fails, we only give the audio conditioning sample. For dialogs, as Orpheus is single speaker, we generate the turns one by one, only with the speaker conditioning as context.

**CSM.** CSM normally only supports dialogs. For monologues, we have to repeat twice the same speaker in a row, which is out of distribution. We experimented with various parameters (deduplicating

the speaker conditioning to pretend they are two speakers, different context size) and found the best results in terms of WER was obtained by keeping a single speaker conditioning, providing no extra past context, and still having two segments in a row with the same speaker id. For dialogs, we start with giving the turns used for speaker conditioning. Then we experimented with giving no context (short form), or giving up to 6 turns of context (long form). In case where the context would be too long and CSM raised an error, we would try again with a shorter context.

## F Human evaluations

### F.1 Models and dataset

The model are evaluated using the same parameters as described in Section E. For CSM, we only evaluate the version with no context which was giving the best WER. We use a subset of the data provided in Section B, namely 15 monologues, and 9 dialogs (3 from each category) per language. For each script, we rate both the first 30 seconds, and the last 30 seconds separately.

### F.2 Subjective evaluation tasks

Raters were payed on average 9.3£ per hour.

**Quality.** We also organize a MUSHRA style study for evaluating the audio quality, although with no explicit reference. For each script, the rater is presented with all the models’ generations, and must note each one on a scale from 1 to 100. In particular, the instruction is as follows: “*How would you rate the overall quality of these audio clips? Consider aspects such as clarity, balance, richness, and naturalness*”. We report the aggregated ratings, with the confidence intervals being the plus and minus the standard deviation of the mean estimator. For each language, 800 scripts were rated (each time covering all methods).

**Speaker similarity.** We present the rater with the audio used for speaker conditioning (either single speaker or the concatenation of both speakers for dialogs), along with two generations from two random models given the same script. The rater must choose which extract has speakers sound the most like the reference audio. The instruction is as follow: “*The reference contains the voices of one or two speakers. The audio files below it contain either a monologue or a dialogue. Which voices sound more like the speakers in the reference?*” The win-rate is summarized as an *Elo score*, as described in the next paragraph. 1600 pairs were rated per language.

**Bayesian Elo Score.** The Elo score allows the ranking of models based on some pairwise comparisons of audio samples. Given two models  $A$  and  $B$ , the probability that  $A$  is preferred over  $B$  is:

$$P(A > B) = \frac{1}{1 + 10^{(E_B - E_A)/400}}, \quad (8)$$

where  $E_A$  and  $E_B$  are the Elo scores of each model. Unlike a traditional Elo score, the Bayesian Elo score uses a Gamma prior, so that one can derive confidence intervals over the posterior distribution.

We denote  $S_A = 10^{(E_A - c)/400}$ , with  $c$  freely chosen in  $\mathbb{R}$ . Then eq. (8) becomes

$$P(A > B) = \frac{S_A}{S_A + S_B}, \quad (9)$$

which is a Bradley-Terry (Bradley & Terry, 1952) model. We use the iterative by Caron & Doucet (2010) where  $S_A^0$  follows a Gamma prior with parameters  $\alpha^0, \beta^0$ . By denoting  $w_A$  the number of times where method  $A$  won against any other methods and  $w_{AB}$  the number of times where  $A$  and  $B$  are compared,  $S_A^t$  is computed with the following update rule:

$$S_A^{t+1} = \frac{\alpha^0 + w_A}{\beta^0 + \sum_{B \neq A} \frac{w_{AB}}{S_A^t + S_B^t}}, \quad (10)$$

given as the mean of the Gamma distribution with updated parameters  $\alpha_A^{t+1}, \beta_A^{t+1}$  given by

$$\alpha_A^{t+1} = \alpha^0 + w_A, \quad \text{and} \quad \beta_A^{t+1} = \beta^0 + \sum_{B \neq A} \frac{w_{A,B}}{S_A^t + S_B^t}. \quad (11)$$

Iterating over  $t$  allows reaching a fix point, we run 30 of them, once we have collected all the pairs. We use  $\alpha = 0.1, \beta = 0.1, c = 2000$  so that, in absence of any data,  $S_A = 1$  and  $E_A = 2000$ . Confidence intervals are 95% confidence interval according to the posterior.



## 1014 G Latency, RTF, and throughput

1015 We report in Table 5 latency, RTF, and throughput for various batch sizes for our model. Those  
1016 numbers are obtained under real workload, with no synchronization between the batch elements, that  
1017 is, requests for TTS arrive at the service at *random times*.

1018 **Batch size of 1.** The latency represents the time to first audio, including all computations. Given  
1019 that we use a delay  $\tau = 2$  sec., or 25 steps, we need to account for the time that it takes to run those.  
1020 We have to account for two times:  $d_b$  the duration it takes to run a single step of the backbone, and  $d_q$   
1021 the time it takes to run the small Transformer over the  $Q$  dimensions. For the first 25 steps, we need  
1022 only to run the backbone transformer, as no audio is produced. When the batch size is 1, the time  
1023 to first token is given by  $\tau f_r \cdot d_b + 2 \cdot (d_b + d_q)$ . The extra  $2 \cdot (d_b + d_q)$  comes from the fact that  
1024 following Défossez et al. (2024), we use an *acoustic delay* of 2 steps between the first codebook and  
1025 the remaining ones. With values of  $d_b = 5$  ms, and  $d_q = 25$  ms, we obtained the given results. The  
1026 throughput and RTF are derived using only the time per step of  $d_b + d_q$ . Note that even though the  
1027 small Transformer over  $Q$  is smaller in terms of dimensions and layers per codebook, it represents a  
1028 significant amount of computation due to us using  $Q = 32$  codebooks, leading to a large number of  
1029 kernels being scheduled.

1030 **Interleaved steps for higher batch sizes.** When needing to handle requests that are not in sync,  
1031 with a larger batch size, we cannot just skip the small Transformer over  $Q$ , as not everyone will be  
1032 at the same point in the generation. We use a 2-for-1 interleaving pattern, where if any of the batch  
1033 items is in the initial phase, we interleave two steps of backbone only, followed by a full step. For  
1034 items in the initial phase, this gives an effective time per step of  $(d_b + 2 \cdot d_q)/3$ .

## 1035 H TTS results on LibriSpeech training on public datasets

1036 We experiment with training DSM-TTS using only the publicly available datasets used for the  
1037 short-form fine tuning of the ASR model, as described in Section A.

1038 **Architecture and training.** This variant of the model uses a 300m backbone (24 layers with  
1039 dimension 1024),  $Q = 16$  codebooks. The small Transformer over  $Q$  is made more compact  
1040 following (Labiausse et al., 2025), e.g. sharing weights for all codebooks from  $[9 - 16]$ ,  $[17, 24]$ ,  
1041 along with using only 4 layers per step. We use a delay of  $\tau = 1.28$  sec., or 16 steps. It receives no  
1042 speaker conditioning, instead it has a probability of 20% of seeing the start of the audio with no text,  
1043 and a probability of 10% of having this prefix completely empty, which will be used for applying  
1044 classifier-free-guidance (Ho & Salimans, 2022; Kreuk et al., 2023). The model receives no text 20%  
1045 of the time. The model is trained for 500k updates with a batch size of 64 on 16 H100 GPUs, using  
1046 the same optimization parameters as described in Section 4.1.

1047 **Evaluations** We evaluate our approach on LibriSpeech test clean, with punctuation, following the  
1048 same protocol as Chen et al. (2024) (F5-TTS). When evaluating our model, we provide the speaker  
1049 conditioning sample as a prefix. We use classifier-free-guidance with a coefficient of 2., where  
1050 the reference logits are taken evaluating the model with no audio prefix, and with no text tokens,  
1051 which improves the word error rate. We use the exact same evaluation set as F5-TTS, along with the  
1052 same text normalization, ASR model (whisper-large-v3), and reuse their code for evaluating the  
1053 speaker similarity to the original conditioning audio.

1054 **Results** Results are provided in Table 8. We achieve a large improvement of 0.3% of WER over the  
1055 best F5-TTS model, however, with a small decrease in speaker similarity. While the diffusion based  
1056 approach F5-TTS can generate faster a single script, it suffers from a higher latency, due to its non  
1057 causal nature requiring the entire audio to be generated at once. Finally, we again show that the ease  
1058 of batching of our methods allows for more than 100x throughput (duration of audio generated per  
1059 second of computation) even for requests arriving unsynchronized, as explained in Section G.

Table 8: **Results of DSM-TTS on LibriSpeech test-clean.** We compare to F5-TTS (Chen et al., 2024), following their evaluation framework. RTF and Throughput were computed on a H100 for both methods.

Model	#Param.	#Data	WER (%) ↓	SIM-o ↑	Latency ↑	RTF ↑	Throughput ↑
F5-TTS (16 flow steps)	336M	100K (EN, ZH)	2.53	<b>0.66</b>	~ 800 ms	<b>14.2</b>	14.2
F5-TTS (32 flow steps)			2.42	<b>0.66</b>	~ 400 ms	7.1	7.1
DSM-TTS (ours, b.s.=1)	750M	88K (EN)	<b>2.12</b>	0.56	<b>125ms</b>	4.4	4.4
DSM-TTS (ours, b.s.=32)			<b>2.12</b>	0.56	245ms	3.5	<b>111.8</b>

#### Dialog example, daily life

[LIU]: Hey Jean, how's it going with the studies?  
[JEAN]: Not bad, Liu. Just trying to keep up with everything.  
[LIU]: I hear you. Have you tried any study apps lately?  
[JEAN]: Not really. I've been sticking to the old pen and paper method.  
[LIU]: Oh, you should try this new app. It's really helpful.  
[JEAN]: What's it called?  
[LIU]: It's called "StudyBuddy." It helps you organize your notes and has flashcards too.  
[JEAN]: Sounds interesting. How does it work?  
[LIU]: You can upload your notes, create flashcards, and it even has a quiz feature.  
[JEAN]: That sounds really useful. I'll give it a try.  
[LIU]: Great! I think you'll find it really helpful.  
[JEAN]: Alright, I'll download it tonight.  
[LIU]: Awesome. Let me know how it goes.  
[JEAN]: Will do. Thanks for the recommendation, Liu.  
[LIU]: No problem, anytime.  
[JEAN]: Hey, I just downloaded the app. It looks pretty good.  
[LIU]: Glad to hear it. Have you tried any of the features yet?  
[JEAN]: I uploaded some notes and made a few flashcards. It's really user-friendly.  
[LIU]: That's great to hear. I find the quiz feature really helpful too.  
[JEAN]: Yeah, I'm going to try that next. Thanks again, Liu.  
[LIU]: You're welcome, Jean. Good luck with your studies!  
[JEAN]: Thanks! Talk to you later.  
[LIU]: Take care, Jean. See you in class!

Figure 5: Example of dialog generated with an LLM to serve as evaluation data. This dialog covers daily life topics.

#### Dialog example, daily life

[LUIS]: Mrs. Al-Falasi, have you heard of hydroxyapatite before?  
[MRS AL-FALASI]: Yes, Luis. I know it's related to bones and teeth, but that's about it.  
[LUIS]: Great start! Hydroxyapatite is a naturally occurring mineral form of calcium apatite. It has the formula  $\text{Ca}_5(\text{PO}_4)_3(\text{OH})$ .  
[MRS AL-FALASI]: That sounds quite complex. What makes it so important?  
[LUIS]: It's the main mineral component of bones and teeth. It provides rigidity and strength.  
[MRS AL-FALASI]: That makes sense. Is it used in any medical applications?  
[LUIS]: Yes, synthetic hydroxyapatite is used in medical and dental applications for bone grafts and implants.  
[MRS AL-FALASI]: Really? How does it work in those contexts?  
[LUIS]: It has biocompatible properties, which means it's suitable for use in body tissue repair.  
[MRS AL-FALASI]: That's fascinating. Are there any other uses for it?  
[LUIS]: Absolutely. Hydroxyapatite can be used in coatings on orthopedic implants to promote bone growth.  
[MRS AL-FALASI]: I didn't know that. It seems very versatile.  
[LUIS]: It is. It's also used in toothpaste and mouthwash to help remineralize enamel.  
[MRS AL-FALASI]: That explains why it's so beneficial for dental health. Anything else I should know?  
[LUIS]: The material is effective in drug delivery systems, especially for bone diseases.  
[MRS AL-FALASI]: That's impressive. Are there any non-medical uses?  
[LUIS]: Yes, hydroxyapatite is utilized in chromatography for protein purification.  
[MRS AL-FALASI]: Interesting. How about its sources? Where does it come from?  
[LUIS]: It can be derived from natural sources such as animal bones or synthesized chemically.  
[MRS AL-FALASI]: That's good to know. Any other applications I might not be aware of?  
[LUIS]: Its porous structure makes it useful in filtering and ion-exchange applications.  
[MRS AL-FALASI]: That's a lot of information. Thanks for explaining, Luis.  
[LUIS]: You're welcome, Mrs. Al-Falasi. Hydroxyapatite is truly a remarkable material.

Figure 6: Example of dialog generated with an LLM to serve as evaluation data. This dialog covers technical topics.