Full-stream TTS with dynamic speaking rate control via duration distribution guidance

Anonymous ACL submission

Abstract

Full-stream text-to-speech (TTS) for interactive systems must start speaking with minimal delay while remaining controllable as text arrives incrementally. We present VoXtream2, a zero-shot full-stream TTS model with dynamic speaking-rate control that can be updated mid-utterance on the fly. VoXtream2 combines duration distribution guidance with classifier-free guidance across conditioning signals to improve controllability and synthesis quality. Prompt-text masking enables textless audio prompting, removing the need for prompt transcription. Across standard zero-shot benchmarks and a dedicated speaking-rate test set, VoXtream2 achieves competitive objective and subjective results against public baselines despite a smaller model and less training data. In full-stream mode, it runs 4 times faster than real time with 74 ms first-packet latency on a consumer GPU, outperforming publicly available full-stream models.

1 Introduction

Recent progress in neural text-to-speech (TTS) has produced highly natural and intelligible speech, yet most systems still assume a fixed speaking rate within an utterance or allow only global control. This differs from spontaneous human speech, where speakers slow down while formulating thoughts, insert pauses or fillers such as "uhm" and "hmm", and accelerate when content is prepared. Such variation reflects cognitive load, discourse structure, and communicative intent, and its absence makes synthetic speech less spontaneous.

Conversational agents have long been argued to require incremental speech generation (Schlangen and Skantze, 2009; Skantze and Hjalmarsson, 2013). Voice-driven interfaces based on large language models (LLMs) make this need more urgent (Shikhar et al., 2025). In dialogue and speech-to-speech translation, text unfolds token by token and must be converted into waveform chunks on the fly (Bai et al., 2025). However, most recent neural TTS systems remain offline (Chen et al., 2025a; Ju et al., 2024; Mehta et al., 2024), while streaming-capable systems usually lack explicit speaking-rate control (SRC) or keep it fixed at the utterance level.

These issues are linked: full-stream TTS needs more than low first-packet latency. As input evolves, pace should adapt to uncertainty, emphasis, or urgency. Acoustic prompts convey style, but prompt-based systems usually inherit only average prosody, and instruction following can conflict with voice cloning. We therefore target explicit frame-level SRC that varies within an utterance and can trigger pauses or fillers even when the prompt lacks the desired timing.

We introduce VoXtream2, a zero-shot full-stream TTS model with dynamic SRC adjustable mid-utterance. It uses duration distribution guidance for frame-level control, classifier-free guidance (CFG) for conditioning, prompt text masking for textless audio prompts, and prompt enhancement for robust voice cloning. VoXtream2 runs four times faster than real time with 74 ms first-packet latency on a consumer GPU. Pretrained model, code, and interactive demo are available online.[1] Our main contributions are as follows:

- We introduce a dynamic speaking-rate control mechanism in the stream, which can be modified in real-time.
- In full-stream generation, VoXtream2 achieves the lowest first-packet latency and real-time factor among publicly available models reproduced on the same hardware.
- The performance of our model on zero-shot TTS is on par with state-of-the-art systems, even though it uses less training data and is among the smallest compared to competitors.

[1] https://voxtream2.github.io/demo

2 Related Work

Duration modeling has long been a natural tool for SRC. Earlier work uses sentence-level rate descriptors or manipulates alignments through encoder representations (Bae et al., 2020; Lenglet et al., 2022). More recent systems control sentence, word, or phoneme durations through explicit predictors, scaling factors, or probabilistic duration models (Bandekar et al., 2023; Lee et al., 2024; Kim et al., 2024; Ogura et al., 2025). Although these methods achieve fine-grained temporal control, they are typically evaluated on single-speaker data and do not address voice cloning or streaming synthesis.

Modern zero-shot and controllable TTS systems expose utterance-level SRC through text descriptions, style instructions, category labels, or target-generated duration. Description-based approaches can be accurate but often lack voice cloning (Guo et al., 2023; Lyth and King, 2024). Instruction-based models add style control with acoustic prompting (Yang et al., 2024a; Ji et al., 2025; Du et al., 2025), and duration-control models can enforce target length (Wang et al., 2025d; Chen et al., 2025b; Peng et al., 2025; Zhou et al., 2025). Recent works providing more fine-grained control are MAGIC-TTS (Mai et al., 2026) and WeSCon (Wang et al., 2025b). The first conditions flow-based zero-shot TTS on explicit token-level content duration and pause values for local edits, and the second provides word-level SRC through speech-token resampling. However, none of them support full-stream TTS.

Full-stream systems synthesize speech from incremental text before the full utterance is known. Recent zero-shot models operate with limited look-ahead using interleaved text-speech tokens, temporal masking, or monotonic alignment (Yang et al., 2024b; Sheng et al., 2026; Torgashov et al., 2026). Several require prompt transcription, which is error-prone for fast or noisy prompts. Kyutai-TTS avoids this but delays output to accumulate text context (Zeghidour et al., 2025).

3 Method

As a foundation, we selected the VoXtream (Torgashov et al., 2026) model, one of the fastest full-stream architectures with publicly available training and generation code, making it a natural choice for rapid experimentation.

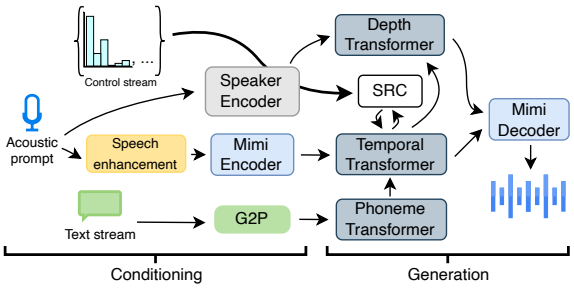

Figure 1: Overview of VoXtream2 architecture.

3.1 Architecture and conditioning updates

Apart from the SRC, we also introduce a couple of other extensions to the VoXtream architecture (Figure 1), to improve the overall performance. For the encoding of phoneme sequences, we use the incremental Phoneme Transformer (PT) and increase the maximum look-ahead size to 25 phonemes. At least 3 phonemes of look-ahead are required for intelligible full-stream generation. Following (Peng et al., 2024) and (Hussain et al., 2025), we automatically extract IPA phonemes. Although our experiments are English-only, IPA is a practical choice for future multilingual finetuning.

The main generative component is the Temporal Transformer (TT), conditioned on PT outputs and audio tokens extracted by the Mimi codec (Défossez et al., 2024) at 12.5 Hz. Following Moshi terminology, we call the first Mimi codebook token a *semantic* token because that codebook was distilled from a WavLM model. The autoregressive Depth Transformer (DT) is conditioned on the TT output embedding, speaker embedding, and the semantic token, and predicts the remaining 15 Mimi codebooks. TT and DT are optimized with cross-entropy losses over respective targets.

To handle punctuation, which was missing in VoXtream, we add punctuation marks as separate PT input tokens and remove their corresponding PT outputs before TT. Thus, TT models duration only for phoneme tokens while still receiving punctuation context. Pauses are not predicted as separate duration labels. Instead, silence and filler-word intervals are merged into adjacent phoneme alignments during preprocessing, so the model learns that staying at a phoneme position for longer results in a pause or a contextually appropriate filler.

The original VoXtream model requires a text transcription of the acoustic prompt and an external phoneme aligner during generation. We remove this requirement through prompt text masking. Dur-

ing training, we select the first 3–10 seconds of each 50-second training window and replace the corresponding text tokens with <UNK> tokens. This masks text only at the beginning of each example, covering about 13% of text tokens on average, and teaches the model both to ignore prompt transcription and to use <UNK> tokens required for CFG at generation. During generation, we assign one <UNK> token to each acoustic frame of the prompt. Similar to (Liu et al., 2026), this enables textless and translingual audio prompting. Translingual generation is demonstrated on the project page, but is not the primary focus of this work.

Following (Hussain et al., 2025), we apply the CFG to each conditioning pathway. The audio tokens in the prompt prefix are masked with probability 10% during training, while the speaker embedding used by DT is dropped with probability 10%. We use separate guidance scales for TT and DT: $\gamma_{temp} = 1.5$ for TT to preserve prosodic variation and $\gamma_{depth} = 3.0$ for DT to improve speaker similarity. To further improve voice cloning, we increase the speaker-embedding conditioning weight by 50%. After applying CFG, we observe that stronger speaker similarity can make the synthesized signal quality converge toward the acoustic prompt quality. We therefore enhance only the acoustic prompt using Sidon (Nakata et al., 2026), following (Giraldo et al., 2026). This does not increase generation latency and improves robustness when prompts contain noise or recording artifacts.

3.2 Dynamic speaking-rate control

VoXtream2 controls speaking rate by guiding the model's local duration-token distribution to a target distribution for the desired syllables-per-second (SPS) value. At each acoustic timestep, the TT predicts logits for a joint distribution over the first Mimi codebook token and a duration state. Let N be the Mimi vocabulary size and let $D = 6$ be the number of duration states. Each duration state encodes a phoneme shift $s \in \{0, 1, 2\}$, which specifies how many phoneme positions to advance before the next frame, together with the number of unique phonemes $u \in \{1, 2\}$ aligned to the current frame. Thus, TT emits a single vector of length ND per acoustic frame, rather than six independent acoustic-token distributions.

We inherit the monotonic alignment mechanism from VoXtream (Torgashov et al., 2026): generation maintains a pointer into the phoneme sequence, and each duration state moves it forward, assigning audio frames to phonemes in left-to-right order. Compared with VoXtream, we extend the duration-state set from four to six states and replace plain duration sampling with the distribution-guided sampler below, enabling dynamic SRC.

Figure 2 illustrates the sampling procedure. Without SRC, we reshape the TT output into a matrix $A \in \mathbb{R}^{D \times N}$ and marginalize over the acoustic-token axis to obtain a duration distribution. We sample a duration token using top-$p = 0.9$, select the corresponding row of A, and sample the first Mimi codebook token from that row using top-$k = 5$. The selected duration token updates the monotonic alignment state, while the selected first-codebook token conditions the Depth Transformer, which predicts the remaining codebooks.

We construct target distributions from the training corpus by aligning phonemes to audio frames and converting the resulting monotonic alignments into duration tokens. For each utterance, we estimate its speaking rate by dividing the transcript syllable count by the audio duration. We then aggregate duration-token counts for utterances with integer SPS labels from 1 to 7 and normalize them into target distributions. At generation time, the user provides only the target SPS. Integer SPS values select the corresponding target distribution directly, while continuous values use linear interpolation between the two nearest integer distributions. This makes it possible to vary the target speaking rate frame by frame during generation.

SRC enforces the target rate by comparing the target distribution P_{target} with an accumulated distribution P_{acc} estimated over the most recent 3 seconds of generated speech. The resulting weights reweight the predicted duration distribution:

$$P_{\text{predicted}} = \text{Softmax}\left(\frac{1}{T}\log\sum_{n=1}^{N}\exp(A_{dn})\right), \tag{1}$$

$$W = \exp\left(\beta\left[\log_{10}(P_{\text{target}}) - \log_{10}(P_{\text{acc}})\right]\right), \tag{2}$$

$$P_{\text{updated}} = \frac{P_{\text{predicted}} \odot W}{\sum_{i=1}^{D} P_{\text{predicted},i} W_i}. \tag{3}$$

Here, $T = 0.9$ is the sampling temperature, A_{dn} is the TT logit for duration state d and first-codebook token n, and P_{acc} is initialized uniformly. The 3-second window provides a stable estimate while still allowing the control signal to change over time. The reweighted distribution P_{updated}

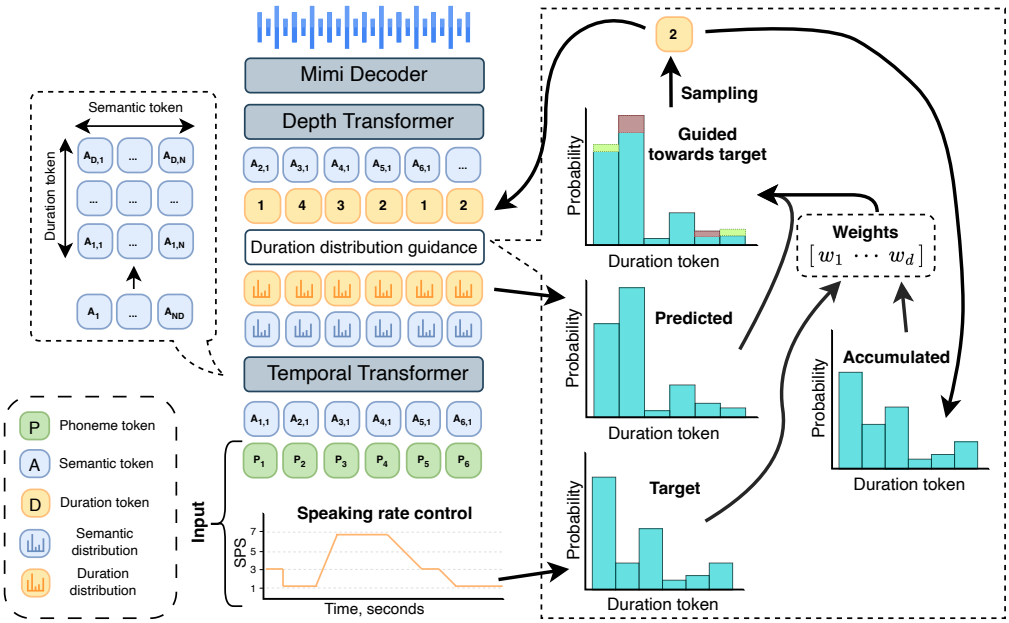

Figure 2: Speaking-rate control mechanism.

is used only for duration-token sampling. The acoustic token is then sampled from the row of A corresponding to the selected duration state. The strength of SRC is controlled by β: $\beta = 1$ has little effect on intelligibility but provides weak control, whereas $\beta = 10$ improves rate tracking at the cost of higher WER due to hallucinations. We therefore use $\beta = 5$ as a trade-off. Smoothing ensures that all bins in P_{target} and P_{acc} remain strictly positive.

Distribution guidance is optional and is applied only when SRC is enabled. Otherwise, duration tokens are sampled directly from $P_{\text{predicted}}$. To avoid the speaking-rate increase observed in Parakeet (Darefsky et al., 2024) and SSR-Speech (Wang et al., 2025a), we do not apply CFG to the duration state. Instead, CFG is applied only to model conditioning, as described in Section 3.1.

To improve naturalness at slow speaking rates, we encourage implicit filler-word generation during training. Instead of providing explicit cues, the model must infer filler placement from prosodic patterns, primarily elongated phoneme durations. Consequently, it learns to insert fillers automatically according to speaking rate and text context. See Appendix E for analysis.

4 Experiment Setup

4.1 Datasets

We used the Emilia (He et al., 2024) spontaneous speech dataset and HiFiTTS-2 (Langman et al., 2025), derived from LibriVox (Kearns, 2014) audiobooks, as the basis of our training corpus. From Emilia, we selected the 47k-hour English subset. From HiFiTTS-2, we used a 15k-hour 22kHz training subset, removing utterances shorter than 5 seconds or with a word error rate (WER) above 10%. IPA phonemes were extracted from transcripts using espeak-ng phonemizer[2]. Phoneme-to-audio alignment was performed with the Clap-IPA forced aligner (Zhu et al., 2024). After filtering failed alignments and invalid transcripts, the final corpus comprised 30k hours from Emilia and 10k hours from HiFiTTS-2, totaling 40k hours. For speech tokenization, we used the Mimi (Défossez et al., 2024) audio codec at 24 kHz. Filler words (e.g., “uh”, “uhm”, “yeah”) were removed from transcripts, and their phoneme timestamps, along with silence intervals, were merged with adjacent phonemes in the alignment. To reach a target duration of 55 seconds per sample, we concatenated utterances at the speaker level and padded shorter clips with silence. This exposes the model to within-sample changes in speaking rate and reduces the tendency to simply continue the speaking rate of the acoustic prompt, which is important for the target rate changes mid-utterance during SRC.

4.2 Model

Following VoXtream (Torgashov et al., 2026), we adopt an open-source implementation of the Llama-3.2 (Dubey et al., 2024) transformer as the backbone of our model. The TT module has 12 layers,

[2] https://github.com/espeak-ng/espeak-ng

Table 1: Evaluation results of zero-shot TTS models. Bold indicates the **best** result and underline indicates the second-best result. Naturalness is measured as MUSHRA scores (0–100).

Model	#Data(h)	#Params	SEED *test-en*			LibriSpeech-PC *test-clean*			Naturalness ↑
			WER (%) ↓	SPK-SIM ↑	UTMOS ↑	WER (%) ↓	SPK-SIM ↑	UTMOS ↑	$\mu \pm 95\%$ CI
Human	-	-	1.50	0.734	3.53	1.95	0.695	4.10	56.7 ± 2.6
VoiceStar	65k EN	840M	2.20	0.605	3.92	2.14	0.603	4.25	60.0 ± 2.6
MaskGCT	100k Multi.	1048M	2.03	**0.712**	3.56	2.58	0.691	3.92	57.6 ± 2.6
Spark-TTS	102k Multi.	507M	1.85	0.572	3.94	2.11	0.578	**4.35**	63.2 ± 2.5
CosyVoice3	1M Multi.	820M	1.64	0.698	**3.96**	1.76	**0.695**	4.29	$\mathbf{66.2} \pm 2.5$
Kyutai-TTS	88k EN	750M	**1.34**	0.689	3.63	**1.63**	0.660	4.04	60.5 ± 2.4
VoXtream2	40k EN	462M	1.45	0.679	3.56	2.08	0.658	4.03	60.2 ± 2.5

16 attention heads, an embedding size of 1024, and a feed-forward dimension of 4096. The PT consists of 6 layers with 8 attention heads. The DT contains 4 layers, 8 attention heads, and a feed-forward dimension of 8192, following the Sesame-CSM model (Schalkwyk et al., 2025). We keep the DT weights frozen, as it was pretrained on a large-scale conversational speech dataset[3]. Speaker embeddings are extracted using a ReDimNet (Yakovlev et al., 2024) trained on over 100k identities.

Training was performed for 230k steps and took 28 hours on 2×NVIDIA H200 GPUs with a batch size of 64 per GPU for 10 epochs. During training, we randomly crop fixed 50 s audio segments. We use the AdamW optimizer (Loshchilov and Hutter, 2019), linearly warming up the learning rate during the first epoch from 1×10^{-5} to 2×10^{-4}. The training graph is compiled with `torch.compile` to maximize GPU utilization. During generation, TT and DT are wrapped with CUDA Graphs following Moshi (Défossez et al., 2024), and the streaming state of the Mimi codec is cached, significantly improving performance over VoXtream. CFG is implemented via batching with negligible overhead. All random seeds are fixed during training and generation to ensure reproducibility.

4.3 Baseline models

We selected publicly available zero-shot TTS models with full-stream or duration-control capabilities as baselines. MaskGCT (Wang et al., 2025d) is a non-autoregressive (NAR) model, and VoiceStar (Peng et al., 2025) and Spark-TTS (Wang et al., 2025c) are autoregressive (AR) models with explicit duration control. CosyVoice3 (Du et al., 2025), Kyutai-TTS (Zeghidour et al., 2025), and VoXtream (Torgashov et al., 2026) are AR full-stream models. CosyVoice3 additionally supports SRC via instructed generation. For a fair comparison, we restricted reproduced systems to those with up to 1B parameters, prioritizing systems trained on public datasets such as Emilia and LibriVox. All reproduced results use official implementations and publicly available checkpoints.

We also considered F5-TTS (Chen et al., 2025b), as a model with duration control, and Qwen3-TTS (Hu et al., 2026) as a full-stream model. However, we exclude these models, as we were unable to generate intelligible speech with strict duration control for F5-TTS in our experiments, and a streaming code was not released for the Qwen3-TTS.

4.4 Evaluation

We evaluate VoXtream2 on three test sets. The first is SEED-TTS (Anastassiou et al., 2024) *test-en*, evaluated with the original protocol. The second is LibriSpeech-PC (Meister et al., 2023) *test-clean* subset, using the cross-sentence protocol introduced in F5-TTS (Chen et al., 2025b).

The third set, denoted Emilia *speaking-rate*, targets SRC evaluation and is derived from Emilia-EN (He et al., 2024). It contains 62 gender-balanced unseen speakers, half of whom have filler words in the prompt, uniformly distributed across genders. Each speaker provides three audio prompts corresponding to slow (2.0 ± 0.5 SPS), normal (4.0 ± 0.6 SPS), and fast (5.6 ± 0.5 SPS) speaking rates. The prompt length distribution matches SEED *test-en*, with a mean duration of 4.5 seconds. Texts were generated using ChatGPT (OpenAI, 2025), prompted with conversational-style sentences from the selected speakers and instructed to produce stylistically similar 30–40-syllable sentences, corresponding to roughly 9–10 seconds at a moderate speaking rate. For dynamic SRC, we followed the same procedure but requested 50–80-syllable sentences to capture multiple speaking-rate transitions within a single utterance. The dataset is publicly available[4].

[3] https://github.com/SesameAILabs/csm

[4] https://hf.co/datasets/voxtream2/test

Table 2: Performance in FP16 on an RTX3090 GPU in full-stream. **TC** denotes Torch Compile, **CG** denotes CUDA graphs, and **TRT** corresponds to TensorRT.

Model	FPL(ms) ↓	RTF ↓
CosyVoice3:**TRT**	2201	0.381
Kyutai-TTS:**CG**	277	0.272
VoXtream:**TC**	78	0.223
VoXtream2:**CG**	74	0.256
VoXtream2:**TC**	**63**	**0.173**

We report three reproducible objective metrics. Intelligibility is measured by WER between the input text and Whisper-large-v3 ASR (Radford et al., 2023) transcriptions of synthesized speech, after applying the Whisper text normalizer as in (Zeghidour et al., 2025). Speaker similarity (SPK-SIM) is computed as the cosine similarity between embeddings from the acoustic prompt and synthesized speech, extracted using a WavLM-based (Chen et al., 2022) ECAPA-TDNN (Desplanques et al., 2020) model. Signal quality is monitored with the UTMOS predictor (Saeki et al., 2022). All metrics are calculated as a mean for a single model run.

For naturalness evaluation, we conducted two Prolific user studies: one for short-form TTS and one for controllable speaking rate. We sampled 100 unique sentences with WER below 10% to exclude hallucinations. In each study, 40 native listeners rated each model on a 0–100 naturalness scale using a MUSHRA-like protocol. Multiple attention checks were included, and participants were compensated at an average rate of 9 GBP/hour.

5 Results

5.1 Overall performance

We start by analyzing the overall performance of VoXtream2 in a non-streaming setting, without SRC. Table 1 shows the results, compared to prior zero-shot TTS systems that support streaming, duration control, or both. Despite its smaller model size and substantially less training data, **VoXtream2 achieves objective performance comparable to state-of-the-art systems**. Its UTMOS is not the highest among baselines, but Table 4 shows that this drop is mainly associated with stronger CFG-based voice conditioning, and can be mitigated with prompt enhancement described in Section 3.1. The corresponding enhanced-prompt evaluation is provided in Appendix D. In subjective evaluation, VoXtream2 obtains an average naturalness score, but remains on par with Kyutai-TTS despite using a smaller model and training corpus. The original VoXtream is excluded from this comparison due to its much smaller training corpus.

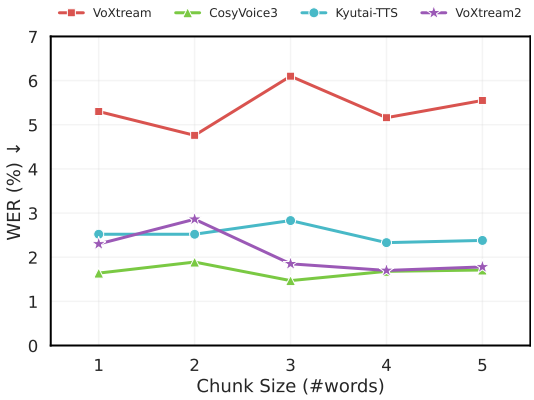

Figure 3: Evaluation of text look-ahead in full-stream.

5.2 Full-stream performance

We benchmark full-stream latency performance by providing the model with the text incrementally, word by word. We measure real-time factor (RTF), the ratio of generated speech duration to wall-clock time, and first-packet latency (FPL), the time to the first speech frame. Both are measured from text input to waveform output. We use the first 10 rows of the SEED *test-en* protocol as a benchmark, with one additional warmup utterance. Table 2 compares the performance against prior work, including different GPU inference optimizations, showing that **VoXtream2 achieves the lowest FPL and RTF among publicly available full-stream models**. For additional analysis, see Appendix F.

We further analyze how input text look-ahead affects intelligibility in the full-stream setting on the Emilia *speaking-rate* dataset (Figure 3). The original VoXtream is less accurate overall, partly because it still requires a prompt aligner during inference, while VoXtream2 no longer needs this step. Kyutai-TTS is comparable to VoXtream2 under word-level streaming, but gains only marginally from additional context. CosyVoice3 achieves the lowest WER on this test, while VoXtream2 steadily closes the gap as text look-ahead increases. However, this accuracy comes with substantially higher latency, as CosyVoice3 has more than 2 s FPL, compared to 74 ms for VoXtream2 (Table 2). We do not report SPK-SIM or UTMOS for the full-stream setting, as these metrics show no significant differences compared to non-streaming evaluation.

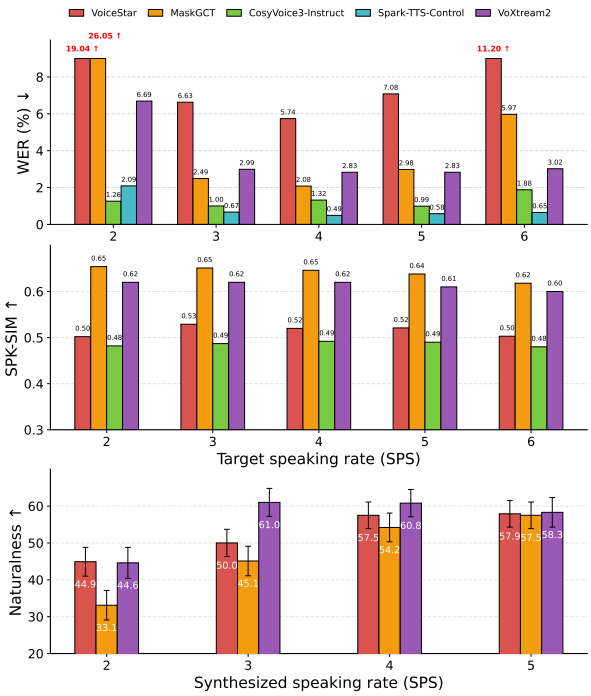

Figure 4: Comparison of different TTS models across various speaking rates for utterance-level control.

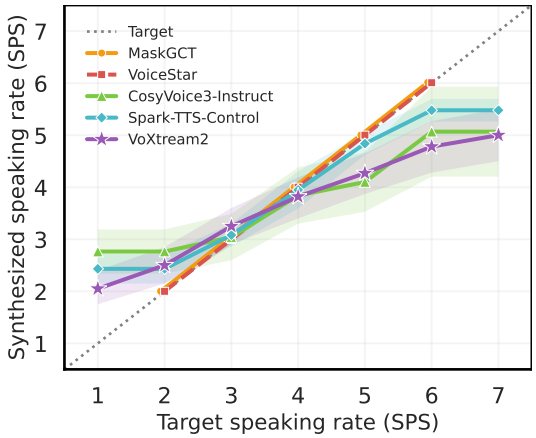

Figure 5: Correlation between target and synthesized speaking rates across different TTS models.

5.3 Static speaking rate control

We use the Emilia *speaking-rate* dataset for SRC evaluations. Figure 5 shows correlation between generated and target speaking rates. CosyVoice3 with instructed generation follows rate changes over 3–5 SPS, and its controllability is close to VoXtream2 in this interval. However, it remains unreliable for very slow speech and shows the largest standard deviation among the evaluated systems. Spark-TTS is more accurate for fast speech, but supports only five discrete speaking-rate states. **VoXtream2 instead provides continuous control within the 2–5 SPS range**. VoiceStar and MaskGCT closely match the target through duration control, but with degraded speech quality. We evaluate duration-control models only within 2–6 SPS to match the operating range of other systems. For prompt speaking rate analysis see Appendix B.

Objective results for SRC on Emilia *speaking-rate* are reported in the first two plots of Figure 4. VoXtream2 has higher WER than CosyVoice3 and Spark-TTS, but achieves stronger voice-cloning performance. MaskGCT is competitive at moderate rates (3 and 5 SPS) and performs best at 4 SPS, but its intelligibility degrades sharply at extreme rates, and in terms of voice cloning it slightly outperforms VoXtream2. VoiceStar shows the highest WER and degrades at extreme speaking rates, with lower SPK-SIM than VoXtream2. Together, these results suggest that explicit duration control can perfectly generate the requested speaking rate, but often trades off against intelligibility. Spark-TTS is excluded from SPK-SIM evaluation because its controllable version does not support voice cloning, yielding an average SPK-SIM of 0.05.

Subjective results appear in the third plot of Figure 4. We exclude Spark-TTS, as the controllable version does not support voice cloning. We also exclude CosyVoice3 because it does not cover the full 2–5 SPS range used in the study. For VoXtream2, samples are selected by achieved speaking rate, e.g., target rate 1 for 2 SPS and target rate 7 for 5 SPS. **VoXtream2 is generally perceived as more natural over varying speaking rates**. It significantly outperforms the other systems at 3 SPS and is comparable at 5 SPS. It achieves slightly higher preference scores at 4 SPS and performs on par with VoiceStar at 2 SPS. For slow speech (2–3 SPS), VoXtream2 clearly outperforms MaskGCT.

5.4 Dynamic speaking rate control

To evaluate dynamic speaking-rate control, we use the Emilia *speaking-rate* test set and report objective metrics. Acoustic prompts are the same as in the static evaluation, but we generate longer sentences of 50–80 syllables. We consider four scenarios: two with gradual rate changes, where the SRC control signal is linearly increased or decreased within 1–7 SPS, and two more challenging cases, where the target rate alternates between slow (1 SPS) and fast (7 SPS) every 40 phonemes. To estimate the speaking-rate trajectory, we extract phoneme alignments from our model, convert phonemes to syllables, and compute SPS using a 3-second sliding window with 25% overlap. This

Table 3: Evaluation of dynamic speaking rate control. S, N, and F denote Slow, Normal, and Fast speaking rates.

Model	Corr.	WER (%) ↓	SPK-SIM ↑	UTMOS ↑
No SRC	-	3.26	0.624	3.86
S (1 SPS)	-	16.51	0.620	3.53
N (4 SPS)	-	3.20	0.626	3.83
F (7 SPS)	-	3.52	0.609	3.81
S → N → F	0.701	4.73	0.628	3.75
F → N → S	0.826	11.25	0.626	3.71
S ↔ F	0.619	9.55	0.629	3.66
F ↔ S	0.660	11.20	0.627	3.64

window can smooth brief local fluctuations in the generated rate. The resulting curve is interpolated onto the time axis. For examples see Appendix H. Static SRC results are reported for reference.

In the reference static SRC setting (Table 3), the model behaves consistently at a normal speaking rate. At fast rates, SPK-SIM decreases slightly while WER remains almost unchanged. At slow rates, however, WER increases substantially. Manual inspection suggests two main causes: hallucinations caused by the underrepresentation of very slow duration states in the training data, and increased word repetition or filler insertion. The former reflects genuine generation errors, whereas the latter can be perceptually acceptable, and sometimes more human-like. Therefore, WER may overestimate the intelligibility loss at very slow rates.

Table 3 shows Pearson correlation (Corr.) between the dynamic SRC signal and the generated speaking rate, indicating a strong correlation. Relative to the baseline, WER increases at extreme control values as hallucinations become more frequent, while SPK-SIM remains stable and UTMOS degrades only slightly. The Corr. metric further indicates that starting from a fast rate yields stronger controllability than starting slowly, although at the cost of intelligibility. The fast SRC starts to produce more frequent phoneme shifts, providing the distribution guidance with more alignment events and a clearer early estimate of P_{acc}. Slow starts instead depend on long dwell times, which are underrepresented in training and may trigger fillers or repetitions before the controller accumulates enough local evidence to recover.

Because dynamic SRC unfolds through real-time interaction, and due to the lack of comparable systems, a controlled subjective study is outside the scope of this paper. Instead, we provide a live demo[5] for readers to interact with our system.

[5] https://hf.co/spaces/voxtream2/demo

Table 4: Modular analysis of VoXtream2 modifications.

Model	WER (%) ↓	SPK-SIM ↑	UTMOS ↑
VoXtream	3.16	0.529	3.88
+ Data & updates	2.29	0.578	<u>3.94</u>
+ CFG text	<u>1.37</u>	0.578	3.91
+ CFG audio	1.62	0.661	3.65
+ CFG speaker	1.55	<u>0.674</u>	3.60
+ Speaker weight (VoXtream2)	1.45	**0.679**	3.56
+ Prompt enhance.	**1.32**	0.656	**4.05**

5.5 Ablations

Table 4 presents a modular ablation of VoXtream2 on the SEED *test-en* dataset. The first row shows consistent gains over VoXtream across all objective metrics, driven by dataset scaling and architectural changes. Applying CFG only to text conditioning substantially reduces WER, while adding audio CFG notably improves SPK-SIM, at the cost of higher WER and lower UTMOS. Speaker-embedding CFG further improves SPK-SIM and slightly reduces WER, but also causes a small UTMOS drop. Increasing the speaker-embedding conditioning weight has a similar effect and yields the best SPK-SIM. Finally, adding speech enhancement to the acoustic prompt achieves the best UTMOS and WER, with a slight reduction in SPK-SIM due to comparison to non-enhanced audio.

6 Conclusion

VoXtream2 shows that ultra-low-latency full-stream zero-shot TTS can be paired with practical, fine-grained control. We have proposed a novel algorithm based on duration distribution guidance, supporting dynamic speaking rate control that can be updated mid-utterance, enabling smooth transitions between slow, normal, and fast speech while maintaining stable intelligibility and voice cloning in the operating range. Prompt text masking enables textless prompts, removing a major usability barrier and reducing sensitivity to prompt content and speaking rate. Classifier-free guidance applied across conditioning signals improves intelligibility and speaker similarity in zero-shot TTS and provides an effective lever to trade off similarity, signal quality, and controllability. Overall, evaluations indicate that VoXtream2 is competitive with publicly available zero-shot baselines and significantly faster than real-time on user-grade hardware, making it suitable for real-time conversational applications.

Limitations

The main limitation of VoXtream2 is the complexity of the data preprocessing pipeline. Although the external phoneme aligner is not required during generation, it remains a crucial component of training data preparation. As shown in Section 4.1, approximately 35% of the data had to be discarded due to invalid alignments. In future work, we plan to explore alternative alignment methods to reduce this dependency.

Another limitation is English-only generation. Although the model architecture is language-agnostic and supports any language for an acoustic prompt, in this work, we were focused on speaking rate control and did not explore cross-lingual capabilities. We plan to extend VoXtream2 to support multiple languages in future work.

References

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, and 1 others. 2024. Seed-TTS: A family of versatile speech generation models. *arXiv:2406.02430*.

Jae-Sung Bae, Hanbin Bae, Young-Sun Joo, Junmo Lee, Gyeong-Hoon Lee, and Hoon-Young Cho. 2020. Speaking Speed Control of End-to-End Speech Synthesis Using Sentence-Level Conditioning. In *Interspeech 2020*, pages 4402–4406.

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

Jesuraj Bandekar, Sathvik Udupa, Abhayjeet Singh, Anjali Jayakumar, Sandhya Badiger, Saurabh Kumar, Pooja VH, Prasanta Kumar Ghosh, and 1 others. 2023. Speaking rate attention-based duration prediction for speed control tts. *arXiv preprint arXiv:2310.08846*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Sanyuan Chen, Chengyi Wang, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and 1 others. 2025a. Neural codec language models are zero-shot text to speech synthesizers. *IEEE Transactions on Audio, Speech and Language Processing*, 33:705–718.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Zhao Jian, Kai Yu, and Xie Chen. 2025b. F5-TTS: A fairytaler that fakes fluent and faithful speech with flow matching. In *ACL*.

Jordan Darefsky, Ge Zhu, and Zhiyao Duan. 2024. Parakeet.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *Proc. Interspeech*.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, Keyu An, Guanrou Yang, Yabin Li, Yanni Chen, Zhifu Gao, Qian Chen, Yue Gu, Mengzhe Chen, Yafeng Chen, and 3 others. 2025. CosyVoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *arXiv preprint arXiv:2505.17589*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and 1 others. 2024. The Llama 3 herd of models. *arXiv:2407.21783*.

Jose Giraldo, Alex Peiró-Lilja, Rodolfo Zevallos, and Cristina España-Bonet. 2026. Zero-shot TTS with enhanced audio prompts: Bsc submission for the 2026 wildspoof challenge TTS track. *arXiv preprint arXiv:2602.05770*.

Zhifang Guo, Yichong Leng, Yihan Wu, Sheng Zhao, and Xu Tan. 2023. PromptTTS: Controllable text-to-speech with text descriptions. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, and 1 others. 2024. 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.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, Xinyu Zhang, Pei Zhang, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*.

Shehzeen Samarah Hussain, Paarth Neekhara, Xuesong Yang, Edresson Casanova, Subhankar Ghosh, Roy Fejgin, Mikyas T Desta, Rafael Valle, and Jason Li. 2025. Koel-TTS: Enhancing LLM based speech generation with preference alignment and classifier free guidance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21230–21245.

Shengpeng Ji, Qian Chen, Wen Wang, Jialong Zuo, Minghui Fang, Ziyue Jiang, Hai Huang, Zehan Wang, Xize Cheng, Siqi Zheng, and 1 others. 2025. ControlSpeech: Towards simultaneous and independent zero-shot speaker cloning and zero-shot language style control. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6966–6981.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, and 1 others. 2024. NaturalSpeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *Proc. ICML*.

Jodi Kearns. 2014. LibriVox: Free public domain audiobooks.

Taewoo Kim, Choongsang Cho, and Young Han Lee. 2024. Masked duration model for utterance duration-controllable text-to-speech. *IEEE Access*.

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

Jaeuk Lee, Sohee Jang, and Joon-Hyuk Chang. 2024. Neural ATSM: Fully Neural Network-based Adaptive Time-Scale Modification Using Sentence-Specific Dynamic Control. In *Interspeech 2024*, pages 4903–4907.

Martin Lenglet, Olivier Perrotin, and Gérard Bailly. 2022. Speaking Rate Control of end-to-end TTS Models by Direct Manipulation of the Encoder's Output Embeddings. In *Interspeech 2022*, pages 11–15.

Qingyu Liu, Yushen Chen, Zhikang Niu, Chunhui Wang, Yunting Yang, Bowen Zhang, Jian Zhao, Pengcheng Zhu, Kai Yu, and Xie Chen. 2026. Cross-lingual F5-TTS: Towards language-agnostic voice cloning and speech synthesis. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proc. ICLR*.

Dan Lyth and Simon King. 2024. Natural language guidance of high-fidelity text-to-speech with synthetic annotations. *arXiv preprint arXiv:2402.01912*.

Jialong Mai, Xiaofen Xing, and Xiangmin Xu. 2026. Magic-tts: Fine-grained controllable speech synthesis with explicit local duration and pause control. *Preprint*, arXiv:2604.21164.

Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter. 2024. Matcha-TTS: A fast TTS architecture with conditional flow matching. In *Proc. ICASSP*.

Aleksandr Meister, Matvei Novikov, Nikolay Karpov, Evelina Bakhturina, Vitaly Lavrukhin, and Boris Ginsburg. 2023. Librispeech-PC: Benchmark for evaluation of punctuation and capitalization capabilities of end-to-end ASR models. In *2023 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 1–7. IEEE.

Wataru Nakata, Yuki Saito, Yota Ueda, and Hiroshi Saruwatari. 2026. Sidon: Fast and robust open-source multilingual speech restoration for large-scale dataset cleansing. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Tadashi Ogura, Takuma Okamoto, Yamato Ohtani, Erica Cooper, Tomoki Toda, and Hisashi Kawai. 2025. Phoneme-level duration controllable neural text-to-speech with phoneme embedding skip connection and modified gaussian duration modeling. *IEEE Access*.

OpenAI. 2025. GPT-5 Technical Report. https://openai.com/.

Puyuan Peng, Po-Yao Huang, Shang-Wen Li, Abdelrahman Mohamed, and David Harwath. 2024. VoiceCraft: Zero-shot speech editing and text-to-speech in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12442–12462, Bangkok, Thailand. Association for Computational Linguistics.

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

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. UTMOS: Utokyo-sarulab system for VoiceMOS challenge 2022. In *Proc. Interspeech*.

Johan Schalkwyk, Ankit Kumar, Dan Lyth, Sefik Emre Eskimez, Zack Hodari, Cinjon Resnick, Ramon Sanabria, Raven Jiang, and the Sesame team. 2025. Crossing the uncanny valley of conversational voice. Blog post, Sesame research.

David Schlangen and Gabriel Skantze. 2009. A general, abstract model of incremental dialogue processing. In *Proceedings of EACL*, pages 710–718.

Zhengyan Sheng, Zhihao Du, Shiliang Zhang, Zhijie Yan, and Liping Chen. 2026. SyncSpeech: Efficient and low-latency text-to-speech based on temporal masked transformer. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16682–16686. IEEE.

Sambal Shikhar, Mohammed Irfan Kurpath, Sahal Shaji Mullappilly, Jean Lahoud, Fahad Shahbaz Khan, Rao Muhammad Anwer, Salman Khan, and Hisham Cholakkal. 2025. LLMVoX: Autoregressive streaming text-to-speech model for any LLM. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20481–20493.

Gabriel Skantze and Anna Hjalmarsson. 2013. Towards incremental speech generation in conversational systems. *Computer Speech and Language*, 27(1):243–262.

Nikita Torgashov, Gustav Eje Henter, and Gabriel Skantze. 2026. VoXtream: Full-stream text-to-speech with extremely low latency. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Helin Wang, Meng Yu, Jiarui Hai, Chen Chen, Yuchen Hu, Rilin Chen, Najim Dehak, and Dong Yu. 2025a. SSR-Speech: Towards stable, safe and robust zero-shot text-based speech editing and synthesis. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Tianrui Wang, Haoyu Wang, Meng Ge, Cheng Gong, Chunyu Qiang, Ziyang Ma, Zikang Huang, Guanrou Yang, Xiaobao Wang, Eng Siong Chng, and 1 others. 2025b. Word-level emotional expression control in zero-shot text-to-speech synthesis. *Proc. NeurIPS*.

Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, and 1 others. 2025c. Spark-TTS: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. *arXiv preprint arXiv:2503.01710*.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. 2025d. MaskGCT: Zero-shot text-to-speech with masked generative codec transformer. In *Proc. ICLR*.

Ivan Yakovlev, Rostislav Makarov, Andrei Balykin, Pavel Malov, Anton Okhotnikov, and Nikita Torgashov. 2024. Reshape dimensions network for speaker recognition. In *Proc. Interspeech*.

Dongchao Yang, Songxiang Liu, Rongjie Huang, Chao Weng, and Helen Meng. 2024a. InstructTTS: Modelling expressive TTS in discrete latent space with natural language style prompt. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2913–2925.

Yifan Yang, Shujie Liu, Jinyu Li, Hui Wang, Lingwei Meng, Haiyang Sun, Yuzhe Liang, Ziyang Ma, Yuxuan Hu, Rui Zhao, and 1 others. 2024b. Interleaved speech-text language models for simple streaming text-to-speech synthesis. *arXiv preprint arXiv:2412.16102*.

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

Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. 2025. Indextts2: A breakthrough in emotionally expressive and duration-controlled auto-regressive zero-shot text-to-speech. *arXiv preprint arXiv:2506.21619*.

Jian Zhu, Changbing Yang, Farhan Samir, and Jahurul Islam. 2024. The taste of IPA: Towards open-vocabulary keyword spotting and forced alignment in any language. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 750–772.

A Prompt text masking

Table 5 presents a comparison between models trained with and without prompt text masking on the Emilia *speaking-rate* dataset. The proposed approach removes the need for an external phoneme aligner while maintaining stable WER across different prompt speaking rates relative to the baseline. The speaking-rate block should not be interpreted as SRC: neither CFG nor SRC was applied in this experiment. It shows that the generated speaking rate remains influenced by the prompt rate even with prompt text masking.

Table 5: Evaluation of prompt text masking on different prompt speaking rates. Mean $\pm$ STD is reported.

Model	Slow	Normal	Fast
WER (%) $\downarrow$			
Baseline	2.80 ± 5.64	2.10 ± 4.32	4.21 ± 8.10
Prompt mask	2.14 ± 3.27	2.20 ± 3.54	2.13 ± 3.73
Speaking rate			
Baseline	3.15 ± 0.82	3.48 ± 0.80	3.58 ± 0.84
Prompt mask	2.76 ± 0.72	3.09 ± 0.74	3.37 ± 0.84

B Prompt speaking rate analysis

To analyze the influence of prompt speaking rate, we compute the correlation between generated and target rates for prompts with different speaking speeds. Figure 6 shows that SRC substantially reduces dependence on the prompt rate. The dotted curves indicate generation without SRC. With SRC enabled, the achievable speaking-rate range is significantly broader than the variation induced by "slow", "normal", and "fast" prompts, and differences between curves are much smaller than those observed without SRC.

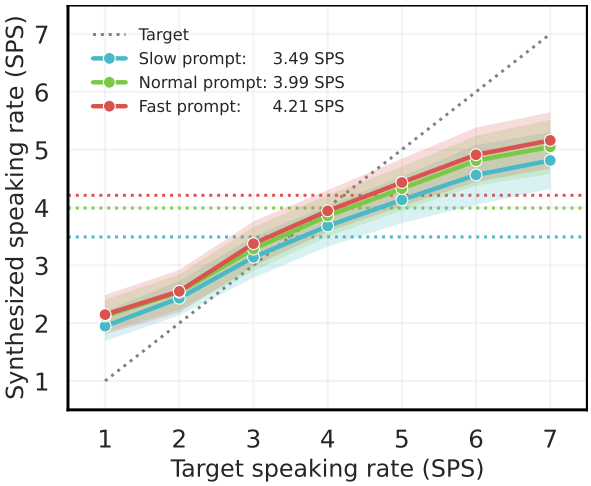

Figure 6: Influence of the prompt speaking rate on the predicted speaking rate.

C CFG effects on speaking rate control

Motivated by the speaking-rate acceleration effect of CFG observed in (Darefsky et al., 2024; Wang et al., 2025a), we investigated the influence of the γ_{temp} parameter on VoXtream2 under SRC. Since CFG is not applied during duration-token sampling, we did not observe a significant change in speaking rate. However, we found a clear relationship between γ_{temp} and WER in SRC (Figure 7). Lower γ_{temp} values, around 1.15–1.25, yield the best WER for slow speech but the worst for fast speech, while larger values, around 2.5–3.0, produce the lowest WER for fast speech but degrade intelligibility at slow rates. Importantly, tuning γ_{temp} did not noticeably change SPK-SIM or UTMOS in our experiments, likely because most Mimi codebooks are generated by DT, whose guidance scale is kept fixed. These results indicate that CFG still indirectly influences speaking-rate behavior even without being applied to duration states. Under SRC, this effect mainly appears as changes in intelligibility rather than speed.

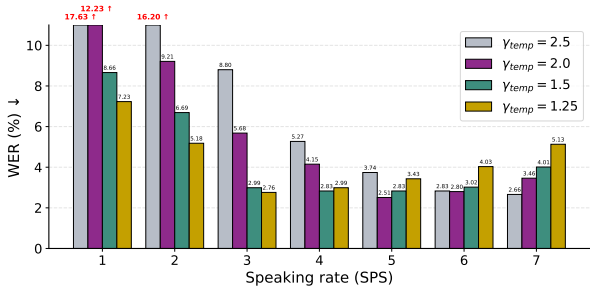

Figure 7: Influence of the CFG γ on the speaking rate control.

D Evaluation on enhanced datasets

Table 6 shows that the main conclusions from the original evaluation remain unchanged: despite using less training data and a smaller model, VoXtream2 remains competitive with larger systems, achieving the second-best WER on SEED. The main difference is the gain in predicted quality with enhanced prompts: VoXtream2 UTMOS increases from 3.56 to 4.05 on SEED and from 4.03 to 4.23 on LibriSpeech-PC, while its WER and SPK-SIM are maintained or improved on both datasets.

E Filler word analysis

We further analyze filler-word generation (Figure 8). A clear relationship is observed between filler frequency and speaking rate: fillers are more frequent at slow rates and nearly disappear at fast rates,

Table 6: Evaluation results of zero-shot TTS models on enhanced datasets. Bold indicates the **best** result and underline indicates the second-best result. Acoustic prompts enhanced with Sidon (Nakata et al., 2026) model.

Model	#Data(h)	#Params	SEED *test-en* Enhanced			LibriSpeech-PC *test-clean* Enhanced		
			WER (%) ↓	SPK-SIM ↑	UTMOS ↑	WER (%) ↓	SPK-SIM ↑	UTMOS ↑
Human	-	-	1.58	0.697	4.03	1.95	0.658	4.15
Spark-TTS	102k Multi.	507M	2.11	0.556	**4.35**	2.04	0.549	**4.42**
MaskGCT	100k Multi.	1048M	1.73	**0.712**	4.17	2.11	0.688	4.21
CosyVoice3	1M Multi.	820M	1.64	0.710	4.28	1.63	**0.692**	4.38
VoiceStar	65k EN	840M	1.46	0.622	4.34	1.70	0.605	4.43
Kyutai-TTS	88k EN	750M	**1.25**	0.699	4.21	**1.49**	0.669	4.32
VoXtream2	40k EN	462M	1.32	0.691	4.05	2.05	0.674	4.23

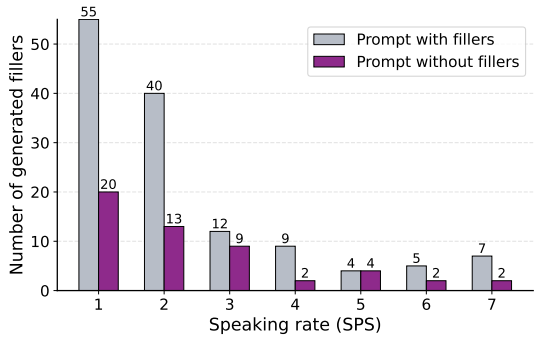

Figure 8: Dependency between the number of generated filler words and speaking rate.

consistent with human speech. Prompts containing fillers further increase their occurrence. However, even without such cues, VoXtream2 inserts fillers automatically at slow rates. Among the evaluated systems, only VoXtream2 exhibits this adaptive filler insertion behavior.

F Full-stream analysis

To evaluate robustness when receiving text from an upstream LLM, we varied the tokens-per-second (TPS) rate of the input text stream (treating each word as a token) and measured its impact on performance. The look-ahead (LA) column indicates how many future phonemes are required to generate the current audio frame. Infinite TPS corresponds to immediate text availability with no delay. Results are summarized in Table 7. The first row reports metrics for output streaming to quantify the increase in FPL and WER in the full-stream setting. The next rows show that TPS=40 (25 ms per token, similar to (Sheng et al., 2026)) does not affect the model, as speech generation remains slower than text arrival, effectively providing ample look-ahead. At TPS=20, WER increases because the TTS model and text generator operate at similar speeds, reducing effective look-ahead. In the final rows, TPS is further reduced so text arrives slower than speech generation. We then analyze the effect of LA on WER and generation speed. With LA below 3, FPL is minimal, but the output becomes unintelligible. LA=3 provides the best trade-off, maintaining moderate WER even at TPS=10 while keeping FPL low. Increasing LA to 5 phonemes still reduces WER at TPS=10 because generation waits until enough text has arrived to satisfy the larger look-ahead. This waiting time is reflected by the higher FPL values.

Table 7: Effect of input text streaming rate (TPS) and phoneme look-ahead (LA) on full-stream TTS performance.

TPS	LA	WER (%) ↓	FPL(ms) ↓	RTF ↓
-	∞	1.44	44	0.229
∞	3	2.29	75	0.256
40	3	2.29	75	0.254
20	3	2.98	67	0.259
10	1	56.13	56	0.262
10	2	22.89	56	0.274
10	3	5.11	81	0.274
10	4	3.46	146	0.271
10	5	2.66	176	0.280

G Voice cloning improvements

Prior modular analysis of VoXtream (Torgashov et al., 2026) found a trade-off: freezing the pre-trained Depth Transformer improves UTMOS, whereas training it from scratch improves SPK-SIM at substantially lower UTMOS. We investigate an intermediate approach by fine-tuning the pre-trained Depth Transformer (VoXtream2-FDT). As shown in Table 8, this preserves UTMOS and significantly improves SPK-SIM, approaching state-of-the-art voice-cloning performance, at the cost of a small WER increase.

Table 8: Zero-shot TTS results for the VoXtream2-FDT comparison on the SEED *test-en* dataset.

Model	WER (%) ↓	SPK-SIM ↑	UTMOS ↑
Human	1.50	0.734	3.53
VoXtream2	**1.45**	0.679	3.56
VoXtream2-FDT	1.67	**0.707**	**3.57**

H Dynamic speaking rate examples

Figure 9 illustrates examples of dynamic SRC. The model follows gradual changes and adapts rapidly to abrupt transitions in the control signal. Short-term fluctuations around the target rate resemble natural speaking-rate variation in human speech and help maintain naturalness rather than indicating control failure. The system reliably generates speech at 1 SPS, while the maximum reliable speaking rate is around 5–6 SPS.

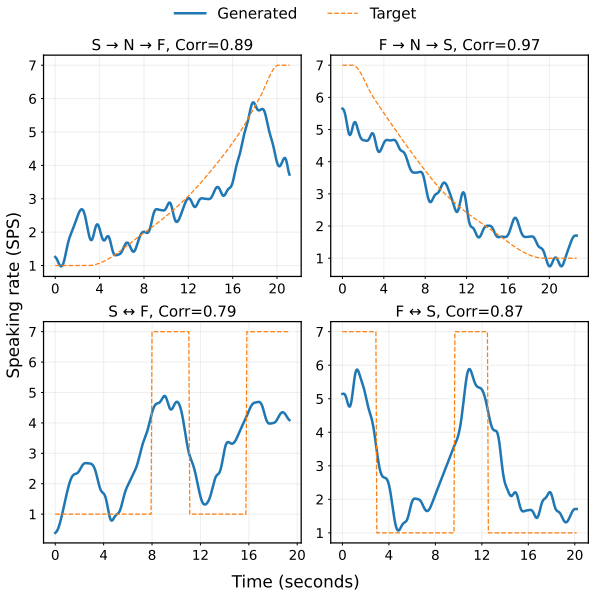

Figure 9: Examples of dynamic speaking rate control. S, N, and F denote Slow, Normal, and Fast target speaking rates.

I Potential Risks

VoXtream2 can reproduce speaker identity from an acoustic prompt, creating risks of impersonation, deception, and non-consensual speech generation. We therefore release the configuration in Table 1, whose speaker similarity is comparable to public systems such as Kyutai-TTS, but withhold the higher-similarity Depth Transformer fine-tuning ablation in Table 8. This reports the technical finding while limiting access to the configuration that most increases cloning strength.

We do not rely on watermarking as a primary mitigation. Kyutai-TTS (Zeghidour et al., 2025), which also uses Mimi (Défossez et al., 2024), reports that watermarking in open TTS is fragile: source-level mechanisms can be disabled, and common audio watermarks can be largely removed by a single 32-codebook Mimi encode-decode pass. We expect similar limitations for VoXtream2. Reliable traceability and abuse prevention remain open problems.

J Artifacts License

We use HiFiTTS-2 (Langman et al., 2025) under the Creative Commons Attribution 4.0 International (CC BY 4.0) license and the Emilia English subset (He et al., 2024) under the license and access conditions stated in its gated public repository. Our use of Emilia is limited to research on full-stream TTS and speaking-rate control.

The released VoXtream2 model and Emilia *speaking-rate* test set remain subject to the applicable licenses and access conditions of their source datasets, with attribution to the dataset creators. The test set is derived from Emilia-EN and intended for research evaluation of controllable and streaming TTS. Training and evaluation code is released under the MIT License, which applies only to the software and does not override terms governing data, derived evaluation data, or model weights.

K Subjective evaluation

Figure 10 shows the screenshot of the user consent form and instructions for the subjective evaluation. Participants were informed about the purpose of the study, the nature of the audio samples, and their rights as participants. Participants were encouraged to use headphones and listen in a quiet environment for optimal evaluation. We also inserted random attention checks to ensure data quality.

Figure 11 shows the interface for the subjective evaluation. Participants were asked to listen to 7 audio samples per page generated by different TTS models and rate their naturalness on a 0–100 scale. The instructions and hints on the next steps were provided on each page. The order of samples was randomized for each participant to mitigate order effects. The volume of all audio samples was normalized to ensure a consistent listening experience.

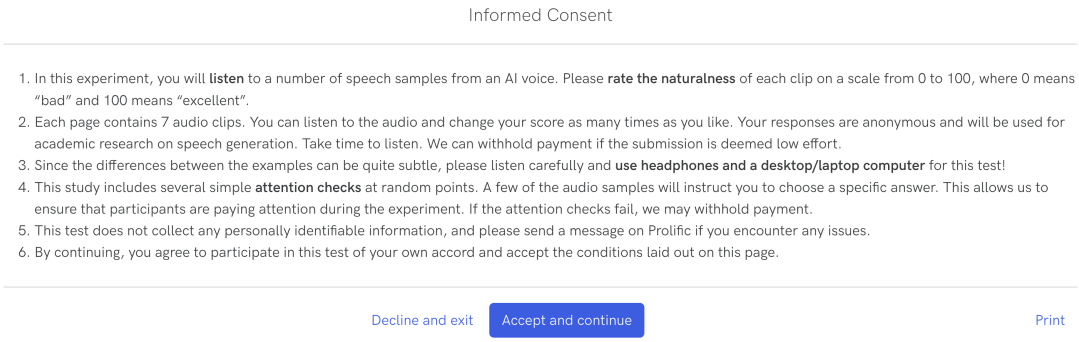

Figure 10: Informed consent form for subjective evaluation.

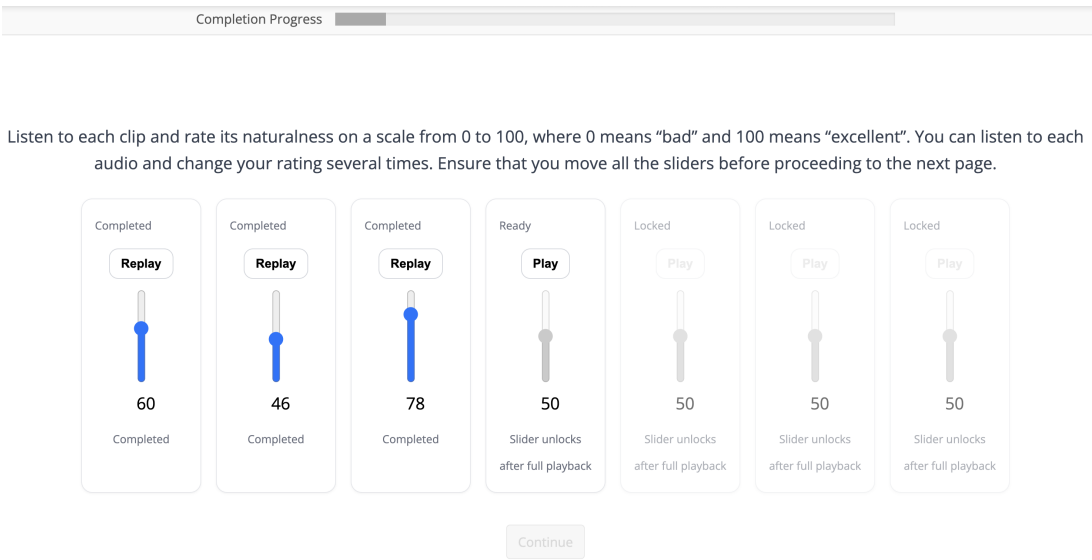

Figure 11: Subjective evaluation interface.