# Unbounded Length and Streamable Text-to-Speech with Prefixed Sliding Window Attention

**Théodor Lemerle***
STMS Lab
IRCAM, CNRS, Sorbonne Université
Paris, France
theodorlemerle@gmail.com

**Diego Torres**
STMS Lab
IRCAM, CNRS, Sorbonne Université
Paris, France
dtorres@ircam.fr

**Téo Guichoux**
STMS Lab
IRCAM, CNRS, Sorbonne Université
ISIR, Sorbonne Université
Paris, France
guichoux@ircam.fr

**Nicolas Obin**
STMS Lab
IRCAM, CNRS, Sorbonne Université
Paris, France
obin@ircam.fr

**Axel Roebel**
STMS Lab
IRCAM, CNRS, Sorbonne Université
Paris, France
roebel@ircam.fr

## Abstract

Existing text-to-speech (TTS) systems achieve high quality, yet streaming and long-form generation remain challenging. We propose simple adaptations to a standard encoder–decoder Transformer TTS architecture that enable both capabilities without limits on total duration. We first observe that attention patterns in these models are well organized, following a structure that motivates a specific windowing strategy: a prefixed sliding window for autoregressive decoding and a sliding cross-attention for text conditioning. Together, we show that these adaptations address length generalization and error accumulation while being fully streamable in text and audio. Trained exclusively on segments shorter than 30 seconds, our system produces seamless, consistent speech at practically unbounded lengths, outperforming baselines trained with full-context attention. We validate the approach through continuous synthesis lasting several hours and demonstrate competitive results against state-of-the-art open-source baselines on both short- and long-form benchmarks.

## 1 Introduction

Recent advances in large language models have enabled coherent text generation over long contexts, fueling interactive applications such as conversational agents, audiobooks, and embodied AI [Naveed et al., 2025]. These settings place new demands on text-to-speech (TTS) systems, which must produce long, coherent speech while responding incrementally to evolving textual input.

*Corresponding author.
Audio samples: https://psw-speech.github.io/
Code: https://github.com/theodorblackbird/psw-speech

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Long-form TTS introduces challenges beyond short-utterance synthesis. Models must maintain prosodic consistency and speaker identity over extended durations, yet they are typically trained and evaluated on short segments and dramatically fail to generalize in length beyond their training horizon [Battenberg et al., 2025]. Simply enlarging the training context is impractical: standard Transformer attention scales poorly with sequence length, and long, coherent speech recordings are scarce and expensive to collect. The difficulty is compounded by error accumulation [Pasini et al., 2024]: long synthesis suffers from instabilities that manifest, for instance, as speaker identity drift [Gong et al., 2026, Peng et al., 2025b].

Practical deployment further requires *streamability*—incremental text consumption and audio generation—which is poorly supported by Transformer architectures relying on full-context attention. As a result, even models capable of long-form synthesis struggle to scale efficiently or run in real time.

Starting from a standard encoder–decoder TTS architecture and the empirical observation that (i) cross-attention follows a near-monotonic path, concentrating on a short text window around the current speech position, and (ii) self-attention in both encoder and decoder places most of its mass on nearby tokens, we propose PSW-Speech, a principled windowing approach that aligns attention with the local structure exhibited by encoder–decoder TTS models. These adaptations address length generalization and error accumulation, enabling stable synthesis on hour-long inputs despite training solely on short excerpts. PSW-Speech matches the synthesis quality of its global-attention counterpart on short-form synthesis and outperforms substantially larger baselines on long-form synthesis, while being fully streamable in text and audio. To support our evaluation, we introduce Very-Long LibriSpeech, a benchmark inspired by LibriSpeech-Long [Park et al., 2025] that assesses long-form synthesis quality at hour-long durations.

## 2 Related Works

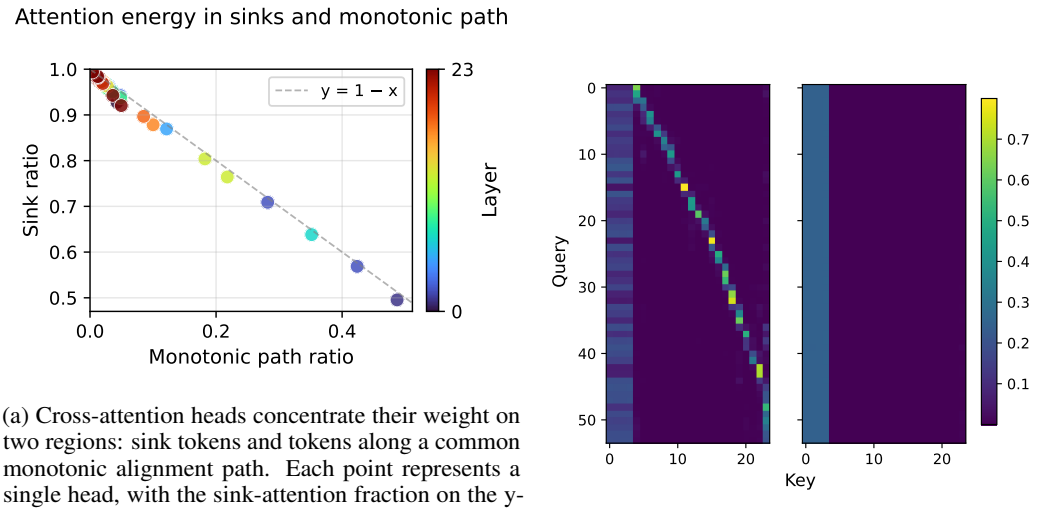
### 2.1 Attention and Length Generalization in TTS

Cross-attention [Vaswani et al., 2017] plays a central role in Transformer-based TTS models by enabling flexible, data-driven alignment between text and speech, which contributes to natural prosody and expressive timing. However, its unconstrained and global formulation provides little inductive bias toward monotonic progression, making alignment unstable and limiting generalization to long-form or streaming synthesis [Battenberg et al., 2020]. A substantial body of work in the Tacotron 2 [Shen et al., 2018] literature has focused on shaping cross-attention through additional inductive biases, such as location-sensitive attention [Shen et al., 2018], monotonic constraints [He et al., 2019], and guided alignment losses [Tachibana et al., 2018], to improve robustness and stability during decoding. Similar concerns have resurfaced in the context of LLM-based and decoder-only TTS, where monotonic alignment priors [Neekhara et al., 2024, Han et al., 2024], attention-constrained inference [Chen et al., 2024], and stepwise monotonic attention [Zhang et al., 2025] have been proposed to mitigate the hallucinations, repetitions, and skips that arise from unconstrained cross-attention.

A complementary line of work targets length generalization through architectural modifications to the attention mechanism itself. VoiceStar [Peng et al., 2025a] introduces PM-RoPE, a cross-attention–specific extension of rotary positional embeddings, enabling duration control and moderate length generalization (typically $1.5$–$2\times$ the training horizon), though synthesis quality still degrades as output length increases. Very Attentive Tacotron [Battenberg et al., 2025] proposes a learned relative positional bias for cross-attention in an encoder–decoder Transformer, achieving stable synthesis up to 80 s despite training on 10 s excerpts. While promising, this approach retains the quadratic complexity of global attention, is not streamable, and relies on a recurrent cross-attention bias computation that limits its scalability.

### 2.2 Text Streaming TTS

Interleaved TTS models such as CosyVoice [Du et al., 2024] and SpeakStream [Bai et al., 2025] support streaming by interleaving text and audio tokens in a single sequence, allowing incremental text consumption at inference. This formulation, however, breaks the parallelization of standard Transformers—since audio tokens are generated causally, text encoding can no longer run in parallel—and offers no natural length extrapolation, as the joint sequence remains bounded by training-




(a) Cross-attention heads concentrate their weight on two regions: sink tokens and tokens along a common monotonic alignment path. Each point represents a single head, with the sink-attention fraction on the y-axis and the monotonic-path attention fraction on the x-axis. Points cluster along the line $x + y = 1$, indicating that together these two components account for nearly all of the attention mass.

(b) Two representative heads: monotonic-dominated (left) tracks the diagonal alignment; sink-dominated (right) collapses onto the first four encoder (sink) tokens.

Figure 1: Cross-attention heads split into sink- and monotonic-dominated regimes.

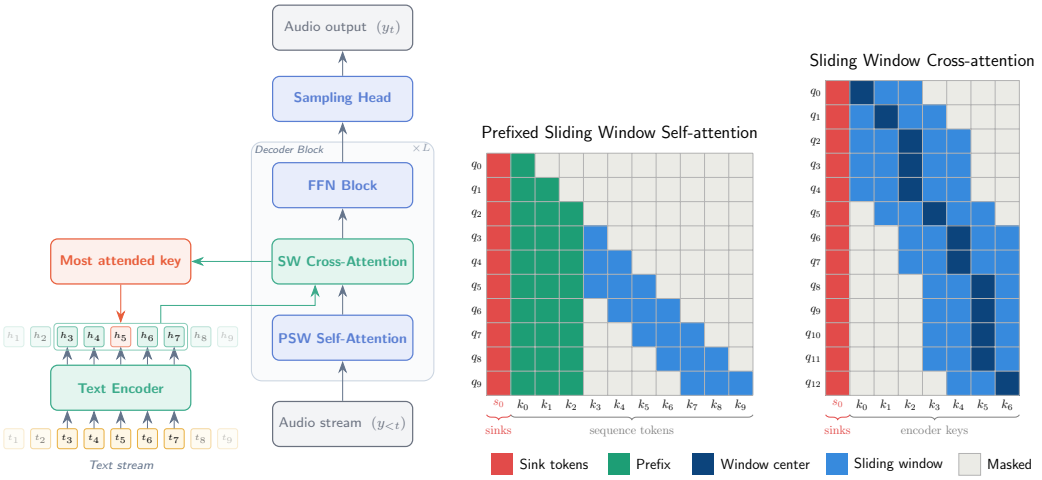
time lengths. Long-form synthesis in this paradigm therefore typically relies on sequence-length curriculum learning [Peng et al., 2025b] or synthetic distillation [Zeghidour et al., 2025], both of which require long recordings or expensive teacher pipelines.

Delayed Streams Modeling (DSM) [Zeghidour et al., 2025] reformulates streaming sequence-to-sequence learning as decoder-only language modeling by pre-aligning multiple token streams with fixed delays, enabling low-latency inference and competitive ASR/TTS performance within a unified framework. DSM nevertheless remains sensitive to training sequence length and does not generalize to long-form generation, and its reliance on external alignment tools limits reproducibility in low-resource settings.

## 3 Attention in Encoder–Decoder TTS

In a standard transformer encoder–decoder TTS [Hussain et al., 2025, Peng et al., 2025a], which comprises a text encoder and an autoregressive decoder, the interaction between text and speech is mediated through cross-attention. Following prior work on streaming Transformers for text generation [Xiao et al., 2024], which showed that *sink tokens* help length generalization, we prepend learned sink tokens to the key and value sequences. These input-independent tokens are visible to every query and provide a global fallback for softmax-based attention; in cross-attention TTS specifically, they also improve interpretability by allowing the model to explicitly *unattend* the text.

In this context, we analyze cross-attention in light of the optimal monotonic path provided by Monotonic Alignment Search (MAS) [Kim et al., 2020], computed on the sum of all cross-attention weights, with sink tokens discarded after the softmax. We observe that most of the behavior can be explained by two patterns. A subset of the heads exhibits strongly monotonic behavior, where attention mass progresses steadily from left to right along the text sequence (Figure 1b left). On the other hand, heads which do not exhibit monotonicity tend to follow a sink-dominated behavior, resulting in nearly constant, input-independent contributions (Figure 1b right). A complementary analysis of self-attention in both encoder and decoder (see Section A) reveals an analogous empirical locality: attention mass concentrates overwhelmingly at short relative distances, with distant tokens being confined to low levels of attention. Taken together, these observations suggest that the global receptive field of attention might be under-exploited in practice, and motivate an architecture in which both cross- and self-attention are explicitly restricted to sliding windows.




(a) Architecture (b) Attention masks

Figure 2: **Overview of PSW-Speech. (a)** A text stream $\mathbf{x}$ is consumed by a text encoder producing hidden states $\mathbf{h}$. An autoregressive decoder of $L$ blocks generates audio tokens $y_t$ from past audio $y_{<t}$, attending to text via sliding-window (SW) cross-attention and to its own past via prefixed sliding-window (PSW) self-attention. The most attended encoder key recenters the cross-attention window at the next step. **(b)** Decoder attention masks. *Left:* PSW self-attention — each query $q_i$ attends to a sink token $s$ (red), a prefix (green), and a local window over the most recent positions (blue); other positions are masked. *Right:* SW cross-attention — each query attends to a sink and a local window of encoder keys centered on the previously most-attended text token.

## 4 Proposed Method

The observations of the previous section suggest that, despite operating with global context, cross-attention in encoder–decoder TTS is effectively driven by a local, monotonically advancing region of the text. We therefore propose PSW-Speech, a streamable and length-generalizable TTS model based on two sliding windows: one over the text, for cross-attention, and one over past speech tokens, for decoder self-attention. The decoder window is advanced by one position at each decoding step. The cross-attention window, in contrast, progresses non-uniformly, following the flow of generated speech, and must be moved accordingly. We find that a simple heuristic—centering the next window on the most attended text token—is sufficient in practice (Algorithm 1). The tracked text position also serves as an indicator of reading progress: rather than predicting termination with a dedicated head, we stop generation with a heuristic condition on the history of such positions. To anchor speaker identity over long generations, we additionally allow the first tokens of the speech sequence—corresponding to the speaker prompt—to remain visible to every query. We find that the combination of a sliding window with this permanent prefix is instrumental in preventing the accumulation of errors (see Figure 5). Consequently, training reduces to masking operations: a fixed-shape causal sliding mask on the decoder, and a cross-attention mask derived from alignments produced by a pre-trained TTS model via dynamic programming (MAS) [Kim et al., 2020].

### 4.1 Jittered Cross-Attention Windowing

Centering the cross-attention window on the position obtained by teacher forcing leaks information about the attended text token, encouraging the model to rely on the window center rather than actively attending to text features. We prevent this leak by *jittering* the mask during training: a uniform per-row random shift is applied to the window center at each decoder step (Figure 3). This has a substantial effect on intelligibility in our ablations (Table 2).

**Algorithm 1** PSW-Speech Inference

**Require:** Pre-trained PSW-Speech model $\mathcal{M}$
**Require:** Text hidden states $\mathbf{h} = (h_1, \ldots, h_N)$
**Require:** Cross-attention window radius $R$, decoder window size $W$, prefix length $P$
**Require:** Stopping condition STOP, maximum length $T_{\max}$
**Ensure:** Speech tokens $\mathbf{y}$

$\phi_0 \leftarrow 0$ ▷ *Initialize text window center*
**for** $t = 1$ **to** $T_{\max}$ **do**
  $\mathcal{W}_t^{\mathrm{x}} \leftarrow [\![\phi_{t-1} - R,\ \phi_{t-1} + R]\!]$ ▷ *Cross-attention window*
  $\mathcal{W}_t^{\mathrm{d}} \leftarrow [\![1, P]\!] \cup [\![t - W,\ t - 1]\!]$ ▷ *Prefixed self-attention window*
  $(y_t, \boldsymbol{\alpha}_t) \sim \mathcal{M}(\cdot \mid \mathbf{y}_{\mathcal{W}_t^{\mathrm{d}}},\ \mathbf{h}_{\mathcal{W}_t^{\mathrm{x}}})$ ▷ *Sample token & collect cross-attention weights*
  $\phi_t \leftarrow \arg\max_{j \in \mathcal{W}_t^{\mathrm{x}}} \alpha_{t,j}$ ▷ *Retrieve most attended text position*
  **if** $\textsc{Stop}(\phi_{1:t}, N)$ **then break** ▷ *Stopping condition*
**end for**
**return** $\mathbf{y}$

### 4.2 A TTS Model for Alignment of Text and Audio

As shown in Figure 8, alignments provided by a small encoder–decoder model closely match those of a larger one. We therefore use a small encoder–decoder model as an alignment provider. Once trained, the aligner's cross-attention weights are processed with Monotonic Alignment Search (MAS) [Kim et al., 2020] to obtain, for each speech frame of our dataset, its corresponding text position. This position defines the center of the cross-attention window used for training PSW-Speech.

### 4.3 Text Stream Encoding

We adopt a bidirectional self-attention text encoder equipped with RoPE [Su et al., 2024]. Since RoPE does not extrapolate reliably beyond training lengths, we avoid encoding the entire text sequence at once. Instead, we perform *blockwise packet encoding*: the input is partitioned into variable-size packets, each encoded independently, and the resulting representations are concatenated along the time dimension. During training this boils down to *block diagonal* masking with variable block size.

Under this design, the packet size directly defines the maximum amount of future textual context available to the decoder—what we refer to as the *text look-ahead*. A notable consequence is that look-ahead becomes a free parameter at inference time, rather than being fixed at training as in prior work [Zeghidour et al., 2025, Du et al., 2024, Torgashov et al., 2025]. To avoid strictly zero look-ahead at packet boundaries, we prepend three depthwise convolutional layers before the Transformer encoder. The effect of packet size on objective metrics is reported in Table 1c.

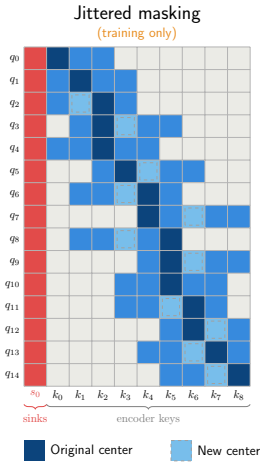



Figure 3: Jittered cross-attention mask.

### 4.4 Continuous Speech Tokenizer and AR-Flow Matching

We use a continuous speech tokenizer trained following the recipe of Lemerle et al. [2026] on a speaker-balanced 3k-hour subset of HiFiTTS-2 together with the LibriTTS training split, at 24 kHz. It encodes speech into 32-dimensional continuous latents at a frame rate of 12.5 Hz, which serve as the autoregressive modeling targets of PSW-Speech. The sampling head is a 4-layer MLP following the design of Li et al. [2024], in line with autoregressive diffusion-based TTS systems such as VibeVoice [Peng et al., 2025b] and CALM [Rouard et al., 2025]. It is conditioned on the output of the autoregressive decoder and trained with a Conditional Flow-Matching [Lipman et al., 2022] objective, regressing the velocity field $\mathbf{v}$ of a linear interpolant from Gaussian noise to the ground-truth latent.

# 5 Training and Inference Details

## 5.1 Data

We train our models on a combination of two English speech datasets: HiFiTTS-2 [Langman et al., 2025], derived from LibriVox recordings, and the English subset of Emilia [He et al., 2024]. All audio is resampled to 24 kHz, and transcripts are lower-cased and tokenized with a 512-entry BPE tokenizer trained on the transcript corpus.

## 5.2 Training Setup

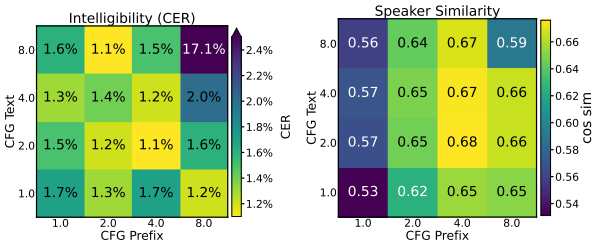
Following our methodology, we train two distinct TTS models. Detailed hyperparameters and training setup are described in Section C.

**Alignment provider encoder–decoder TTS.** A smaller version of the encoder–decoder without windowing with approximately 50M parameters, used exclusively to pre-compute cross-attention alignments under teacher forcing. Training requires approximately 6 hours on a single RTX 4090 GPU.

**PSW-Speech.** A 330M parameter model trained using sliding-window cross-attention masking derived from the alignments produced by the alignment provider. Training is performed for 2 days on two RTX 4090 GPUs. We sample the text packet size in tokens from a log-uniform distribution over $[\![16, 256]\!]$ and the window radius $R$ uniformly in $[\![1, 4]\!]$. Prefix lengths are uniformly drawn in $[\![48, 128]\!]$ and dropped at a rate $0.1$. Decoder sliding-window lengths $W$ are uniformly drawn in $[\![4, 128]\!]$ and dropped at a rate $0.1$ (*i.e.*, full causal masking). We only use cross-attention in decoder layers 4 to 7 (inclusive), which saves 70M parameters compared to cross-attention at every layer, without impacting objective metrics (Table 2). We prepend 4 learned sink tokens to cross-attention and decoder key-values.

## 5.3 Sampling Guidance

We employ sampling guidance during inference using Classifier-Free Guidance (CFG) [Ho and Salimans, 2021]. We distinguish three complementary forms of guidance.




(a) Intelligibility (b) Speaker Similarity

Figure 4: Effect of prefix and text guidance scales on objective metrics.

**Sampling-Head Guidance.** Similarly to prior AR continuous TTS models [Li et al., 2024, Peng et al., 2025b, Jia et al., 2025], we apply guidance at the sampling head level in order to encourage consistency with the autoregressive backbone prediction, which has been shown to improve output quality. This requires randomly dropping the AR condition of the sampling head during training (rate 0.1).

**Prefix and Text Guidance.** To encourage adherence to the prefix (resp. the text), we also randomly drop each of them during training (rate 0.1). At inference, additional forward passes with different conditioning subsets enable sampling guidance. We write $\mathbf{v}_\emptyset$ for the fully unconditioned prediction (sampling head drops the AR backbone output), $\mathbf{v}_P$ for the prediction conditioned on the prefix alone (text dropped), $\mathbf{v}_T$ for the prediction conditioned on the text alone (prefix dropped), and $\mathbf{v}_{P,T}$ for the fully conditioned prediction. The fully guided velocity is

$$\tilde{\mathbf{v}} = \mathbf{v}_\emptyset + \alpha\Big[\mathbf{v}_P - \mathbf{v}_\emptyset + \alpha_T\Big(\mathbf{v}_T - \mathbf{v}_P + \alpha_P(\mathbf{v}_{P,T} - \mathbf{v}_T)\Big)\Big]. \tag{1}$$

We found that the prefix and text guidance are mutually beneficial (see Figure 4). During inference, unless otherwise stated, we use $\alpha_P = 2.0$, $\alpha_T = 2.0$, and $\alpha = 1.3$, and scale the Gaussian noise by $0.95$ during sampling with 15 steps of ODE sampler for the sampling head.

# 6 Experiments

## 6.1 Evaluation Metrics

We measure character error rate (CER) with the Whisper-v3 automatic speech recognition model [Bain et al., 2023], speaker similarity (SIM) to the prompt speaker with the WavLM-based metric of the SEED evaluation protocol [Chen et al., 2025], and predicted naturalness with UTMOS [Saeki et al., 2022]. We complement these with subjective naturalness (N-MOS) and speaker similarity (S-MOS) tests, detailed in Section E.5.

## 6.2 Baselines

We compare against recent TTS systems covering long-form, streaming, and length extrapolation.

**VoxCPM** [Zhou et al., 2025]: an autoregressive diffusion TTS supporting up to 5 minutes of synthesis.

**VibeVoice** [Peng et al., 2025b]: a large-scale autoregressive diffusion TTS trained with long-context curriculum learning, supporting theoretically up to 90 minutes.

**VoXtream** [Torgashov et al., 2025]: a fully autoregressive zero-shot streaming TTS enabling real-time synthesis from the first word.

**VoiceStar** [Peng et al., 2025a]: an encoder–decoder transformer TTS with duration control and limited length extrapolation via dedicated positional encoding.

For short-form evaluation, we also include a standard encoder–decoder Transformer with global attention, sharing the architecture and training protocol of PSW-Speech.

## 6.3 Very-Long LibriSpeech Benchmark

Inspired by the LibriSpeech-Long benchmark [Park et al., 2025], which evaluates synthesis on utterances up to four minutes in duration, we introduce a new benchmark designed for hour-long evaluation. Starting from the LibriSpeech test-clean split, we retrieve the corresponding original LibriVox recordings together with their aligned Project Gutenberg books. We provide details in Section E.

# 7 Results

We report: (i) the effect of sliding-window width and text look-ahead, (ii) component ablations, (iii) short-form evaluation on SEED, and (iv) long-form evaluation on Very-Long LibriSpeech. Window, guidance, and ablation studies are run on Very-Long LibriSpeech capped at 10 minutes. Inference speed is reported in Section B, where our approach benefits from linear decoding complexity, unlike baselines whose global attention scales quadratically.

## 7.1 Sliding Window Size and Text Look-Ahead

Table 1: Effect of self-attention window, cross-attention window, and look-ahead (in tokens) on CER and speaker similarity (SIM).

(a) Self-attention window length

| Tok | CER (%) ↓ | SIM ↑ |
|---|---|---|
| 2 | 2.4 | 0.61 |
| 4 | 1.8 | 0.64 |
| 8 | 1.5 | 0.65 |
| 16 | 1.1 | 0.65 |
| 32 | 1.1 | 0.65 |
| 64 | 1.1 | 0.65 |
| 128 | 1.2 | 0.63 |

(b) Cross-attention window radius

| Tok | CER (%) ↓ | SIM ↑ |
|---|---|---|
| 1 | 67.3 | 0.20 |
| 2 | 1.1 | 0.65 |
| 3 | 1.1 | 0.65 |
| 4 | 1.1 | 0.65 |

(c) Text look-ahead

| Tok | CER (%) ↓ | SIM ↑ |
|---|---|---|
| 2 | 1.8 | 0.63 |
| 4 | 1.5 | 0.65 |
| 8 | 1.1 | 0.65 |
| 16 | 1.1 | 0.65 |
| 32 | 1.1 | 0.65 |
| 64 | 1.2 | 0.65 |
| 128 | 1.1 | 0.65 |

We study the window sizes and the text look-ahead set by the encoder packet size. Table 1a reports CER and SIM as the audio window varies from 2 to 128 tokens. Performance improves rapidly up to

16 tokens and then plateaus: below 8, the window is too short and degrades objective metrics; above 64, longer windows give no further benefit and slightly hurt SIM (0.63 at 128 tokens), consistent with the self-attention locality in Section A. For the cross-attention window (Table 1b), a radius of 1 is insufficient and synthesis collapses; increasing it to 2 recovers full performance, with no further gain for larger radii. Table 1c reports the analogous study on text encoder packet size, which controls how much future text the decoder can attend to. Quality improves quickly at very small look-aheads, saturates around 8–32 tokens (≈3–10 words), and remains stable up to 128. During inference, unless otherwise stated, we set the decoder window length to 64 and the cross-attention window radius to 2, and sample the text packet size uniformly in $[\![16, 128]\!]$.

### 7.2 Ablation Study

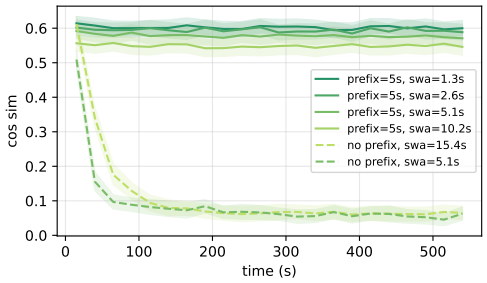



Figure 5: Prefix ablation and decoder window size comparison at inference time, measured by speaker similarity. A short prefix (5s) suffices to stabilize generation.

| Configuration | CER ↓ | SIM ↑ |
|---|---|---|
| PSW-Speech | 1.1% | 0.65 |
| w/o Cross-attention masking | 65.3% | 0.21 |
| w/o Cross-attention jitter | 13.6% | 0.59 |
| w/o Cross-attention sink tokens | 1.1% | 0.65 |
| w/o Decoder sink tokens | 1.1% | 0.65 |
| w/o Prefixed SWA masking | 1.7% | 0.60 |
| Cross-attention at every layer | 1.1% | 0.65 |

Table 2: Ablation of training-time components, measured by character error rate (CER) and speaker similarity (SIM).

Table 2 reports a component-wise ablation. Removing the sliding-window cross-attention mask during training collapses synthesis entirely (CER 65.3%, SIM 0.21): without an explicit prior, the model fails to track the text, showing that the monotonic structure from Section 3 must be *enforced* during training rather than exploited only at inference. Jittering the mask center is also critical for intelligibility (CER 13.6% without it), preventing the model from using the center as a positional cue instead of the underlying text features. Prefixed sliding-window masking at training time improves both metrics, which we attribute to reduced train/inference discrepancy. Removing sink tokens from either the decoder or the cross-attention has no measurable impact, contrasting with StreamingLLM [Xiao et al., 2024] for text generation; this may stem from training-time masking or from the prefix acting as an implicit sink. Finally, adding cross-attention at every decoder layer (+70M params) yields no improvement over using only four.

### 7.3 Short-Form Synthesis

Table 3: Short-form evaluation on SEED-en. Metrics: CER, WavLM speaker similarity (SIM), naturalness (N-MOS) and speaker similarity (S-MOS) MOS (mean ± 95% bootstrap CI).

| | **Model** | | **SEED-en** | | | |
|---|---|---|---|---|---|---|
| **Model** | **#Params.** | **Stream.** | CER↓ | SIM↑ | N-MOS↑ | S-MOS↑ |
| Ground Truth | | | 0.8 | 0.73 | $4.04_{\pm 0.19}$ | $4.27_{\pm 0.20}$ |
| VoiceStar | 0.8B | × | 1.4 | 0.55 | $3.69_{\pm 0.26}$ | $3.72_{\pm 0.25}$ |
| VoXtream | 0.4B | ✓ | 1.6 | 0.51 | $3.50_{\pm 0.27}$ | $3.88_{\pm 0.21}$ |
| VoxCPM | 0.8B | × | **0.9** | **0.71** | $\mathbf{3.87}_{\pm 0.22}$ | $\mathbf{4.47}_{\pm 0.18}$ |
| VibeVoice | 1.5B | × | <u>1.0</u> | 0.68 | $3.59_{\pm 0.24}$ | $3.38_{\pm 0.27}$ |
| Enc./Dec. Transformer | 0.3B | × | 1.4 | 0.51 | $3.79_{\pm 0.21}$ | $3.79_{\pm 0.22}$ |
| PSW-Speech | 0.3B | ✓ | 1.4 | 0.52 | $\underline{3.84}_{\pm 0.19}$ | $\underline{3.89}_{\pm 0.21}$ |

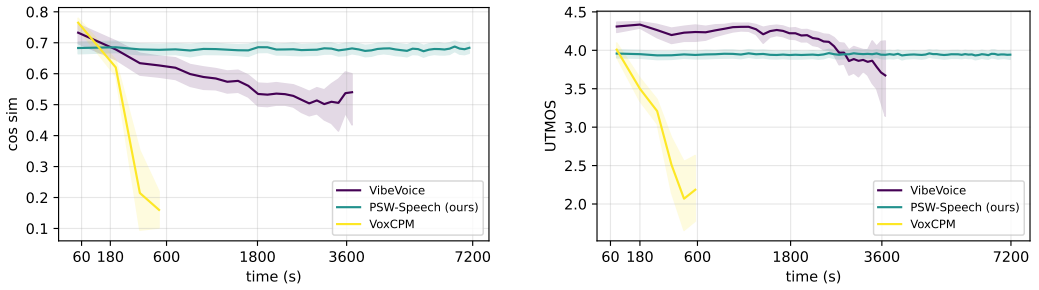
Table 3 reports SEED-en results. PSW-Speech matches its non-windowed encoder–decoder counterpart of equal size on CER and slightly improves both perceptual scores, confirming that restricting

Table 4: Long-form evaluation on Very-Long LibriSpeech, stratified by utterance duration. CER (%, ↓); SIM, N-MOS, S-MOS (↑). MOS values are reported as mean ± 95% bootstrap CI. Models are evaluated only on the duration strata supported by their original protocols.

| | Model | | < 1 min | | | | 1–10 min | | | | 10–100 min | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **Stream.** | **#Params** | CER | SIM | N-MOS | S-MOS | CER | SIM | N-MOS | S-MOS | CER | SIM | N-MOS | S-MOS |
| Original | – | – | – | 0.90 | $4.20_{\pm 0.26}$ | $4.64_{\pm 0.22}$ | – | 0.84 | $4.16_{\pm 0.24}$ | $4.44_{\pm 0.22}$ | – | 0.81 | $4.20_{\pm 0.26}$ | $4.60_{\pm 0.20}$ |
| VoiceStar | × | 0.8B | 2.3 | 0.47 | $2.11_{\pm 0.26}$ | $1.95_{\pm 0.24}$ | – | – | – | – | – | – | – | – |
| VoXtream | ✓ | 0.4B | 1.9 | 0.56 | $3.16_{\pm 0.28}$ | $2.80_{\pm 0.26}$ | – | – | – | – | – | – | – | – |
| VoxCPM | × | 0.8B | 0.9 | 0.77 | $\underline{3.94}_{\pm 0.21}$ | $\underline{4.26}_{\pm 0.19}$ | 2.0 | 0.62 | $3.04_{\pm 0.29}$ | $2.66_{\pm 0.32}$ | – | – | – | – |
| VibeVoice | × | 1.5B | 0.9 | 0.75 | $\mathbf{4.05}_{\pm 0.22}$ | $4.18_{\pm 0.24}$ | 1.0 | 0.66 | $\underline{3.63}_{\pm 0.25}$ | $\underline{3.76}_{\pm 0.27}$ | 1.4 | 0.55 | $3.17_{\pm 0.27}$ | $2.70_{\pm 0.27}$ |
| PSW-Speech | ✓ | 0.3B | 1.1 | 0.67 | $3.88_{\pm 0.20}$ | $\mathbf{4.33}_{\pm 0.19}$ | 1.2 | 0.67 | $\mathbf{3.76}_{\pm 0.23}$ | $\mathbf{4.03}_{\pm 0.23}$ | 1.1 | 0.67 | $\mathbf{3.85}_{\pm 0.22}$ | $\mathbf{4.20}_{\pm 0.22}$ |

attention to sliding windows incurs no measurable cost on short-form synthesis. Against streaming baselines, we improve over VoXtream on every metric despite using fewer parameters, and we match VoiceStar's CER at less than half its size. Compared to the much larger non-streaming systems VoxCPM and VibeVoice, our model trails on objective speaker similarity, which we attribute primarily to corpus size and parameter count rather than to architectural limitations. On perceptual quality, however, PSW-Speech is competitive: it surpasses VibeVoice on both N-MOS and S-MOS and approaches VoxCPM on naturalness (but not on S-MOS, where VoxCPM even exceeds the ground truth), indicating that the proposed local-attention design preserves the quality of short-form synthesis while unlocking streaming and length generalization.

### 7.4 Long-Form Synthesis




(a) Speaker similarity over time.

(b) UTMOS over time.

Figure 6: Objective metrics over time on long-form synthesis. (Left) Speaker similarity (Right) UTMOS.

We evaluate on Very-Long LibriSpeech (Section E), which stratifies utterances into three duration ranges: <1 min, 1–10 min, and 10–100 min. Results are reported in Table 4 and complemented by speaker similarity and UTMOS trajectories in Figure 6.

**Objective results.** PSW-Speech is the only system whose CER and speaker similarity remain flat across all duration strata, while every baseline degrades as duration grows. VibeVoice—a model $5\times$ larger and trained with explicit long-context curriculum learning—and VoxCPM both lead on speaker similarity in the short-form stratum, where their larger capacity helps, but our approach overtakes them on the next stratum (1–10 min). The speaker similarity trajectories in Figure 6 (Left) make this contrast explicit, with baselines decaying steadily over time and PSW-Speech remaining flat. VoiceStar and VoXtream cannot synthesize beyond their original protocols, and VoxCPM is unable to reach the longest stratum.

**Subjective results.** In the short stratum, PSW-Speech trails the larger non-streaming baselines on naturalness while achieving the best speaker similarity score, despite having 3–$5\times$ fewer parameters. As duration grows, the gap widens in favor of our system: PSW-Speech obtains the best naturalness and speaker similarity in both longer strata, while VibeVoice's perceptual scores collapse over the same range. PSW-Speech's MOS values remain essentially flat across all three duration strata, confirming that the model produces consistent quality independent of generation length. Baselines

exhibit distinct artifacts: VoXtream shows progressive decay in speaker similarity, with reduced energy and slowing speaking rate; VoiceStar produces stuttering and repetition, which we attribute to PM-RoPE's unreliable extrapolation even at moderate horizons ($1.5$–$2\times$ training context); VoxCPM and VibeVoice both suffer from error accumulation and speaker drift, earlier for VoxCPM and more gradually for VibeVoice. In contrast, PSW-Speech remains perceptually stable over extended durations, consistent with the flat metrics in Table 4 and Figure 6.

## 8 Limitations

Our study is restricted to English read speech; extending our method to other languages, conversational speech, or code-switching is left for future work. Due to limited access to compute, we restricted our training to relatively small model size. Confirming the scalability of the proposed approach is left for future work. Moreover, our reliance on sampling guidance incurs an additional inference cost. Investigating guidance distillation is left for future work.

## 9 Broader Impact

Our work enables stable, streaming long-form synthesis with modest compute, which benefits accessibility applications such as audiobook generation, screen readers, and low-latency conversational agents. Like all voice cloning systems, it also carries the risks of misuse for impersonation, fraud, and disinformation.

## 10 Conclusion

We introduced PSW-Speech, a streaming encoder–decoder TTS model built on a simple observation: cross-attention in standard TTS models is mostly monotonic and self-attention is overwhelmingly local. Restricting both to sliding windows—with a permanent prefix to anchor speaker identity—yields a fully streamable architecture that generalizes beyond its training horizon. Trained on segments shorter than 30 seconds, PSW-Speech produces stable, consistent speech over multi-hour generations.

## Acknowledgments and Disclosure of Funding

This research was supported by the project EXOVOICES ANR-21-CE23-0040 and funded by the French National Research Agency. This work was granted access to the HPC resources of IDRIS under the allocation 2025-AD011015461R1 made by GENCI. The authors declare no competing interests.

## References

Richard He Bai, Zijin Gu, Tatiana Likhomanenko, and Navdeep Jaitly. SpeakStream: Streaming text-to-speech with interleaved data. *arXiv preprint arXiv:2505.19206*, 2025.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. WhisperX: Time-accurate speech transcription of long-form audio. *Proc. Interspeech 2023*, 2023.

Eric Battenberg, RJ Skerry-Ryan, Soroosh Mariooryad, Daisy Stanton, David Kao, Matt Shannon, and Tom Bagby. Location-relative attention mechanisms for robust long-form speech synthesis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6194–6198. IEEE, 2020.

Eric Battenberg, RJ Skerry-Ryan, Daisy Stanton, Soroosh Mariooryad, Matt Shannon, Julian Salazar, and David Teh-Hwa Kao. Robust and unbounded length generalization in autoregressive transformer-based text-to-speech. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11789–11806, 2025.

Hankun Chen et al. Attention-constrained inference for robust decoder-only text-to-speech. *arXiv preprint arXiv:2404.19723*, 2024.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. F5-TTS: A fairytaler that fakes fluent and faithful speech with flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6255–6271, 2025.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. CosyVoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*, 2024.

Yitian Gong, Botian Jiang, Yiwei Zhao, Yucheng Yuan, Kuangwei Chen, Yaozhou Jiang, Cheng Chang, Dong Hong, Mingshu Chen, Ruixiao Li, et al. MOSS-TTS technical report. *arXiv preprint arXiv:2603.18090*, 2026.

Bing Han, Long Zhou, Shujie Liu, Sanyuan Wang, Jinyu Yang, and Furu Wei. VALL-E R: Robust and efficient zero-shot text-to-speech synthesis via monotonic alignment. *arXiv preprint arXiv:2406.07855*, 2024.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, et al. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 885–890. IEEE, 2024.

Mutian He, Yan Deng, and Lei He. Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural TTS. In *Interspeech*, 2019.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Shehzeen Samarah Hussain, Paarth Neekhara, Xuesong Yang, Edresson Casanova, Subhankar Ghosh, Roy Fejgin, Mikyas T. Desta, Rafael Valle, and Jason Li. Koel-TTS: Enhancing LLM based speech generation with preference alignment and classifier free guidance. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21219–21234, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1076. URL https://aclanthology.org/2025.emnlp-main.1076/.

Dongya Jia, Zhuo Chen, Jiawei Chen, Chenpeng Du, Jian Wu, Jian Cong, Xiaobin Zhuang, Chumin Li, Zhen Wei, Yuping Wang, et al. DiTAR: Diffusion transformer autoregressive modeling for speech generation. In *Forty-second International Conference on Machine Learning*, 2025.

Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. Glow-TTS: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33:8067–8077, 2020.

Ryan Langman, Xuesong Yang, Paarth Neekhara, Shehzeen Hussain, Edresson Casanova, Evelina Bakhturina, and Jason Li. HiFiTTS-2: A large-scale high bandwidth speech dataset. *arXiv preprint arXiv:2506.04152*, 2025.

Théodor Lemerle, Diego Torres, Téo Guichoux, Nicolas Obin, and Axel Roebel. Low-framerate speech tokenization via two-stage latent patch modeling. In *Interspeech 2026*, Sydney, Australia, 2026. ISCA.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37: 56424–56445, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72, 2025.

Paarth Neekhara, Shehzeen Hussain, Subhankar Ghosh, Jason Li, Rafael Valle, Rohan Badlani, and Boris Ginsburg. Improving robustness of LLM-based speech synthesis by learning monotonic alignment. In *Interspeech*, 2024.

Se Jin Park, Julian Salazar, Aren Jansen, Keisuke Kinoshita, Yong Man Ro, and RJ Skerry-Ryan. Long-form speech generation with spoken language models. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=4AmFA0qNQ2`.

Marco Pasini, Javier Nistal, Stefan Lattner, and George Fazekas. Continuous autoregressive models with noise augmentation avoid error accumulation. *arXiv preprint arXiv:2411.18447*, 2024.

Puyuan Peng, Shang-Wen Li, Abdelrahman Mohamed, and David Harwath. VoiceStar: Robust zero-shot autoregressive TTS with duration control and extrapolation. *arXiv preprint arXiv:2505.19462*, 2025a.

Zhiliang Peng, Jianwei Yu, Wenhui Wang, Yaoyao Chang, Yutao Sun, Li Dong, Yi Zhu, Weijiang Xu, Hangbo Bao, Zehua Wang, et al. VibeVoice technical report. *arXiv preprint arXiv:2508.19205*, 2025b.

Simon Rouard, Manu Orsini, Axel Roebel, Neil Zeghidour, and Alexandre Défossez. Continuous audio language models. *arXiv preprint arXiv:2509.06926*, 2025.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. In *Proc. Interspeech*, pages 4521–4525, 2022. doi: 10.21437/Interspeech.2022-439.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, et al. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *ICASSP*, 2018.

Nikita Torgashov, Gustav Eje Henter, and Gabriel Skantze. VoXtream: Full-stream text-to-speech with extremely low latency. *arXiv preprint arXiv:2509.15969*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024.

Neil Zeghidour, Eugene Kharitonov, Manu Orsini, Václav Volhejn, Gabriel de Marmiesse, Edouard Grave, Patrick Pérez, Laurent Mazaré, and Alexandre Défossez. Streaming sequence-to-sequence learning with delayed streams modeling. *arXiv preprint arXiv:2509.08753*, 2025.

Yike Zhang, Yiming Li, et al. Monotonic attention for robust text-to-speech synthesis in large language model frameworks. In *Interspeech*, 2025.

Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Ziyang Wang, Runchuan Ye, Weiyue Sun, Jiancheng Gui, Kehan Li, et al. VoxCPM: Tokenizer-free TTS for context-aware speech generation and true-to-life voice cloning. *arXiv preprint arXiv:2509.24650*, 2025.

# A Self-Attention Locality in the Encoder and Decoder

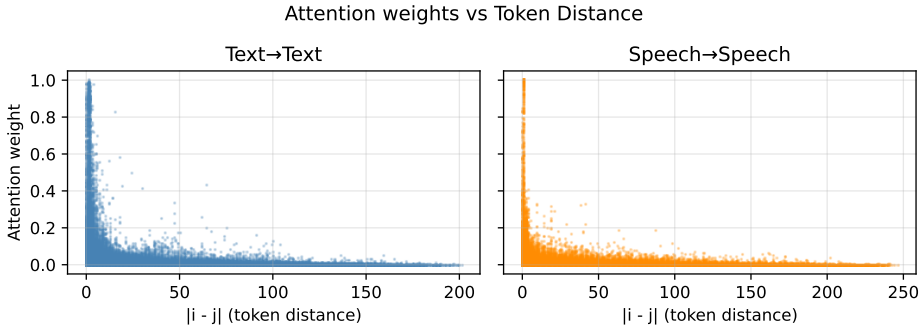



Figure 7: Self-attention normalized score as a function of token distance $|i - j|$ for the encoder (left) and decoder (right), aggregated across all layers and heads for 1000 random utterances from HiFiTTS-2. Sink tokens are excluded. Each point is a (query, key) attention score within the same modality: text→text (left) and speech→speech (right). In both modalities the attention mass is strongly concentrated at short distances and decays rapidly, with only a sparse tail of non-negligible weights reaching the far context.

Figure 7 reports self-attention weights against token distance for the standard (non-windowed) encoder–decoder baseline, pooled across all layers and heads of the encoder and decoder. The decay envelope is remarkably steep: weights above $0.1$ are almost entirely confined to $|i-j| \lesssim 20$ tokens in both modalities, while beyond roughly $50$ tokens the typical attention weight is near zero. Given that the decoder is trained on sequences of several hundred tokens, this indicates that the overwhelming majority of attention mass is allocated within a short local neighborhood of each query, and that long-range interactions, though possible, are rarely exercised in practice.

# B Inference Speed

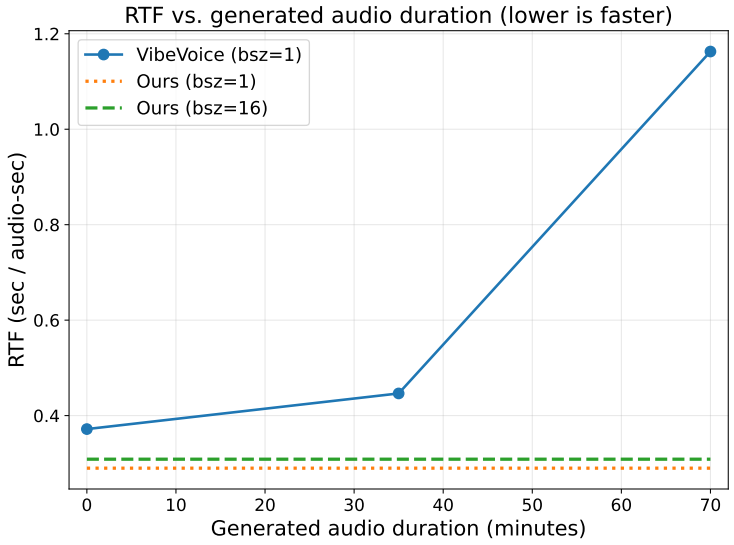



Figure 9: Inference speed on long samples evaluated on an RTX 4090. Timings include all forward passes needed for guidance, which in practice multiplies the AR backbone batch size by 3 and the sampling head batch size by 4.

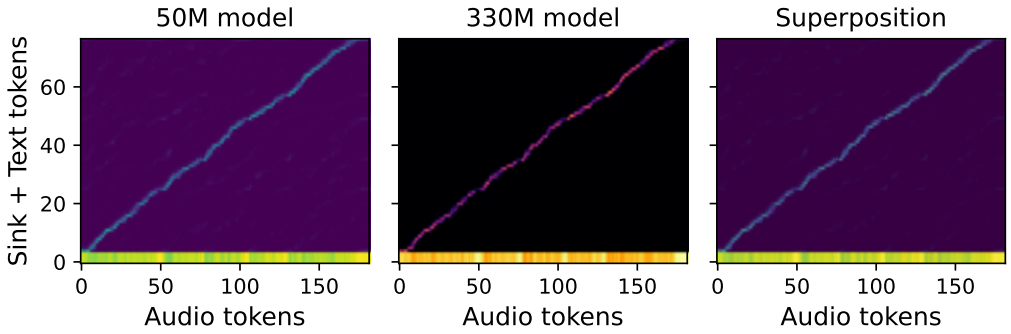



Figure 8: Cross-attention comparison between the 50M alignment provider (left) and a 330M model (middle). The superposition (right) shows that both models exhibit highly similar monotonic alignment structures.

## C Model Architecture and Training Hyperparameters

The PSW-Speech architecture is a standard encoder–decoder transformer with RoPE [Su et al., 2024] positional encoding in both the encoder and decoder, and a pre-norm layout using LayerNorm. The encoder is preceded by three depthwise convolutional blocks [Liu et al., 2022] and followed by a depthwise convolution with stride 2 that downsamples the text hidden features, following Battenberg et al. [2025]. The decoder applies a short causal depthwise convolution to the $Q$, $K$, and $V$ projections.

The encoder–decoder transformer baseline introduced in Section 7.3 shares this architecture (apart from the masking operations) but adds a linear projection head that predicts the end of generation, trained with a binary cross-entropy loss on the last-frame target. This loss is added to the Conditional Flow Matching objective using a weight of $0.1$. PSW-Speech, by contrast, does not rely on explicit stop prediction; it instead uses the tracked cross-attention position as an indicator of reading progress and stops generation once the last text hidden states is reached.

The ablations in Table 2 follow the same protocol, except that the models are trained solely on HiFiTTS-2.

Table 5: Alignment provider encoder–decoder TTS hyperparameters.

| **Parameter** | **Value** |
|---|---|
| Model type | Encoder–decoder Transformer |
| Total parameters | ∼50M |
| Encoder | Bi-Dir. Transformer |
| Decoder | Causal Transformer |
| Training steps | 120k |
| Encoder layers | 6 |
| Decoder layers | 12 |
| Model dimension | 512 |
| Attention heads | 8 |
| Number of sink tokens | 4 |
| Optimizer | AdamW |
| Learning rate | 1e-4 |
| Warmup steps | 5k |
| LR schedule | Cosine |
| Batch size (per GPU) | 50k tokens |
| Training time | 6h on 1× RTX 4090 |

Table 6: PSW-Speech hyperparameters.

| Parameter | Value |
|---|---|
| Model type | Encoder–decoder Transformer |
| Total parameters | $\sim$330M |
| Encoder | Bi-Dir. Transformer |
| Decoder | Prefixed Sliding-Window Decoder Transformer |
| Encoder layers | 6 |
| Decoder layers | 24 |
| Model dimension | 1024 |
| Attention heads | 8 |
| Number of sink tokens | 4 |
| Cross-attention window radius $R$ | $R \sim \mathcal{U}([\![1, 4]\!])$ |
| Self-attention window length | $\sim \mathcal{U}([\![4, 128]\!])$ |
| Prefix length | $\sim \mathcal{U}([\![48, 128]\!])$ |
| Text packet size (tokens) | $\sim \mathrm{LogUniform}(16, 256)$ |
| Optimizer | AdamW |
| Learning rate | 1e-4 |
| Warmup steps | 5k |
| LR schedule | Cosine |
| Training steps | 200k |
| Batch size (per GPU) | 20k tokens |
| Training time | 2 days on 2$\times$ RTX 4090 |

# D Baseline Limitations

Table 7: Maximum input text length and observed synthesis duration during subjective evaluation. Values correspond to the longest samples successfully generated and evaluated for each model. We cap text input lengths following each model's authors' recommendations. For our model, we cap the maximum audio length to keep the study tractable.

| Model | Max input text (chars) | Max audio length |
|---|---|---|
| VoiceStar | 700 | 51s |
| VoXtream | 1000 | 58s |
| VoxCPM | 5k | 6 min |
| VibeVoice | 75k | 67 min |
| **PSW-Speech** | **114k** | **2 h** |

# E Construction of the Very-Long LibriSpeech Dataset

This appendix details the procedure used to construct the *Very-Long LibriSpeech* benchmark, designed for evaluating text-to-speech (TTS) systems under extreme long-form generation conditions.

## E.1 Source Material

The dataset is derived from the *LibriSpeech test-clean* split, which itself originates from public-domain audiobooks distributed by LibriVox. For each utterance in the test-clean set, we retrieve the corresponding full-length LibriVox recordings, together with the associated Project Gutenberg books from which the recordings were read. For each test-clean entry, we retrieve the full chapter and append the following chapters until reaching a maximum of 150k characters, a cap set by the cost of LLM-based text normalization.

This ensures that both speech and text originate from identical literary sources, enabling reconstruction of long-form aligned material.

Table 8: Statistics of the Very-Long LibriSpeech dataset used for long-form evaluation.

| Statistic | Value |
|---|---|
| Number of speakers | 39 |
| Number of books | 53 |
| Mean references per book | 10 |
| Maximum characters per book | 150k |
| Mean prompt duration (s) | 9.7 |
| Mean reference duration (s) | 9.4 |

### E.2 Text Collection and Normalization

Raw Project Gutenberg texts contain substantial non-narrative material, including licensing information, editorial notes, formatting artifacts, and structural metadata. To obtain speech-ready textual content, all books were processed using an automatic normalization pipeline based on a lightweight large language model (OpenAI GPT-5 nano).

**LLM-based normalization prompt.** Text normalization is performed using a LLM (GPT-5 nano) guided by the following fixed prompt:

> **System instruction:**
> You are a text normalization system designed to prepare book-length narrative text for long-form speech synthesis.
>
> **Task:**
> Given a raw Project Gutenberg book, output a cleaned and normalized version of the narrative text suitable for audiobook-style reading.
>
> **Constraints:**
>
> - **Book-content filtering.** Remove all non-narrative material, including Project Gutenberg headers and footers, licensing text, legal notices, recording credits, tables of contents, chapter indices, page numbers, footnotes, editorial notes, and structural markers such as "End of chapter".
> - **Content preservation.** Preserve the original narrative wording, ordering, and dialogue structure exactly. Do *not* summarize, paraphrase, rewrite, or modify the style.
> - **Text normalization.** Convert all numbers (cardinals, ordinals, years, ranges, and percentages) into fully written-out spoken forms. Expand abbreviations and symbols into their spoken equivalents. Normalize quotation marks to standard ASCII single or double quotes.
> - **Formatting cleanup.** Remove all paragraph breaks and line breaks, producing a single continuous text stream. Discard inline reference markers and footnote indices.
>
> **Output format:**
> Return a single continuous text string containing only normalized narrative content, with no additional commentary or metadata.

The output of this step consists of normalized book texts suitable for long-form speech synthesis.

### E.3 Audio–Text Alignment Using WhisperX

To identify reliable long-form regions, we applied WhisperX [Bain et al., 2023] to the LibriVox recordings to obtain word-level timestamps and segmented transcription hypotheses.

For each audiobook, the resulting WhisperX segments were compared against the normalized Gutenberg reference text using text similarity measures. Only segments exhibiting high similarity with the reference text were retained, ensuring that the spoken audio faithfully follows the original book content and excludes deviations such as ad-lib commentary or skipped passages.

### E.4 Construction of Long-Form Evaluation Samples

From the aligned audiobook–text pairs, we extracted continuous segments at multiple temporal scales. Evaluation samples are stratified by duration into three regimes:

- less than 1 minute,
- between 1 and 10 minutes,
- between 10 minutes and several hours.

All samples originate from a single speaker and a single book, enabling controlled evaluation of long-term speaker consistency, alignment stability, and robustness to extreme context-length extrapolation.

### E.5 Subjective Evaluation

We conduct subjective listening tests to assess the perceptual quality of the generated speech. For each experiment, listeners are asked to evaluate two perceptual dimensions:

- **Naturalness:** *"How close to human does the speech sound?"*
- **Speaker similarity:** *"How close to the reference sample does the speech sound?"*.

Both criteria are rated using a standard five-point (1 to 5) Mean Opinion Score (MOS) scale, where higher scores indicate better perceptual quality.

Listening tests are conducted on the Prolific platform. We perform two independent tests for the short-form (Section 7.3) and the long-form (Section 7.4). For both tests, we recruit 50 native or fluent English speakers, each evaluating 20 randomly selected samples per experiment. We reject any trial that rates a ground-truth sample lower than 3.

The reported MOS results correspond to the mean opinion score together with 95% confidence intervals estimated via bootstrap resampling.

## 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 (Section 1) state our two main contributions—(i) a prefixed sliding-window self-attention and sliding cross-attention mechanism enabling streaming and length generalization, and (ii) stable synthesis at hour-long durations despite training only on segments shorter than 30 seconds. These claims are supported by the ablations in Section 7.2, the long-form evaluation in Section 7.4 and Table 4, and the inference-speed analysis in Section B.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Section 8 explicitly discusses the limitations of our work, namely that our study is restricted to English read speech, that scalability of the method is left for future work, and that sampling guidance incurs additional inference cost.

Guidelines:

- The answer [N/A] 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: [N/A]

   Justification: The paper does not include theoretical results; our contributions are empirical, consisting of architectural design choices and experimental validation.

   Guidelines:

   - The answer [N/A] 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: The architecture is described in detail in Section 4 and Figure 2, the inference procedure is given in Algorithm 1, the training and sampling details are provided in Sections 5.2 and 5.3, and full hyperparameters for both the alignment provider and PSW-Speech are reported in Section C (Tables 5 and 6). The continuous speech tokenizer is described in its original publication [Lemerle et al., 2026], and the construction of the Very-Long LibriSpeech benchmark is documented in Section E.

Guidelines:

- The answer [N/A] 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: Code and model checkpoints are not released with the paper, as our codebase still partly relies on internal tools that are not meant to be shared. We nonetheless plan to open-source a public version of our code alongside checkpoints and training scripts at `https://github.com/theodorblackbird/psw-speech`. We provide audio samples on the demo page (`https://psw-speech.github.io/`), and the architecture, training procedure, hyperparameters (Section C), and benchmark construction (Section E) are described in sufficient detail to allow reproduction. The training datasets (HiFiTTS-2, Emilia) and the source materials for our benchmark (LibriSpeech, LibriVox, Project Gutenberg) are publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Training data, optimizer, learning rate schedule, batch size, training steps, and hardware are described in Section 5.2 and detailed in Section C (Tables 5 and 6). Sampling guidance settings (CFG values, ODE steps, noise scaling) are reported in Section 5.3. Evaluation details, including the use of Whisper-v3 for CER and WavLM for speaker similarity, are described in Section 6.1, with subjective evaluation protocols documented in Section E.5.

   Guidelines:

   - The answer [N/A] 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: Subjective MOS scores in Tables 3 and 4 are reported as means with 95% bootstrap confidence intervals, as described in Section E.5. Long-form trajectories in Figure 6 also include shaded confidence regions. Objective metrics (CER, SIM) are computed over the full evaluation sets and are deterministic given the model output, so no error bars are reported for these aggregate values.

   Guidelines:

   - The answer [N/A] 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: Compute requirements are reported in Section 5.2: the alignment provider takes approximately 6 hours on a single RTX 4090 GPU, and PSW-Speech is trained for 2 days on two RTX 4090 GPUs. The compute used to train the speech tokenizer is reported in its original publication [Lemerle et al., 2026]. Inference speed measurements on an RTX 4090 are reported in Section B (Figure 9).

   Guidelines:

   - The answer [N/A] 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: We have reviewed the NeurIPS Code of Ethics and our research conforms to it. We use only publicly available datasets (HiFiTTS-2, Emilia, LibriSpeech, LibriVox, Project Gutenberg), and our subjective evaluation on Prolific recruits adult participants who are compensated according to platform standards.

   Guidelines:

   - The answer [N/A] 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**

    Answer: [Yes]

    Justification: Section 9 discusses positive societal impacts (accessibility applications such as audiobook generation, screen readers, and low-latency conversational agents) and negative ones (misuse of voice cloning for impersonation, fraud, and disinformation).

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release pre-trained model checkpoints or scraped datasets with this paper, so no specific safeguards are required at this stage. The Very-Long LibriSpeech benchmark is derived entirely from public-domain LibriVox and Project Gutenberg material.

    Guidelines:

    - The answer [N/A] 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: All existing assets used in the paper are properly cited: HiFiTTS-2 (Langman et al., 2025), Emilia (He et al., 2024), LibriSpeech and LibriVox (public domain), Project Gutenberg (public domain), Whisper-v3 (Bain et al., 2023), WavLM-based SIM metric (Chen et al., 2025), and baseline systems (VoiceStar, VibeVoice, VoXtream, VoxCPM). Datasets derived from LibriVox and Project Gutenberg are in the public domain; HiFiTTS-2 and Emilia are released under permissive research licenses, which we comply with.

    Guidelines:

    - The answer [N/A] 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` 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: [N/A]

    Justification: We do not release new assets (model weights, code, or datasets) with this paper. The Very-Long LibriSpeech benchmark construction procedure is fully documented in Section E to allow others to reconstruct it from the publicly available source materials.

    Guidelines:

    - The answer [N/A] 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: Section E.5 describes the subjective evaluation protocol: 50 native or fluent English speakers were recruited via Prolific, each evaluating 20 randomly selected samples per experiment, rating naturalness ("How close to human does the speech sound?") and speaker similarity ("How close to the reference sample does the speech sound?") on a five-point MOS scale. Participants are compensated according to Prolific's standard rates, which exceed the minimum wage in the country of recruitment.

    Guidelines:

    - The answer [N/A] 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: [N/A]

Justification: Our subjective evaluation consists of listening to short, non-sensitive speech samples and providing standard MOS ratings. The study presents minimal risk to participants and does not collect personal or sensitive information beyond what is required by the Prolific platform. Participants consent to participation through the Prolific platform terms.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: A lightweight LLM (OpenAI GPT-5 nano) is used to normalize Project Gutenberg texts when constructing the Very-Long LibriSpeech evaluation benchmark. The full prompt and procedure are documented in Section E.2. The LLM is not part of the core method (training or inference of PSW-Speech) but is used in the evaluation data preparation pipeline; we declare it for transparency.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.