LACONIC: CONFIGURABLE LOW-RATE SPEECH REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Speech representations spend capacity along two axes: how many tokens are emitted per second, and how many dimensions each token carries. Yet they are usually characterized by token rate alone, with token dimensionality treated as a fixed architectural choice. We ask how capacity should be allocated between the two. We introduce LACONIC, a continuous speech representation whose token rate is chosen at encoding time and whose token dimensionality is chosen at reading time, as a nested prefix of each token, so that a single encoder exposes a two-axis family of operating points. Mapping this space, we find that token rate and dimensionality are partially interchangeable between roughly 3 and 6 tokens per second, but the exchange reaches a temporal floor at 1 token per second, where no tested dimensionality recovers the lost content. Different information also occupies different regions of the space. Under our nested objective, linguistic content becomes readable from low-dimensional prefixes while speaker identity emerges only at high-dimensional ones, and is carried more compactly by a small set of utterance-level tokens. The same geometry governs generation. In zero-shot TTS, low-dimensional tokens are easiest to generate at higher token rates, high-dimensional tokens help as the token rate decreases, and speaker similarity and perceptual quality favor different dimensionalities. These results suggest treating speech representation capacity not as a single rate budget, but as an allocation across token rate and dimensionality.

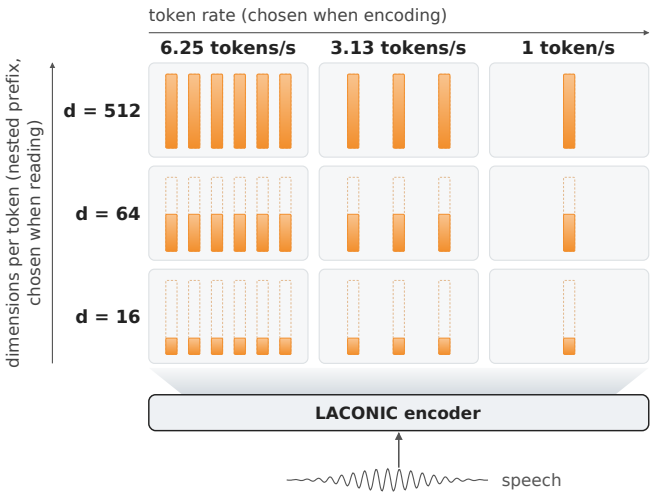


Figure 1: One LACONIC encoder yields a family of speech representations. The token rate r is chosen when speech is encoded; the token dimensionality d is chosen when the tokens are read, as a nested prefix of each token, without re-encoding. Over 3.13–6.25 tokens/s, 16–32 dimensions per token keep ASR WER within 20% relative of uncompressed WavLM (Section 4.1); at 1 token/s, additional dimensions do not compensate for the lost temporal resolution.

1 INTRODUCTION

Modern speech representations are quite verbose. The same recording is represented by most self-supervised speech encoders at 50 vectors per second natively (Hsu et al., 2021; Chen et al., 2022), and most neural codecs at 12.5 or above (Défossez et al., 2024). For example, WavLM-Large (Chen et al., 2022) produces 50 1024-dimensional vectors per second of speech (resulting in 51,200

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

numeric values per second. we note here that our experiments hold at finite precisions). The dimensionality and token rate impose different costs. Token rate determines sequence length and therefore the number of steps processed in an autoregressive model, as well as the cost of attention in a transformer Vaswani et al. (2017) model, while the dimensionality determines how much information should be represented or generated at each step. A compact speech representation should therefore answer two questions: how many tokens should represent a second of speech, and how much information should each token carry?

These questions are usually coupled. However, speech representations are commonly characterized by their token rate, while the dimensionality of each representation is largely treated as a fixed architectural choice. Yet equal-sized representations can have very different shapes: the same budget of n numeric values per second can be allocated to many low dimensional tokens or fewer higher dimensional ones. It's unclear how different choices of such allocation affect model performance across tasks. This raises a more general question:

How should speech representation capacity be allocated across token rate and dimensionality?

Inspired by this question, we introduce LACONIC, a continuous speech representation with independently configurable token rate and dimensionality. LACONIC maps features from a frozen self-supervised representation model into a sequence of local tokens L together with a fixed set of utterance-level tokens U. A single encoder is trained across token rates from 1 to 6.25 tokens per second with reconstruction and text objectives to produce dimension-nested representations, while the nested structure allows each local token to be read at different dimensions *without* re-encoding the utterance.

Contributions. We (1) formulate and empirically characterize the rate–dimension frontier of speech representations; (2) introduce LACONIC, a single encoder that independently exposes token rate and nested token dimensionality; and (3) show across linguistic understanding, speaker identity, reconstruction, and zero-shot TTS that different information and downstream consumers occupy different regions of this rate–dimension space.

2 RELATED WORK

The development of neural universal speech representations has been closely tied to self-supervised learning. Starting with wav2vec 2.0 (Baevski et al., 2020), large-scale self-supervised pre-training has become a standard approach for speech processing. Self-supervised speech representations such as HuBERT (Hsu et al., 2021) and WavLM (Chen et al., 2022) have demonstrated strong performance across a wide range of speech tasks (Yang et al., 2021), particularly those requiring phonetic information (Hsu et al., 2021). More recently, neural audio codecs (Zeghidour et al., 2021; Défossez et al., 2023; Kumar et al., 2023) have emerged as another mainstream approach to universal speech representation. Initially developed for speech generation, subsequent work has demonstrated their utility for speech understanding as well (Wu et al., 2024). SpeechTokenizer (Zhang et al., 2024) and Mimi (Défossez et al., 2024) further bridge the two paradigms by distilling self-supervised speech representations into audio codec models. However, a common challenge across these approaches is their high token rates, typically up to 50 tokens per second. When such representations are used as inputs to downstream models, the resulting long token sequences substantially increase computational costs, particularly the quadratic attention cost of modern Transformer architectures.

Recent efforts have sought to reduce the token rate of speech representations (Casanova et al., 2025; Yang et al., 2025b; Della Libera et al., 2025; Libera et al., 2026). Inspired by the natural syllabic structure of speech, SyllableLM (Baade et al., 2025) and Sylber (Cho et al., 2025) reduce the token rate to $\sim$ 4–8,Hz by discovering acoustic boundaries, while TASTE (Tseng et al., 2025) pushes the rate to around 3,Hz, at the cost of requiring text information at *inference* time; we note here that Laconic uses text at training to help guide the linguistic information in lower dimensions, but does not use any text at inference. Other work instead compresses learned speech representation sequences to reduce the computational cost of downstream Transformer models (Lu et al., 2024; Bhati et al., 2025; Xiang et al., 2026). Despite these efforts to reduce sequence length, the trade-off between token rate and feature dimensionality remains largely unexplored. Luo et al. (2026) touch on this trade-off by arguing that high-dimensional continuous representations with short sequences can be more stably predicted by autoregressive models than existing speech representations; however, their

study focuses exclusively on speech generation and does not systematically vary the two axes. FlexiCodec (Li et al., 2025) introduces an inference-time configurable token rate to accommodate the rapidly varying information density of speech, but focuses exclusively on the temporal dimension. In contrast, LACONIC explicitly studies the trade-off between token rate and feature dimensionality across both speech understanding and generation, providing independent inference-time control over both axes. Our nested token dimensionality is inspired by Matryoshka Representation Learning (Kusupati et al., 2022) and its precursor, Nested Dropout (Rippel et al., 2014), which encourage prefixes of learned representations to remain useful at reduced dimensionalities. Together, these two forms of flexibility allow LACONIC to decouple temporal and dimensional compression, enabling the representation to accommodate the needs of different downstream tasks while providing a framework for systematically studying the trade-off between token rate and feature dimensionality.

3 LACONIC

We study speech representations along two independently controllable axes: token rate r, measured in tokens per second, and token dimensionality d, measured in dimensions per local token. The same representation budget rd can therefore be allocated to a higher token rate with lower-dimensional tokens, or a lower token rate with higher-dimensional tokens. Studying this tradeoff requires varying the two axes without training a different representation for every operating point.

LACONIC provides these controls within a single encoder. Given speech, it produces a sequence of *local tokens* together with a fixed set of *utterance-level tokens* U. Token rate is selected when the speech is encoded, while token dimensionality is selected by reading a prefix of the resulting local tokens. Thus one encoder exposes a two-dimensional family of representations that we use in Section 4 to map how speech information changes across rate and dimensionality.

3.1 CONTROLLING TOKEN RATE AND DIMENSIONALITY

Figure 2 summarizes the architecture. Starting from a frozen WavLM-Large encoder, we layer-normalize and average its 24 Transformer layers, producing $X \in \mathbb{R}^{T\times 1024}$ at 50 Hz. A Transformer encoder maps these features to five utterance tokens $U \in \mathbb{R}^{5\times 512}$ and a sequence of local tokens $L \in \mathbb{R}^{\lceil T/R\rceil \times 512}$.

Token rate. Local queries are inserted every R frames, giving a token rate of $r = 50/R$ tokens/s. We train over $R \in \{8, 10, 12, 16, 25, 50\}$, corresponding to $r \in \{6.25, 5, 4.17, 3.13, 2, 1\}$ tokens/s. One R is sampled for each training example. The same encoder therefore supports all six token rates rather than assigning a separately trained encoder to each rate.

The encoder uses chunk-causal attention. Local tokens attend to the frames associated with their temporal region and to preceding context, so the local stream can be computed causally. The utterance tokens attend globally.

Token dimensionality. Every local token has maximum dimensionality $d_{\max} = 512$, but a downstream task need not use all 512 dimensions. Instead, it may store/use a nested prefix $L^{(d)} = L_{:,:d}$ The representation is trained so that these prefixes remain useful at progressively lower dimensionalities. We use prefixes down to $d = 8$ during tokenizer training and evaluate even smaller prefixes when probing the resulting representation.

Ignoring the fixed-cost utterance tokens, an operating point (r, d) contains $B = rd$ local scalar values per second. We refer to B as the *representation budget*. We acknowldge here that U isnt part of this budget as it can be seen a constant throughout, and we furthermore show it mostly does not matter for WER D.2. Importantly, equal representation budget does not imply equal representation shape: the same B may consist of a higher token rate with lower-dimensional tokens or a lower token rate with higher-dimensional tokens.

3.2 LEARNING NESTED REPRESENTATIONS

Merely truncating an ordinary representation does not make its early dimensions useful. LACONIC therefore applies three complementary objectives. Reconstruction requires multiple dimensionalities

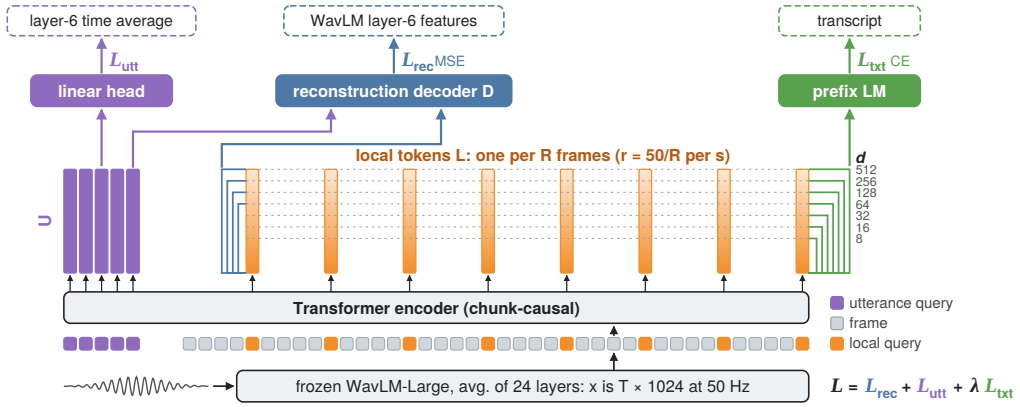


Figure 2: LACONIC. A single encoder exposes two controls over the local representation. Query spacing determines token rate r during encoding; a nested prefix determines token dimensionality d without re-encoding. Five utterance tokens U provide a fixed-cost pathway for utterance-level information.

to retain information about the speech signal; text prediction encourages linguistic information to appear in low-dimensional prefixes; and utterance-level prediction provides a separate pathway for information that need not be temporally localized.

Reconstruction. Let Y denote WavLM layer-6 features from the same utterance. During training we sample $d \leq d_{\max}$ and reconstruct Y from only the selected local prefix and the utterance tokens using a Transformer reconstruction decoder D trained with $\mathcal{L}_{\text{rec}} = \|D(U, L^{(d)}) - Y\|_2^2$. Mask tokens are inserted in-between every local tokens to match the length of the feature sequence Y, and 0 is filled into the suffix dimensions of the decoder input following $L^{(d)}$ when $d < d_{max}$. Because the reconstruction decoder sees different dimensional prefixes during training, information required for reconstruction cannot reside exclusively in the final dimensions.

Linguistic ordering. Reconstruction alone does not specify which information should appear earliest. We therefore use a lightweight causal text decoder to predict the transcript s from local tokens, $\mathcal{L}_{\text{txt}} = -\log p(s \mid L^{(d)})$, where d is sampled during training. The text decoder does not receive U. Requiring both low- and high-dimensional prefixes to predict the same transcript encourages linguistic information to be concentrated in the earliest dimensions.

Utterance-level information. Finally, a linear prediction head P taking the five utterance tokens to predict the temporal average of the reconstruction target $\bar{Y}$ is trained with loss $\mathcal{L}_{\text{utt}} = \|P(U) - \bar{Y}\|_2^2$. Nested dropout is applied to encourage the most important utterance-level information to be encoded in the first of the five tokens. This creates a duration-independent pathway for information that can be represented globally rather than repeated in the local sequence.

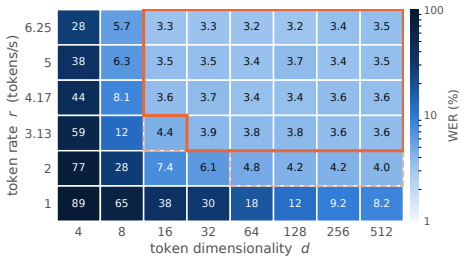
The complete training objective is $\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{utt}} + 0.3\,\mathcal{L}_{\text{txt}}$, which encourages the representation to preserve information across varying token rates and dimensional prefixes, while prioritizing linguistic information in the prefix dimensions. This design does not assume that the two axes will be interchangeable or that equal-budget representations will be equivalent. Whether different types of speech information require different combinations of token rate and dimensionality is an empirical question to be investigate in Section 4.

3.3 TRAINING AND SPEECH DECODING

The LACONIC encoder and feature reconstruction decoder are 8-layer Transformers with model dimension 512 and approximately 25.2M parameters each. We train the model on LibriSpeech 960 h for 20 epochs while keeping WavLM-Large frozen.

Representation	$(r \times d)$	Values/s	WER (%)
WavLM	$(50 \times 1{,}024)$	51,200	3.3
WavLM-Stack	$(3.13 \times 16{,}384)$	51,200	3.9
WavLM-Pool	$(3.13 \times 1{,}024)$	3,200	3.9
WavLM-PCA	(50×64)	3,200	7.7
Laconic	(3.13×32)	100	3.9
Laconic	(3.13×512)	1,600	3.6

(a)



(b)

Figure 3: ASR word error rate (WER) of the 44M-parameter ASR model on LibriSpeech test-clean (4–10 s). (a) WavLM features reshaped along token rate r (tokens/s) or token dimensionality d, and Laconic. Values/s is the number of scalar values per second, $r \times d$; 3.13 denotes $50/16 = 3.125$ tokens/s. (b) Laconic across the full $r \times d$ surface. The solid orange line outlines the regime within 20% relative of the WavLM WER; the dashed line outlines the regime within 50%.

For speech reconstruction we experiment with two setups. The first combines the reconstruction decoder with a frozen HiFi-GAN that converts WavLM layer-6 features into raw audio. The second uses a conditional flow-matching decoder that takes Laconic features U and $L^{(d)}$ as inputs to predict Mel spectrograms, which are then converted to audio using BigVGAN-v2. During flow-matching decoder training, we randomly sample the token rate and dimensionality, allowing the decoder to operate across different (r, d) combinations.

Full architectural specifications, attention masks, optimization settings, flow-matching formulation, vocoders, and training details are given in Appendix A.

4 Experiments

Laconic provides independent control over token rate r and token dimensionality d within a single encoder. We use this control to study how speech information changes as representation budget rd is redistributed between temporal resolution and dimensionality. Our experiments ask three progressively stronger questions. Q1: can token rate and dimensionality substitute for one another? Q2: is there one rate–dimension frontier for all speech information? Q3: can speech be reconstructed and generated from these compressed representations? Throughout the experiments, unless otherwise stated, every Laconic operating point below is obtained from the same encoder.

4.1 Can token rate and dimensionality substitute for one another?

We begin with linguistic information and ask whether representation capacity spent along one axis can compensate for capacity removed from the other. Our diagnostic is a 44M-parameter ASR model trained from scratch on frozen representations. We train on 460 h of LibriSpeech and evaluate on test-clean utterances. The ASR architecture and training procedure are fixed across conditions; only the input representation changes.

Speech features seems to be easier to compress in time than in dimensionality. Figure 3(a) motivates the two-dimensional view using simple transformations of WavLM. At the same budget ($B = 51,200$), the original WavLM shape has better performance than a stack than reduces the token rate. At a more lower budget ($B = 3,200$) reducing the token rate (through pooling) seems to be much better than reducing the dimension (through PCA). This asymmetry does not imply that low-dimensional representations are inherently inadequate. We show through Laconic that the resulting representations lower dimension can also reach competitive performances. The relevant question is therefore not only how much representation budget remains, but *how* that budget is allocated between token rate and dimensionality.

Laconic exposes an exchange regime. Figure 3(b) addresses this question directly by sweeping $r \in \{6.25, 5, 4.17, 3.13, 2, 1\}$ tokens/s and $d \in \{4, 8, 16, 32, 64, 128, 256, 512\}$. At high token rates, linguistic information needs few dimensions, and vice versa. For example, at 6.25 tokens/s, $d = 16$

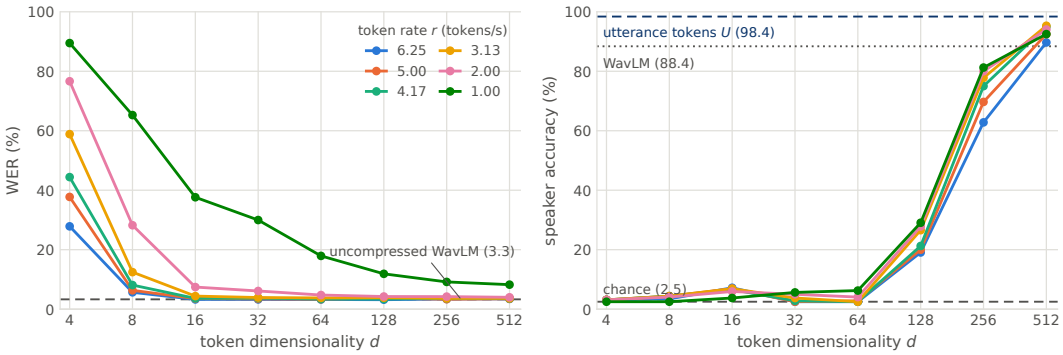


Figure 4: Linguistic content and speaker identity against token dimensionality d, for each token rate r. (a) ASR WER of the 44M-parameter ASR model on LibriSpeech test-clean (4–10 s). (b) Accuracy of a 40-way linear speaker probe on the utterance mean of the local tokens, trained on each speaker's shortest utterances and evaluated on their longest (chance: 2.5%). Horizontal lines mark the five utterance tokens U (98.4%) and mean-pooled uncompressed WavLM (88.4%).

already reaches 3.3% WER, and increasing dimensionality to 512 gives no further improvement. The same 16 dimensions give 3.5% WER at 5 tokens/s and 3.6% at 4.17 tokens/s.

We call the boundary of the operating points that satisfy a fixed performance criterion the *rate–dimension frontier*. The solid contour in Figure 3(b) marks representations within 20% relative WER of the uncompressed WavLM baseline, and the dashed contour marks those within 50%. Over approximately 3–6 tokens/s, the frontier moves toward higher dimensionality as token rate decreases: temporal resolution and dimensionality have a partial trade-off.

The trade-off has a temporal floor. This trade-off eventually breaks down. At 1 token/s, increasing dimensionality from 16 to 512 improves WER from 38% to 8.2%, but even the highest-dimensional representation remains far from the WavLM baseline.

The frontier therefore does not follow a constant-rd law. The representation budget rd counts the scalar values emitted per second, but does not by itself determine linguistic capacity; how those values are distributed between time and dimensionality matters.

Answer to Q1: Partially. Token rate and dimensionality are exchangeable over a broad regime: as token rate decreases, higher dimensionality can preserve linguistic information. The exchange has a temporal floor, however. Below a sufficient token rate (mainly at 1 tokens/sec), increasing dimensionality does not fully recover the lost information.

4.2 Is there one rate–dimension frontier for all speech information?

Q1 established a rate–dimension frontier for linguistic content. Speech representations, however, should ideally encode more than text. Does other information follow the same trend?

We contrast linguistic content with speaker identity. We measure linguistic information with the same ASR model as in Section 4.1. For speaker identity, we train a 40-way linear classifier on the utterance mean of the local tokens L. To test whether the probe generalizes across utterance duration, we train it on each speaker's shortest utterances and evaluate it on their longest; chance accuracy is 2.5%. We evaluate both probes at the same six token rates and eight token dimensionalities.

Linguistic information saturates at low dimensionality. Figure 4(a) shows ASR WER as dimensionality increases. At high and moderate token rates, linguistic information saturates quickly. Throughout the exchange regime, linguistic content therefore occupies a low-dimensional prefix, and dimensions beyond that prefix add little linguistic benefit.

Speaker identity follows a different dimensionality profile. Figure 4(b) shows a different pattern for speaker identity. At all token rates, speaker accuracy remains low through the first 64 dimensions and then rises sharply. Over the same range, from $d = 32$ to $d = 512$, ASR WER changes only from 3.9% to 3.6%: a $16\times$ increase in dimensionality barely changes linguistic performance but recovers

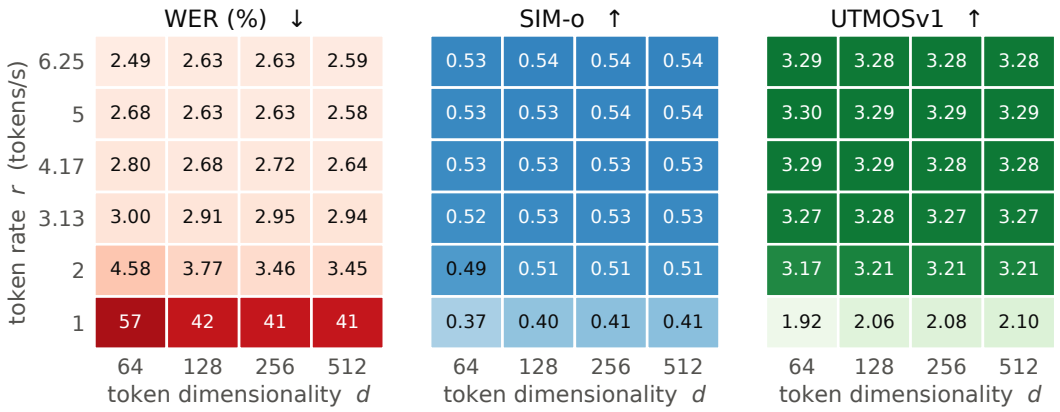


Figure 5: Speech reconstruction across the rate–dimension space. Reconstruction quality on Seed-TTS EN from LACONIC representations as a function of token rate r and token dimensionality d. Reconstruction requires only modest token dimensionality: $d = 64$ is slightly worse, while increasing dimensionality beyond $d = 128$ gives little additional benefit. Performance is similarly stable across token rates from 6.25 down to 3.13 tokens/s, with a sharp collapse at 1 token/s. Speaker similarity (SIM-o) and perceptual quality (UTMOSv1) show the same broad regime of stability above the lowest token rates.

nearly all the speaker identity available to the local representation. In other words, the linguistic performance saturates much earlier in dimensionality than speaker identity does.

Utterance-level information need not occupy the local stream. The utterance tokens U provide a second route for information that does not require fine temporal localization. As the horizontal reference in Figure 4(b) shows, the five utterance tokens alone reach 98.4% speaker accuracy, above both mean-pooled uncompressed WavLM (88.4%) and the full $d = 512$ local representation at 3.13 tokens/s (95.3%). This suggests a complementary allocation. Linguistic content needs a temporally localized sequence but can be represented at low dimensionality. Speaker identity needs many local dimensions, but because it is largely utterance-level, it can instead be carried by a small, fixed-cost global pathway rather than repeated throughout the local sequence. Evaluation on standard downstream understanding benchmarks SiTok/DASB Wang et al. (2026); Mousavi et al. (2024) support this interpretation: the utterance tokens alone remain effective for speaker verification, emotion recognition, and keyword spotting (Appendix H), and English–German speech translation, where a $d = 32$ prefix matches pooled WavLM at $d = 1024$ (Appendix H.3), yielding results sometimes on-par with those with state-of-the-art representations. The equal success on speech translation on CoVoST2 shows that our features are general and beyond just ASR and/or audiobooks speech.

This ordering is deliberately induced, not a property of WavLM: removing the nested text objective raises WER at 3.13 tokens/s from 3.91% to 42.75% at $d = 32$, while barely affecting $d = 512$ (Appendix D.1).

Answer to Q2: No. Different information has different rate–dimension trends. Linguistic content is constrained mainly by token rate and saturates at low dimensionality. In LACONIC's local tokens, speaker identity is constrained mainly by dimensionality and continues to improve through much larger prefixes; it can also be carried compactly by the utterance tokens. There is therefore no single rate–dimension frontier for "speech information."

4.3 Q3: Can speech be reconstructed and generated from these compressed representations?

The previous questions considered what information can be read from LACONIC representations. A compressed representation, however, may preserve enough information for a discriminative probe while discarding structure needed to recover speech itself. We therefore move to our final question: does the representation rate–dimension tradeoff impact speech reconstruction and generation? We study two settings. First, we decode LACONIC representations back to speech and compare them

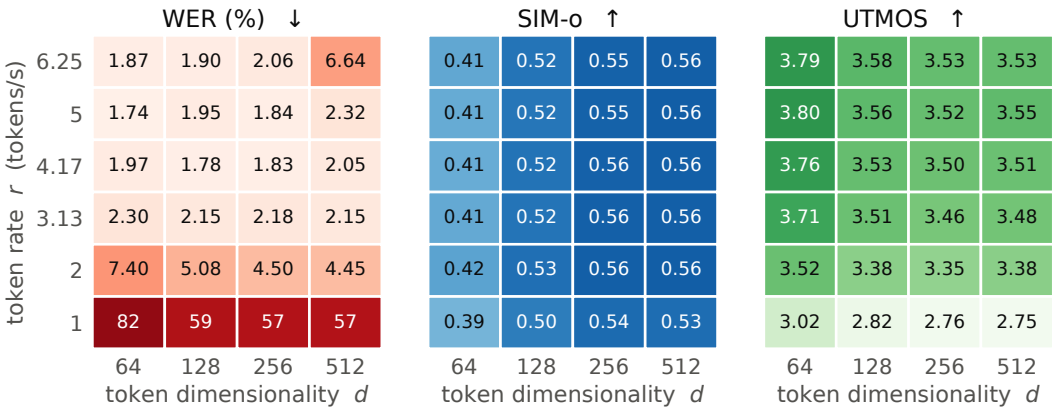


Figure 6: Representation shape determines what is easy to generate. Zero-shot TTS on Seed-TTS EN across token rate r and token dimensionality d, with one 75M-parameter generator per operating point. Left: WER remains low over a broad exchange regime, rises at 2 tokens/s, and collapses at 1 token/s; at high rates, lower-dimensional tokens are easier to generate, whereas higher dimensionality helps as token rate decreases. Middle: speaker similarity (SIM-o) depends mainly on dimensionality and little on token rate. Right: UTMOS generally favors lower-dimensional tokens. Content, speaker fidelity, and perceptual quality therefore favor different allocations of representation capacity. Results on LibriSpeech-PC and the E2 TTS suite are in Appendix F.2.

with other low-rate speech representations. Second, we train zero-shot TTS models to generate LACONIC tokens (which can then be decoded to speech) across the full $r \times d$ space.

Speech remains reconstructable through at low token rates. Figure 5 tracks reconstruction metrics as a function token rate. We use two decoders: an ℓ_2 feature decoder that predicts WavLM layer-6 features, rendered by a frozen HiFi-GAN vocoder, and a conditional flow-matching decoder that predicts mel spectrograms, rendered by BigVGAN. Together they test whether the compressed representation retains enough information to recover intelligible speech, speaker identity, and perceptual quality.

Accurate linguistic reconstruction remains possible even with relatively low token rates, down to 2 tokens/s, but rises sharply at 1 token/s. This is true for both Seed-TTS (Fig. 5) and LibriSpeech (Appendix E). This transition mirrors the temporal floor found with the ASR probe in Section 4.1, where higher dimensionality compensated for reduced token rate down to about 2 tokens/s but not at 1 token/s. The failure at very low rates is therefore not only a limitation of a downstream ASR model: the information needed to recover intelligible speech is itself no longer preserved.

Preserving words is not the same as preserving all properties of speech. Speaker similarity degrades earlier than intelligibility: on LibriSpeech, SIM with the ℓ_2 decoder falls from 0.748 at 6.25 tokens/s to 0.701 at 2 tokens/s while WER stays close to that of the original recordings, and the flow-matching decoder raises SIM on the same tokens (0.841 against 0.748 at 6.25 tokens/s) without changing WER. LACONIC therefore remains linguistically reconstructable at rates where some acoustic information has already degraded. This is consistent with Section 4.2: different kinds of speech information do not share a single rate–dimension requirement.

These operating rates are not unique to LACONIC. Figure 5 places LACONIC alongside recent low-rate speech representations, including FlexiCodec, Sylber, SyllableLM, Mimi, and SiTok. These systems differ in objectives, quantization, training data, decoders, and evaluation protocols, so we use them to show the achievable operating regime rather than as a controlled ranking. Useful speech reconstruction is possible at only a few tokens per second; the role of LACONIC in our study is to make the allocation within that low-rate regime explicitly controllable.

Rate and dimension affect generation differently In de-novo generation, increasing token rate lengthens the sequence that an autoregressive model must generate, whereas increasing dimensionality adds capacity to each generated token without adding generation steps. For a generative model, the two forms of capacity are therefore not computationally equivalent.

We investigate this with zero-shot TTS. We train a 75M-parameter causal Transformer that generates LACONIC tokens, conditioned on a phoneme sequence and a 3 s speech prompt, with a rectified-flow head for continuous next-token prediction. To isolate the effect of representation shape, we train a separate generator for each (r, d) operating point, keeping the LACONIC encoder and speech decoder frozen. Figure 6 reports the resulting surface on Seed-TTS EN.

The WER surface reproduces the temporal boundary seen in ASR and reconstruction. Between 6.25 and 3.13 tokens/s, TTS remains intelligible over a broad range of dimensionalities. At 4.17 tokens/s, WER varies only from 1.78% to 2.05% for $d \in \{128, 256, 512\}$, and at 3.13 tokens/s all four dimensionalities give 2.15–2.30%. At 2 tokens/s, however, WER rises to 4.45–7.40%, and at 1 token/s it skyrockets to 57–82%. The temporal floor therefore appears across all the tasks we study: recognition, reconstruction and generation.

Figure 6 also shows a rate–dimension interaction that is invisible when token rate is reported alone. At high rates, increasing dimensionality can make generation harder: at 6.25 tokens/s, WER rises from 1.87% at $d = 64$ to 6.64% at $d = 512$. As token rate decreases, additional dimensionality becomes useful: at 2 tokens/s, WER improves monotonically from 7.40% at $d = 64$ to 4.45% at $d = 512$. The generator's preferred representation thus shifts from many low-dimensional tokens toward fewer higher-dimensional tokens as token rate decreases, the generative counterpart of the exchange in Section 4.1.

Content, speaker identity, and perceptual quality favor different representation shapes. The other two TTS metrics make the information dependence from Section 4.2 visible. Speaker similarity (between the conditioning input and the generated output) depends on dimensionality more than on token rate. At every rate from 2 to 6.25 tokens/s, SIM-o steadily grows as d increases from 64 to 512, whereas changing token rate over this range has little effect. This mirrors the speaker probe in Figure 4: in LACONIC's local tokens, speaker information needs many dimensions but is comparatively insensitive to token rate once enough temporal observations are present.

Perceptual quality follows yet another profile. UTMOS generally favors lower-dimensional representations: at 6.25 tokens/s it decreases from 3.79 at $d = 64$ to 3.53 at $d = 512$, and at 3.13 tokens/s from 3.71 to 3.48. One possible explanation, which we have not tested, is that higher-dimensional targets preserve more information but present a harder conditional distribution for a finite-capacity generator. We therefore do not treat dimensionality as "more is better": additional capacity helps only when the downstream model can use or generate it.

Taken together, the three surfaces show why a single scalar measure of representation size is insufficient for generation. Low-dimensional representations favor intelligibility and perceptual quality, while speaker fidelity continues to benefit from additional dimensions. Lower token rate shortens generation, but eventually crosses a boundary that higher per-token dimensionality does not repair. The preferred operating point therefore depends on both the information of interest and the model consuming the representation.

Answer to Q3: Yes. The rate–dimension geometry persists when speech must be reconstructed and generated. The same temporal floor appears in probing, reconstruction, and zero-shot TTS. Above this floor, a lower token rate can be compensated by allocating more capacity to each token, but different properties of the generated speech favor different allocations: linguistic content remains accessible at low dimensionality, speaker fidelity benefits from substantially higher dimensionality, and very low temporal resolution is not recovered by increasing dimensionality alone.

5 CONCLUSION

We introduced LACONIC, a single encoder whose token rate and nested token dimensionality can be set independently, and used it to map how speech information changes as capacity moves between token rate and dimensionality. Token rate and dimensionality trade off over roughly 3–6 tokens per second but hit a temporal floor at 1 token per second, and linguistic content, speaker identity, and generation quality each favor different regions of this space. Speech representations should therefore be characterized by their shape and not just their token rate.

AI USE STATEMENT

In this work, we used generative AI tools to aid for polish writing and coding assistant.

REFERENCES

A. Baade, P. Peng, and D. Harwath. SyllableLM: Learning coarse semantic units for speech language models. In *Proc. International Conference on Learning Representations (ICLR)*, 2025.

M. Baas, B. van Niekerk, and H. Kamper. Voice conversion with just nearest neighbors. In *Proc. Interspeech*, 2023.

Kaito Baba, Wataru Nakata, Yuki Saito, and Hiroshi Saruwatari. The t05 system for the VoiceMOS Challenge 2024: Transfer learning from deep image classifier to naturalness MOS prediction of high-quality synthetic speech. In *IEEE Spoken Language Technology Workshop (SLT)*, pp. 818–824, 2024. doi: 10.1109/SLT61566.2024.10832315.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Saurabhchand Bhati, Samuel Thomas, Hilde Kuehne, Rogerio Feris, and James Glass. Towards audio token compression in large audio language models. *arXiv preprint arXiv:2511.20973*, 2025.

C. Busso, M. Bulut, C.-C. Lee, et al. IEMOCAP: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335–359, 2008.

Edresson Casanova, Ryan Langman, Paarth Neekhara, Shehzeen Hussain, Jason Li, Subhankar Ghosh, Ante Jukić, and Sang-Gil Lee. Low frame-rate speech codec: a codec designed for fast high-quality speech llm training and inference. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

S. Chen, C. Wang, Z. Chen, et al. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

C. J. Cho, N. Lee, A. Gupta, D. Agarwal, E. Chen, A. W. Black, and G. K. Anumanchipalli. Sylber: Syllabic embedding representation of speech from raw audio. In *Proc. International Conference on Learning Representations (ICLR)*, 2025.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxCeleb2: Deep Speaker Recognition. In *Interspeech 2018*, pp. 1086–1090, 2018. doi: 10.21437/Interspeech.2018-1929.

A. Défossez, J. Copet, G. Synnaeve, and Y. Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023.

A. Défossez, L. Mazaré, M. Orsini, A. Royer, P. Pérez, H. Jégou, E. Grave, and N. Zeghidour. Moshi: A speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Luca Della Libera, Francesco Paissan, Cem Subakan, and Mirco Ravanelli. Focalcodec: Low-bitrate speech coding via focal modulation networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

Sang gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. BigVGAN: A universal neural vocoder with large-scale training. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=iTtGCMDEzS_.

Aaron Grattafiori, Abhimanyu Dubey, and Abhinav Jauhri et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing: System demonstrations*, pp. 66–71, 2018.

R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar. High-fidelity audio compression with improved RVQGAN. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

A. Kusupati, G. Bhatt, A. Rege, et al. Matryoshka representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

J. Li et al. FlexiCodec: A dynamic neural audio codec for low frame rates. *arXiv preprint arXiv:2510.00981*, 2025.

Luca Della Libera, Cem Subakan, and Mirco Ravanelli. Zipcodec: Ultra-low-frame-rate streaming speech coding. *arXiv preprint arXiv:2609.11642*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Yichen Lu, Jiaqi Song, Chao-Han Huck Yang, and Shinji Watanabe. FastAdaSP: Multitask-adapted efficient inference for large speech language model. In *Proc. EMNLP: Industry Track*, 2024.

Y. Luo, R. Gu, and J. Yao. Stable autoregressive speech generation with low-frame-rate high-dimensional continuous tokens. *arXiv preprint arXiv:2607.29363*, 2026.

Pooneh Mousavi, Jarod Duret, Darius Petermann, Artem Ploujnikov, Luca Della Libera, Anastasia Kuznetsova, Cem Subakan, and Mirco Ravanelli. Dasb-discrete audio and speech benchmark. *arXiv preprint arXiv:2406.14294*, 2024.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

Ankita Pasad, Bowen Shi, and Karen Livescu. Comparative layer-wise analysis of self-supervised speech models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Matt Post. A call for clarity in reporting BLEU scores. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319/.

A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *Proc. International Conference on Machine Learning (ICML)*, 2023.

O. Rippel, M. Gelbart, and R. Adams. Learning ordered representations with nested dropout. In *Proc. International Conference on Machine Learning (ICML)*, 2014.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *arXiv preprint arXiv:2204.02152*, 2022.

L. Tseng et al. TASTE: Text-aligned speech tokenization and embedding for spoken language modeling. *arXiv preprint arXiv:2504.07053*, 2025.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. CoVoST 2 and Massively Multilingual Speech Translation. In *Interspeech 2021*, pp. 2247–2251, 2021. doi: 10.21437/Interspeech.2021-2027.

Y. Wang, Z. Tang, Y. Wang, et al. Scaling speech tokenizers with diffusion autoencoders. *arXiv preprint arXiv:2602.06602*, 2026.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition, 2018. URL https://arxiv.org/abs/1804.03209.

Haibin Wu, Ho-Lam Chung, Yi-Cheng Lin, Yuan-Kuei Wu, Xuanjun Chen, Yu-Chi Pai, Hsiu-Hsuan Wang, Kai-Wei Chang, Alexander H. Liu, and Hung-yi Lee. Codec-SUPERB: An in-depth analysis of sound codec models. In *Findings of ACL*, 2024.

Bajian Xiang, Tingwei Guo, Xuan Chen, and Yang Han. Do we need distinct representations for every speech token? unveiling and exploiting redundancy in large speech language models. In *Findings of ACL*, 2026.

Dongchao Yang, Songxiang Liu, Haohan Guo, Jiankun Zhao, Yuanyuan Wang, Helin Wang, Zeqian Ju, Xubo Liu, Xueyuan Chen, Xu Tan, Xixin Wu, and Helen M. Meng. ALMTokenizer: A low-bitrate and semantic-rich audio codec tokenizer for audio language modeling. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70850–70872. PMLR, 13–19 Jul 2025a. URL https://proceedings.mlr.press/v267/yang25q.html.

Shu-Wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. SUPERB: speech processing universal performance benchmark. In *Proc. Interspeech*, 2021.

Xusheng Yang, Long Zhou, Wenfu Wang, Kai Hu, Shulin Feng, Chenxing Li, Meng Yu, Dong Yu, and Yuexian Zou. U-codec: Ultra low frame-rate neural speech codec for fast high-fidelity speech generation. *arXiv preprint arXiv:2510.16718*, 2025b.

N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30: 495–507, 2021.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In *Interspeech 2019*, pp. 1526–1530, 2019. doi: 10.21437/Interspeech.2019-2441.

X. Zhang, D. Zhang, S. Li, Y. Zhou, and X. Qiu. SpeechTokenizer: Unified speech tokenizer for speech language models. In *Proc. International Conference on Learning Representations (ICLR)*, 2024.

A LACONIC IMPLEMENTATION DETAILS

A.1 REPRESENTATION AND NOTATION

LACONIC maps a sequence of frame-level speech features to two outputs: a fixed set of utterance tokens U and a variable-length sequence of local tokens L. We use $n_u = 5$ utterance tokens and maximum local-token dimensionality $d_{\max} = 512$. Writing $x \in \mathbb{R}^{T\times 1024}$ for the input feature sequence of an utterance, the encoder $\mathcal{E}$ takes x and a compression rate R and returns

$$(U, L) = \mathcal{E}(x, R), \qquad U \in \mathbb{R}^{n_u \times d_{\max}}, \qquad L \in \mathbb{R}^{\lceil T/R \rceil \times d_{\max}}. \tag{1}$$

The local representation has two independently controllable axes.

Token rate. Each local token summarizes a group of R input frames, that is $R/50$ seconds of speech. Since the input representation is sampled at 50 Hz, the resulting token rate is $r = 50/R$ tokens per second. We train with $R \in \mathcal{R} = \{8, 10, 12, 16, 25, 50\}$, corresponding to $r \in \{6.25, 5, 4.17, 3.13, 2, 1\}$ tokens per second. A single R is drawn uniformly for each training example, so one set of encoder weights supports all six rates.

Token dimensionality. Token dimensionality is controlled by reading a prefix of each already-encoded local token. For $d \leq d_{\max}$ we write $L^{(d)} = L_{:,:d}$. Selecting d is a slice of an already-encoded tensor, not a second encoding, so it requires neither re-encoding the utterance nor retraining. This construction follows the general principle of ordered or Matryoshka representations (Rippel et al., 2014; Kusupati et al., 2022).

Representation budget. Ignoring the fixed-cost utterance tokens, an operating point (r, d) emits $B = rd$ local scalar values per second. We call B the *representation budget*; it is the same quantity as the numbers per second used in Section 1, where the uncompressed input costs $B = 50 \times 1024 = 51{,}200$. As the main-paper experiments show, B describes the size of the local representation but does not by itself determine what information that representation preserves.

A.2 Encoder

Input features. Rather than encoding the waveform, we start from features produced by a frozen self-supervised encoder, which speeds up training considerably. Layers of a self-supervised speech model differ in what they encode (Pasad et al., 2023), so rather than commit to one layer we layer-normalize each of the 24 Transformer layers of WavLM-Large (Chen et al., 2022) independently and average them, giving the 1024-dimensional frame representation x at 50 Hz. Layer 6 of the same forward pass gives the reconstruction target y. Appendix C.1 shows that the conclusions do not hinge on this choice.

Token layout. A Transformer encoder reads the sequence

$$\left[u_1, \ldots, u_{n_u}, \quad x_1^1, \ldots, x_R^1, \; l_1, \quad x_1^2, \ldots, x_R^2, \; l_2, \; \ldots\right]$$

where the u_i are learned utterance-level queries and one learned local query l_i follows the i-th group of R input frames. The local queries produce L and the utterance queries produce U.

Attention pattern. Attention is chunk-causal. Frames attend bidirectionally within their own group and causally to earlier groups; each local query attends to its own group and to everything before it; the utterance queries attend to all frames and causally to one another. The utterance queries attend globally and are intended for information that does not require local temporal alignment.

Architecture. The encoder is an 8-layer Transformer with model dimensionality 512, 8 attention heads, feed-forward dimensionality 2048 and dropout 0.1, for approximately 25.2 M parameters.

A.3 Training objectives

LACONIC is trained with three objectives. Reconstruction encourages the representation to retain information sufficient to recover the speech signal; text prediction deliberately concentrates linguistic information in low-dimensional prefixes; and an utterance-level objective provides a fixed-cost pathway for information that need not be repeated through the local stream.

Nested reconstruction. Let $y \in \mathbb{R}^{T \times 1024}$ denote the WavLM layer-6 sequence, normalized per dimension with global statistics. A Transformer decoder $\mathcal{D}$ reconstructs y from the utterance tokens and a sampled dimensional prefix, with mask queries at the frame positions so that the decoder output matches the 50 Hz target:

$$\mathcal{L}_{\text{rec}} = \left\| \mathcal{D}(U, L^{(d)}) - y \right\|_2^2, \qquad d \sim \mathcal{D}_{\text{rec}} = \{64, 128, 256, 512\}, \tag{2}$$

with d drawn at every step and unused suffix coordinates zero-filled before the tokens reach the decoder. We keep the target in feature space, and in this particular feature space, because layer 6 is

known to be the best layer for speaker-preserving synthesis and has an off-the-shelf HiFi-GAN that converts its features to speech (Baas et al., 2023); no vocoder therefore has to be trained alongside LACONIC. Sampling d forces every prefix in $\mathcal{D}_{\text{rec}}$ to remain reconstructable on its own rather than only the full token. We experimented with reconstruction prefixes below $d = 64$, but these produced largely unintelligible speech, so we use $d \geq 64$ for the reconstruction objective.

Linguistic ordering. Reconstruction alone does not determine *which* information should occupy the earliest dimensions. We therefore train a small causal prefix language model to predict the transcript, tokenized into SentencePiece units (Kudo & Richardson, 2018), from the local tokens alone:

$$\mathcal{L}_{\text{txt}} = -\log p\Big(s \mid L^{(d)}\Big), \qquad d \sim \mathcal{D}_{\text{txt}} = \{8, 16, 32, 64, 128, 256, 512\}. \tag{3}$$

The prefix $L^{(d)}$ is followed by the text tokens, attention is causal, and cross-entropy is taken on the text positions only. The text decoder receives only L, never U. Because the smallest prefixes must already support the prediction, this loss explicitly encourages linguistic content to occupy the earliest coordinates.

Utterance-level objective. A linear head P predicts the time average $\bar{y}$ of the layer-6 target from the utterance tokens,

$$\mathcal{L}_{\text{utt}} = \big\|P(U) - \bar{y}\big\|_2^2, \tag{4}$$

and nested dropout (Rippel et al., 2014) keeps a randomly selected prefix $u_1, \ldots, u_m$ of the utterance tokens at every step, so that the utterance-level representation is itself ordered. The loss focuses U on something global, such as speaker identity.

Total objective. The three terms combine as

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_u \mathcal{L}_{\text{utt}} + \lambda_c \mathcal{L}_{\text{txt}}, \qquad \lambda_u = 1, \quad \lambda_c = 0.3. \tag{5}$$

The dimensional ordering observed in LACONIC should therefore *not* be read as a naturally occurring ordering of WavLM coordinates. It is deliberately induced by the nested objectives. Appendix D studies which parts of the ordering depend on which objective.

A.4 SPEECH DECODERS

Feature decoder. Our first speech decoder is the reconstruction decoder $\mathcal{D}$ used during LACONIC training. It predicts WavLM layer-6 features from $[U; L^{(d)}]$, after which the released HiFi-GAN of Baas et al. (2023) converts the features to 16 kHz audio. Neither WavLM nor HiFi-GAN is fine-tuned.

Flow-matching decoder. The mean squared error of Eq. equation 2 makes $\mathcal{D}$ return the conditional mean of the feature trajectories consistent with the tokens; the resulting speech is intelligible but audibly over-smoothed, and both the averaging and the choice of target limit what the tokens can sound like. We therefore train a second decoder that samples rather than averages, and that targets mel spectrograms directly. With the encoder frozen, let z_1 be the normalized log-mel spectrogram and $z_0 \sim \mathcal{N}(0, I)$. For $t \in [0, 1]$ we form the interpolant $z_t = (1-t)z_0 + tz_1$ and train g_θ to predict the velocity of the path,

$$\min_\theta \; \mathbb{E}_{t,\, z_0,\, z_1} \big\| \, g_\theta\Big(z_t,\, t,\, U,\, L^{(d)},\, R,\, d\Big) - (z_1 - z_0) \, \big\|_2^2, \tag{6}$$

which is conditional flow matching with the linear interpolant (Lipman et al., 2023). At inference we draw z_0, integrate g_θ from t=0 to t=1, and map the resulting mel spectrogram to a waveform with a pretrained BigVGAN-v2 vocoder (gil Lee et al., 2023). Both R and d are sampled during training, so *one* flow decoder is shared across the whole rate–dimension surface rather than one speech decoder being trained per operating point.

A.5 Optimization and training

We train the LACONIC encoder and the reconstruction decoder on the 960 hours of LibriSpeech (Panayotov et al., 2015) and hold out `dev-clean` for validation. WavLM-Large remains frozen throughout. The encoder and the feature decoder are each 8-layer Transformers with model dimensionality 512, 8 heads, feed-forward dimensionality 2048 and dropout 0.1, with 25.2 M parameters each.

We train the tokenizer for 20 epochs with AdamW at a base learning rate of 1.5×10^{-4}, scaled by the total batch size relative to 256, under a cosine schedule with 5% warmup and a floor of 1×10^{-5}, with weight decay 0.05 and gradient clipping at 1.0. The batch is 48 utterances per GPU on 7 GPUs.

The flow-matching decoder is trained for 120k steps at an effective batch of 16 on random crops of 512 input frames (10.24 s), with AdamW at 2×10^{-4}, 2k warmup steps, cosine decay, and an exponential moving average of the weights at decay 0.9999. We sample the flow time t from a logit-normal distribution rather than uniformly, which concentrates training on intermediate t, and add an auxiliary cosine term between predicted and target velocity at weight 0.1. The conditioning is replaced by a learned null embedding with probability 0.1 during training so that classifier-free guidance is available at sampling time; each experiment states the guidance scale it uses.

Features from $\mathcal{D}$ are rendered by the released HiFi-GAN (Baas et al., 2023) at 16 kHz and mel spectrograms from g_θ by BigVGAN-v2 (gil Lee et al., 2023) at 24 kHz. Neither vocoder is fine-tuned. Wherever the two are compared, all audio is resampled to 16 kHz before scoring.

A.6 Pooling baselines

WavLM-Pool. We construct a learning-free temporal-compression baseline by replacing each local token with the arithmetic mean of the input features over the same window grid that LACONIC uses. The token rate is therefore identical to the corresponding LACONIC representation and nothing is learned. We note, and deliberately do not correct, two further differences. First, the pooled token keeps all 1024 dimensions of x: averaging induces no ordering over dimensions and so offers no dimensionality dial, meaning WavLM-Pool exists at one shape per rate rather than on a grid. Second, it has no utterance tokens at all, $U = \emptyset$.

WavLM-PoolU. The second difference could by itself account for any speaker-verification gap, so we additionally construct WavLM-PoolU, which prepends a single utterance-level token equal to the mean of all the utterance's frames, $u_1 = \frac{1}{T}\sum_t x_t$. This is again pure arithmetic: the substrate stays entirely learning-free and text-free. Both WavLM-Pool and WavLM-PoolU are decoded by their own flow-matching decoder, trained on the frozen pooled tokens under the identical recipe of Appendix A.5.

A.7 Inference-time shape control

Token rate and token dimensionality are controlled at different stages of inference. The token rate r is chosen when the utterance is encoded, because R determines how many local queries – and therefore how many local tokens – are produced. The token dimensionality d is chosen when the resulting representation is consumed, as the slice $L^{(d)} = L_{:,:d}$ of an already-encoded tensor. The same encoded tensor can therefore be read at different dimensionalities without re-encoding or re-training, and different downstream consumers may use different prefixes: an understanding model can read a small linguistic prefix while a synthesis model reads a larger prefix together with U.

B Evaluation protocols

B.1 ASR probe

We evaluate linguistic accessibility with a 44M-parameter encoder–decoder ASR model trained from scratch on frozen representations. All representation comparisons use the same architecture and training procedure; only the input representation changes, so every difference we report is a difference between representations rather than between readers.

Architecture. The reader is a 6-layer Transformer encoder over the frozen representation and a 6-layer Transformer decoder over characters, both with model dimensionality 512, 8 attention heads, feed-forward dimensionality 2048 and dropout 0.1, for $44.4\,\mathrm{M}$ parameters ($18.9\,\mathrm{M}$ encoder, $25.3\,\mathrm{M}$ decoder). A single linear projection maps the input representation to the model width, so the only thing that changes between representations is the shape of that projection; sinusoidal positional encodings are added on both sides, and a learned type embedding distinguishes utterance tokens from local tokens. The decoder emits lowercase characters over a 31-symbol vocabulary (26 letters, space, apostrophe, and the three special symbols).

Training. We train on `train-clean-100` and `train-clean-360` (460 hours) and validate on `dev-clean`. Batches are formed by token budget rather than a fixed count: utterances are length-bucketed within shuffled chunks and a batch closes when $(|b|+1)\cdot\max_i \mathrm{cost}_i$ would exceed $40{,}000$, where cost is the number of tokens plus the number of target characters plus 8, capped at 128 utterances. Optimization uses AdamW ($\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$) with weight decay 0.01 on matrices and none on biases and normalization parameters, peak learning rate 3×10^{-4}, 1k linear warmup steps followed by cosine decay to 5% of peak, gradient clipping at 1.0, and $40{,}000$ steps. Inputs are normalized with a single global statistic: the per-dimension mean over the training set is subtracted and the result divided by one scalar standard deviation. We select the checkpoint with the best word error rate on a fixed 200-utterance dev subset, evaluated every 1000 steps.

Evaluation. We decode greedily. WER is aggregated at the corpus level – total word errors over total reference words, so long utterances carry proportionally more weight – and we report a 95% confidence interval from 1000 bootstrap resamples over utterances. The evaluation set is the LibriSpeech test-clean utterances between 4 and 10 seconds.

B.2 Speaker probe

We measure speaker information with a 40-way linear classifier trained on the temporal mean of the local representation, using 12 utterances per speaker for training and 8 for evaluation. Unless otherwise stated we use a *length-shift* split: for each speaker the probe is trained on that speaker's shortest utterances and evaluated on their longest. Chance accuracy is 2.5%. This construction reduces the possibility that the classifier solves the task through duration-correlated artifacts.

Because the amount of speaker information accessible from a small prefix is sensitive to the precise train/test split, we additionally report a duration-matched evaluation in Appendix D.4. We therefore interpret the absolute accuracy of any particular small prefix cautiously; the robust observation is the *ordering* across dimensionality, not the value at one d.

B.3 Reconstruction evaluation

We evaluate reconstructed speech along three complementary axes: linguistic fidelity with ASR word error rate, speaker preservation with SIM-o, and perceptual quality with UTMOSv1.

Intelligibility. Reconstructions are transcribed with Whisper large-v3 (Radford et al., 2023) in fp16, one utterance per call. On LibriSpeech we decode greedily and score with Whisper's own `EnglishTextNormalizer`, reporting a *corpus* WER: total word errors over total reference words, so long utterances carry proportionally more weight. On Seed-TTS EN we follow that benchmark's own convention instead – the decoder is forced to `<|en|><|transcribe|><|notimestamps|>`, the normalizer strips punctuation other than the apostrophe and lowercases, and the reported number is the *macro* WER, an unweighted mean of per-utterance rates. The two conventions are not interchangeable, and we keep each set's own so that our numbers are comparable with published ones on that set.

Speaker similarity. SIM is the cosine similarity between speaker embeddings from a WavLM-Large encoder with an ECAPA-TDNN speaker-verification head (Chen et al., 2022; Desplanques et al., 2020), the same "WavLM-TDNN" checkpoint used by F5-TTS and the Seed-TTS benchmark, computed on the full waveforms without cropping. The reference differs by set and this matters for interpretation: on LibriSpeech the reference is the *same utterance's original recording*, so the ground

truth scores 1.0 by construction; on Seed-TTS it is the benchmark's *prompt* recording, which is the SIM-o definition, and the ground truth scores 0.734.

Perceptual quality. UTMOSv1 is the `utmos22_strong` learner of the VoiceMOS 2022 challenge (Saeki et al., 2022), taken from the `SpeechMOS` release and evaluated at 16 kHz. We additionally record UTMOSv2 (Baba et al., 2024) in the released score files, using its `fusion_stage3` configuration at a single fold rather than the five-fold ensemble; the figures and tables in this paper report UTMOSv1.

Sample rate. The feature decoder renders at 16 kHz through HiFi-GAN and the flow decoder at 24 kHz through BigVGAN-v2. Every waveform is resampled to 16 kHz with a Kaiser-windowed sinc filter before any metric is computed, so no comparison in this paper is confounded by render rate.

Evaluation sets. LibriSpeech resynthesis uses 1,237 test-clean utterances between 4 and 10 seconds (mean 6.4 s); Seed-TTS EN uses all 1,088 utterances of the English test set.[1]

B.4 Zero-shot TTS evaluation

We evaluate zero-shot TTS on LibriSpeech-PC, Seed-TTS EN and the E2 TTS suite, reporting WER, SIM-o between the generated utterance and the conditioning speaker reference, and UTMOS. The main paper reports Seed-TTS EN; the complete surfaces on all three sets are in Appendix F.

Sets and sizes. LibriSpeech-PC uses the F5-TTS cross-sentence list (1,127 pairs), the E2 TTS suite uses its 1,132 pairs, and Seed-TTS EN uses its 1,088 utterances.

ASR and aggregation. Each set keeps its own published recipe, so our numbers sit on the same scale as the literature for that set. LibriSpeech-PC is transcribed with faster-whisper large-v3 at beam 5 and scored as a macro WER under the F5-TTS normalizer. The E2 suite is transcribed with a HuBERT-Large CTC model (Hsu et al., 2021) decoded greedily and scored as a corpus WER; its reference is the original LibriSpeech transcript rather than the suite's punctuated prompt text, which is what reproduces the published ground-truth values. Seed-TTS EN uses the forced-English Whisper large-v3 decode and macro WER described above.

SIM-o and UTMOS. For generation, SIM-o is always measured against the *prompt* recording, using the same WavLM-TDNN model as for reconstruction. UTMOS is UTMOSv1 at 16 kHz. Because the flow decoder is stochastic, every reported generation number averages three render seeds.

C Validating the rate–dimension geometry

A skeptical reading of the main paper is that the rate–dimension frontier is an artifact of our particular training procedure rather than a property of the representation problem. This section addresses that reading directly, with three controls: the choice of input feature, dedicated-rate encoders, and dedicated-dimensionality encoders.

C.1 Choice of encoder input

LACONIC uses the layer-normalized mean of all 24 WavLM-Large layers as its input. We test whether the behavior depends strongly on this choice by training otherwise matched models on five alternative WavLM inputs; Table 1 reports resynthesis WER.

The input choices are nearly indistinguishable from 6.25 through 2 tokens per second and separate substantially only at the extreme 1-token/s operating point. At 3.13 tokens per second all five inputs

[1] The ASR probe of Appendix B.1 uses a separately drawn 4–10 s subset of 1,238 utterances that overlaps this one in 1,234 items. The two were selected independently and we report each with its own count rather than silently merging them.

Table 1: Choice of encoder input, d=512. The five inputs agree above 2 tokens per second and separate only at 1. Resynthesis WER (%) on LibriSpeech test-clean 4–10 s (1,237 utterances), ℓ_2 decoder, no nested reconstruction in any row.

input	r (tokens/s)					
	6.25	5.00	4.17	3.13	2.00	1.00
layer 6	2.26	2.33	2.22	2.48	2.56	8.47
layer 6, layer-normalized	2.28	2.26	2.31	2.35	2.48	5.46
layer 6 + layer 24	2.41	2.41	2.28	2.35	2.55	10.29
layer 6‖ layer 24	2.40	2.34	2.39	2.26	2.39	9.09
LN-mean over 24 layers (ours)	2.33	2.32	2.24	2.28	2.46	18.23

Table 2: Training one encoder over six rates costs nothing against encoders trained at a single rate. Word error rate from the 44M ASR encoder (test-clean 4–10 s, 1,238 utterances, ± 0.3); single seed.

tok/s	d	one encoder	single-rate	Δ
6.25	512	3.46	3.48	$+0.02$
	32	3.27	3.44	$+0.17$
3.13	512	3.63	3.59	-0.04
	32	3.91	3.56	-0.35
2	512	3.99	4.21	$+0.22$
	32	6.11	5.03	-1.08

obtain 2.26–2.48% WER, whereas at 1 token per second performance ranges from 5.46% to 18.23%. We therefore do *not* interpret behavior at 1 token per second as independent of the underlying feature choice. The exchange regime studied in the main paper, however, is robust to these alternatives.

C.2 Does one multi-rate encoder distort the frontier?

A central convenience of LACONIC is that one encoder supports all six token rates. This raises the possibility that the observed frontier is an artifact of multi-rate training. We therefore train dedicated encoders at individual token rates and compare them against the corresponding reads of the joint encoder (Table 2).

The two training strategies are closely matched through the useful operating regime. At full dimensionality the differences at 6.25, 3.13 and 2 tokens per second are $+0.02$, -0.04 and $+0.22$ WER respectively. The largest discrepancy is at the most compressed tested combination, r=2 with d=32, where the dedicated model improves WER from 6.11% to 5.03%. Multi-rate training is therefore not cost-free at every extreme operating point, but the main rate–dimension regime does not arise simply from forcing one encoder to serve several rates.

C.3 Does nested dimensionality distort the frontier?

We next ask whether reading a dimensional prefix from one nested representation behaves like training a representation specifically at that dimensionality. For each d we train a model whose heads are permanently pinned to d dimensions and compare it against the d-dimensional prefix of the nested encoder (Table 3).

Agreement is close throughout most of the useful regime: for $d \geq 64$ and $r > 1$, pinned and nested representations generally differ by a few tenths of a WER point. Differences grow under the extreme 1-token/s compression, where the dedicated models can outperform the nested reads substantially.

This control supports a deliberately qualified interpretation. Nested training is a practical way to expose many dimensionalities from one representation and closely approximates dedicated training through the regime the main paper emphasizes. It should not be read as imposing zero cost at arbitrary compression.

Table 3: Models trained with every head pinned to d dimensions, against the d-dimensional read of the one nested encoder. Above 1 token per second the nested read matches the dedicated model to within about the ± 0.3 measurement interval; at 1 token per second the dedicated model pulls ahead. Resynthesis WER (%) as in Table 1; the nested rows are the same checkpoint the paper ships, the 512-dimensional pinned row is the model of Table 1.

d	model	r (tokens/s)					
		6.25	5.00	4.17	3.13	2.00	1.00
16	pinned	2.83	3.03	3.96	7.98	43.29	88.35
32	pinned	2.33	2.40	2.48	2.86	8.45	69.70
64	pinned	2.30	2.35	2.27	2.46	2.88	47.57
	nested	2.39	2.26	2.40	2.48	3.25	43.64
128	pinned	2.24	2.27	2.24	2.34	2.51	30.95
	nested	2.29	2.30	2.26	2.36	2.64	30.21
256	pinned	2.32	2.24	2.21	2.42	2.54	19.92
	nested	2.27	2.28	2.29	2.41	2.61	28.83
512	pinned	2.33	2.32	2.24	2.28	2.46	18.23
	nested	2.33	2.35	2.35	2.21	2.66	29.55

C.4 Temporal versus dimensional compression of pretrained features

Figure 3(a) in the main paper gives a compact diagnostic showing that WavLM features tolerate temporal compression far better than naive dimensional compression. Table 4 reports the expanded set of transformations: lossless stacking, temporal pooling, and variance-ordered PCA applied along either axis, together with the learned LACONIC rows and the 40-way speaker probe.

Two rows carry most of the argument. Lossless stacking concatenates the sixteen frames a token replaces and so discards nothing; it costs the same 51,200 values per second as the uncompressed input and stays within 0.6 WER of it, which establishes that the reader is not merely sensitive to frame rate. The $50\,\text{Hz} \times 64$ row spends exactly the same 3,200 values per second as temporal pooling but spends them on dimensions instead of time, and costs $2.0\times$ the error.

The comparison should not be read as evidence that low-dimensional speech representations are intrinsically inferior. PCA preserves directions of high variance in the original coordinate system; it does not learn to reorganize task-relevant information into a low-dimensional prefix. LACONIC demonstrates the distinction directly: at r=3.13, d=32 it reaches 3.91% WER using only 100 local scalar values per second, where pooling followed by PCA to the same 32 dimensions gives 17.99%.

C.5 A cross-representation diagnostic on Whisper

We repeated the compression diagnostic on Whisper-large-v3 encoder features, whose encoder also emits 50 Hz frames (320 input samples per frame, exactly WavLM's hop), so the two ladders are comparable rung for rung (Table 5).

The qualitative asymmetry reproduces, and is wider. Temporal averaging is not merely harmless but mildly helpful, moving WER from 6.12% at 50×1280 to 5.76% at 3.13×1280, whereas reducing the pooled representation to 32 PCA dimensions raises WER to 45.58% – a factor of 7.9, against 4.6 on WavLM. PCA to 32 dimensions on stacked Whisper frames does not converge to a usable model at all.

We treat this only as a cross-representation diagnostic and do not use Whisper to define the main frontier. Its layerwise organization differs from WavLM's: our features are the layer-normalized mean over all layers, the same definition for both backbones, but for Whisper that is not the strongest content summary. Its last layer alone gives 4.22% WER against the LN-mean's 6.12%, while its 40-way speaker accuracy falls from 76.6% to 10.9%, because ASR pretraining concentrates linguistic content in the deepest layers and discards speaker identity there. We keep the LN-mean in both ladders so that the operation being compared, rather than a layer choice, is what differs. The asymmetry

Table 4: The compression ladder, one rung per row, all at 3.13 tokens per second except the uncompressed input. *Numbers/s* is token rate times token dimensionality — the budget a downstream model pays, and the quantity this section spends — and *compr.* is its reduction against the uncompressed input. ASR is word error rate from the 44M ASR encoder (test-clean 4–10 s, 1,238 utterances, ± 0.3); *Spk* is the 40-way linear speaker probe on the utterance mean of that representation (dev-clean, chance 2.5, length-shift split). Lossless stacking concatenates the 16 frames a token replaces and so discards nothing, which is why it costs the same numbers per second as the uncompressed input it rearranges. The $50\,\text{Hz} \times 64$ row spends the same 3,200 values per second as LACONIC-POOL but on dimensions instead of time, and costs $2.0\times$ the error. Dashes are cells we did not measure rather than zeros: the speaker probe was run on LACONIC slots, its utterance tokens, and the WavLM utterance mean only.

Representation	r	d	numbers/s	compr.	ASR (%) ↓	Spk (%) ↑
Uncompressed WavLM	50	1024	51,200	1×	3.33	88.4
no information discarded						
Lossless stacking	3.13	16,384	51,200	1×	3.92	–
LACONIC-POOL (window mean)	3.13	1024	3,200	16×	3.91	–
linear, variance-ordered truncation						
50 Hz frames + PCA to 64	50	64	3,200	16×	7.68	–
LACONIC-POOL + PCA to 512	3.13	512	1,600	32×	4.81	–
LACONIC-POOL + PCA to 128	3.13	128	400	128×	5.99	–
LACONIC-POOL + PCA to 32	3.13	32	100	512×	17.99	–
stacked frames + PCA to 512	3.13	512	1,600	32×	5.68	–
stacked frames + PCA to 32	3.13	32	100	512×	14.17	–
learned						
LACONIC, d=512	3.13	512	1,600	32×	**3.63**	95.3
LACONIC, d=32	3.13	32	100	**512**×	**3.91**	3.8
utterance tokens U	5/utt.	512	–	–	–	**98.4**

Table 5: The compression ladder on **Whisper** features, the twin of Table 4. Whisper-large-v3's encoder emits 50 Hz frames of 1280 channels, the same rate as WavLM, so every rung is directly comparable. All compressed rungs run at 3.13 tokens per second. The time–channel asymmetry reproduces: rearranging or averaging sixteen frames is free (it even helps), while cutting channels in the original feature space is destructive, and more so than on WavLM. The uncompressed row also shows that the all-layer LN-mean is the wrong summary for Whisper — its last layer alone is 1.9 WER better and carries almost no speaker information — because Whisper concentrates linguistic content in its deepest layers rather than its middle ones.

Representation	r	d	numbers/s	compr.	ASR (%) ↓	Spk (%) ↑
Uncompressed Whisper, LN-mean	50	1280	64,000	1×	6.12	76.6
Uncompressed Whisper, layer 32	50	1280	64,000	1×	**4.22**	10.9
no information discarded						
Lossless stacking	3.13	20,480	64,000	1×	6.03	–
Window mean	3.13	1280	4,000	16×	5.76	–
linear, variance-ordered truncation						
window mean + PCA to 512	3.13	512	1,600	40×	6.92	–
window mean + PCA to 128	3.13	128	400	160×	9.24	–
window mean + PCA to 32	3.13	32	100	640×	45.58	–
stacked frames + PCA to 512	3.13	512	1,600	40×	TBD	–
stacked frames + PCA to 32	3.13	32	100	640×	100.76	–

is unaffected by the choice, and layerwise organization in ASR-pretrained versus self-supervised encoders is a separate question from the one this paper asks.

Table 6: The text loss is what makes a narrow token readable; the utterance tokens cost content almost nothing. Word error rate as in Table 2.

ablation	tok/s	d	reference	ablated	Δ
no text loss	6.25	512	3.46	6.55	$+3.09$
	3.13	512	3.63	8.53	$+4.90$
	3.13	32	3.91	42.75	$+38.84$
no utterance tokens	6.25	512	3.46	3.31	-0.15
	3.13	512	3.63	4.53	$+0.90$
	3.13	32	3.91	3.88	-0.03

D What creates the dimensional ordering?

The rate–dimension geometry and the coordinate ordering are distinct phenomena. LACONIC is designed to expose the former by deliberately constructing the latter. This section ablates the objectives responsible for the ordering.

D.1 The text objective creates readable low-dimensional prefixes

Removing the text loss has a large and highly dimensionality-dependent effect (Table 6). At r=6.25, d=512, WER rises from 3.46% to 6.55%; at r=3.13, d=512, from 3.63% to 8.53%. The effect becomes dramatic at r=3.13, d=32, where WER rises from 3.91% to 42.75%.

Reconstruction alone can therefore preserve substantial linguistic information *somewhere* in the representation, but does not place it reliably in the first coordinates. The nested text objective is what makes dimensionality a useful control variable at all, by concentrating linguistic information into the early prefix.

D.2 Utterance tokens do not create the linguistic frontier

Removing the utterance tokens has comparatively little effect on linguistic accessibility. WER is essentially unchanged at r=6.25, d=512 (3.46 $\rightarrow$ 3.31) and at r=3.13, d=32 (3.91 $\rightarrow$ 3.88). The linguistic frontier is therefore not primarily produced by routing linguistic content through U.

D.3 Where does speaker information go?

One might expect that removing the utterance pathway would force speaker information into the local tokens. We do not observe this (Table 7). Under the length-shift split, removing U substantially *reduces* speaker information at high dimensionality: at d=512, speaker accuracy falls from 95.3% to 26.6%. Speaker information is not simply displaced into the local stream when the global pathway is removed; much of it is instead lost under the training objective.

This supports interpreting U as an efficient pathway for information that does not require temporal localization, rather than merely an additional reservoir whose contents would otherwise appear identically in L.

D.4 Speaker ordering under alternative training conditions

The exact amount of speaker information in a small prefix depends on the evaluation split and the training configuration. We therefore repeat the speaker probe under a duration-matched split, across the joint multi-rate encoder, a dedicated-rate encoder and a model trained without the text objective (Table 8).

Across all three models speaker accuracy increases strongly with dimensionality. However, both single-rate training and removal of the text loss let speaker identity become accessible earlier: at d=32, accuracy is 10.6% for the joint encoder against 31.2% for the single-rate model and 33.1% without the text objective. By d=256 all three exceed 96%.

Table 7: Without utterance tokens the speaker is not pushed into the local tokens, it is largely lost. Speaker probe as in Table 8, under the length-shift split, the only ordering measured for this run.

d	reference	no utterance tokens
4	3.1	2.8
8	4.4	2.8
16	6.9	3.8
32	3.8	11.9
64	2.5	14.1
128	26.6	22.8
256	77.8	29.4
512	95.3	26.6
utterance tokens	98.4	–

Table 8: Multi-rate training and the text loss each keep speaker identity out of the content-bearing channels. 40-way linear speaker probe on the mean-pooled local tokens at 3.13 tokens per second, duration-matched split (chance 2.5); single seed.

d	one encoder	single-rate	no text loss
4	5.9	6.2	8.4
8	7.5	5.6	14.1
16	9.1	9.7	22.8
32	10.6	31.2	33.1
64	35.9	52.8	53.4
128	82.2	87.2	87.2
256	96.6	96.9	97.5
512	99.1	98.4	98.4
utterance tokens	98.8	98.8	98.8

We therefore avoid describing the first d dimensions as strictly speaker-free. The robust finding is an *ordering*: linguistic information becomes useful at substantially lower dimensionality than speaker identity, while the precise degree of separation depends on the training objective and the evaluation split.

E Speech reconstruction

E.1 The full rate–dimension surface

Figure 7 evaluates reconstruction on Seed-TTS EN across the complete 6×4 rate–dimension grid with the ℓ_2 decoder. The result is characterized by a broad plateau. From 6.25 through 3.13 tokens per second, changing either axis has comparatively little effect once $d \geq 64$. At 2 tokens per second reconstruction degrades moderately and additional dimensionality helps. At 1 token per second all three metrics deteriorate sharply.

Dimensionality beyond d=128 buys little through most of the viable regime: at 3.13 tokens per second, WER is 2.91, 2.95 and 2.94% for $d = 128, 256, 512$, with SIM-o approximately 0.53 throughout. Reconstructing speech therefore requires more dimensionality than accessing linguistic content with the ASR probe, but stops benefiting from further dimensions once a modest threshold is reached.

E.2 When does learned re-encoding become necessary?

Temporal averaging is a strong control precisely because it reduces token rate without learning a new representation. At moderate rates it is surprisingly effective, which indicates substantial temporal redundancy in the original WavLM sequence. At lower rates, pooling and learned re-encoding separate sharply.

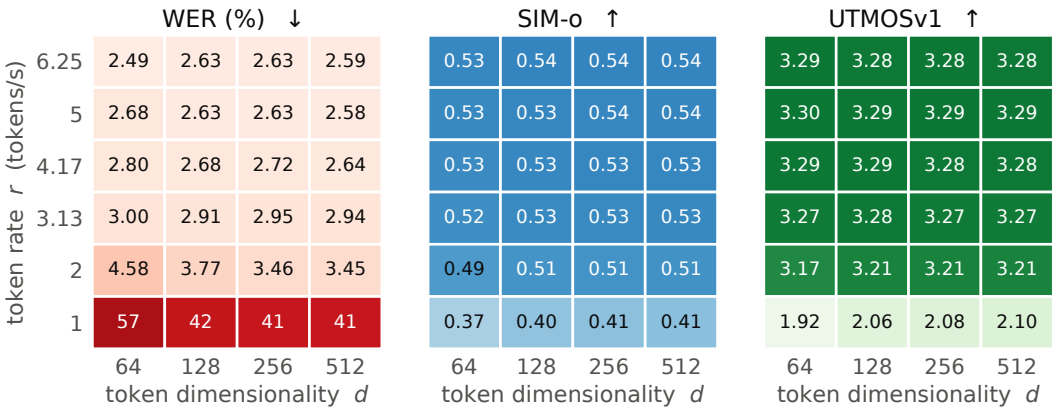


Figure 7: Seed-TTS EN resynthesis over the rate–dimension grid, LACONIC with the ℓ_2 decoder. WER is colour-mapped on a log scale because it spans 2.5 to 57; SIM-o and UTMOSv1 are linear. Ground truth is 2.18% WER, 0.734 SIM-o and 3.52 UTMOSv1. The plateau from 6.25 to 3.13 tokens per second and the collapse at 1 token per second appear in all three metrics simultaneously, which is what identifies the boundary as temporal rather than dimensional.

On LibriSpeech, with identically trained flow-matching decoders, WAVLM-POOL obtains 3.43% WER at 3.13 tokens per second against 2.51% for LACONIC. At 2 tokens per second the gap widens sharply, to 20.63% against 2.98%, and at 1 token per second pooling reaches 91.4% against 34.4%. The low-rate regime therefore cannot be reached indefinitely by discarding redundant frames: information must eventually be reorganized into the surviving tokens.

E.3 RECONSTRUCTION METRICS CAPTURE DIFFERENT INFORMATION

Linguistic intelligibility is not equivalent to acoustic preservation. On LibriSpeech, SIM-o with the feature decoder decreases from 0.748 at 6.25 tokens per second to 0.701 at 2 tokens per second while WER stays close to the original-speech reference. Replacing the feature decoder with flow matching raises SIM from 0.748 to 0.841 at 6.25 tokens per second with no corresponding WER improvement.

These results reinforce the conclusion of Section 4.2: different properties of speech have different representation requirements, and a representation can remain highly intelligible after some speaker or acoustic detail has already become hard to recover.

E.4 COMPARISON WITH PUBLISHED LOW-RATE REPRESENTATIONS

We contextualize LACONIC against published low-rate speech representations including FlexiCodec, Sylber, SyllableLM, Mimi and SiTok.

These comparisons are *not* controlled rankings. The systems differ in representation objective, discrete versus continuous parameterization, training data, reconstruction decoder, vocoder, sampling procedure and in some cases evaluation implementation. We use them to establish the operating regime reached by contemporary low-rate speech representations, not to claim a universal ordering among systems.

Across those differences the comparison supports a conclusion shared by several recent approaches: useful speech reconstruction is possible from representations operating at only a few temporal tokens per second. LACONIC complements this literature by exposing how capacity is allocated between temporal rate and token dimensionality within that regime.

Table 9: Resynthesis on LibriSpeech test-clean (1,237 utterances of 4–10 s) and the Seed-TTS EN test set (1,088 utterances). Each test set is scored with its own standard protocol. LibriSpeech: word-weighted WER from Whisper-large-v3 under its English text normalizer, and SIM between the resynthesis and the original recording. Seed-TTS: the benchmark's macro-averaged WER (`run_wer.py`) and SIM-o against the benchmark prompt, another recording of the same speaker, so ground truth itself reads 0.734. SIM uses ECAPA-TDNN over WavLM-Large features; UTMOS is UTMOSv1. *Hz* is tokens per second: for FlexiCodec the per-utterance target rate (matched within 2%), for Sylber and SyllableLM, whose segmentation is variable, the mean rate measured on that test set. *Dim./CB* is the feature dimension for continuous tokenizers (Sylber, LACONIC) and the number of codebooks for discrete ones. LACONIC uses d=512 and its ℓ_2 decoder; the vocoder ceiling renders ground-truth features through the same vocoder. †Values reported by SiTok, not scored by us.

System	Hz	Dim./CB	WER (%) ↓	SIM ↑	UTMOS ↑
LibriSpeech test-clean					
Ground truth	–	–	2.21	–	4.10
Vocoder ceiling (WavLM-L6 → HiFi-GAN)	–	–	2.35	0.949	4.05
Mimi	12.5	32	2.52	0.937	3.86
SyllableLM	5.77	1	8.01	0.123	3.87
Sylber	3.86	768	3.64	0.394	4.14
FlexiCodec	3.13	24	6.07	0.708	4.01
LACONIC (ℓ_2 decoder)	3.13	512	2.22	0.727	3.73
LACONIC-POOL (ℓ_2 decoder)	3.13	1024	2.88	0.618	3.60
Seed-TTS EN					
Ground truth	–	–	2.18	0.734	3.52
Vocoder ceiling (WavLM-L6 → HiFi-GAN)	–	–	2.28	0.684	3.53
Mimi	12.5	32	2.46	0.696	3.40
SiTok†	12.5	4	2.80	0.660	3.46
SyllableLM	6.29	1	13.14	0.069	3.95
Sylber	3.98	768	6.04	0.256	4.01
FlexiCodec	3.13	24	7.16	0.514	3.68
LACONIC (ℓ_2 decoder)	3.13	512	2.94	0.529	3.27
LACONIC-POOL (ℓ_2 decoder)	3.13	1024	3.92	0.462	3.15

F ZERO-SHOT SPEECH GENERATION

F.1 GENERATOR

To test whether the same geometry matters when the representation itself must be generated, we train autoregressive zero-shot TTS models over frozen LACONIC tokens.

Architecture. The generator is a 12-layer causal pre-LayerNorm Transformer with model dimensionality 512, 8 attention heads, feed-forward dimensionality 2048 and dropout 0.1, totalling 74.7 M parameters (37.8 M backbone, 20.4 M flow head, 16.4 M embeddings).[2] Position is encoded by a learned segment-type embedding plus a per-segment sinusoid, so padding inside one segment never shifts the geometry of another.

Conditioning layout. The model reads a fixed segment order,

$$[\text{STY}]\ p_U\ p_L\ [\text{TXT}]\ \text{phonemes}\ [\text{SPC}]\ t_U\ t_L,$$

where p_U, p_L are the utterance and local tokens of a *different* utterance by the same speaker (the speech prompt), and t_U, t_L are the targets. The five utterance tokens are generated first, then the local

[2]The embedding table is sized for a 32,000-entry vocabulary inherited from the subword frontend, so it accounts for a disproportionate share of the parameter count even though the phoneme inventory is only 66 symbols.

Table 10: Resynthesis on LibriSpeech test-clean (1,237 utterances of 4–10 s) and the Seed-TTS EN test set (1,088 utterances). Each test set is scored with its own standard protocol. LibriSpeech: word-weighted WER from Whisper-large-v3 under its English text normalizer, and SIM between the resynthesis and the original recording. Seed-TTS: the benchmark's macro-averaged WER (`run_wer.py`) and SIM-o against the benchmark prompt, another recording of the same speaker, so ground truth itself reads 0.734. SIM uses ECAPA-TDNN over WavLM-Large features; UTMOS is UTMOSv1. *Hz* is tokens per second: for FlexiCodec the per-utterance target rate (matched within 2%), for Sylber and SyllableLM, whose segmentation is variable, the mean rate measured on that test set. *Dim./CB* is the feature dimension for continuous tokenizers (Sylber, LACONIC) and the number of codebooks for discrete ones. LACONIC uses d=512 and its flow decoder; the vocoder ceiling renders ground-truth features through the same vocoder. †Values reported by SiTok, not scored by us.

System	Hz	Dim./CB	WER (%) ↓	SIM ↑	UTMOS ↑
LibriSpeech test-clean					
Ground truth	–	–	2.21	–	4.10
Vocoder ceiling (mel → BigVGAN)	–	–	2.25	0.980	4.07
Mimi	12.5	32	2.52	0.937	3.86
LACONIC (flow decoder)	6.25	512	2.41	0.841	3.58
LACONIC-POOL (flow decoder)	6.25	1024	2.44	0.867	3.49
SyllableLM	5.77	1	8.01	0.123	3.87
LACONIC (flow decoder)	5.00	512	2.28	0.839	3.62
LACONIC-POOL (flow decoder)	5.00	1024	2.46	0.857	3.52
LACONIC (flow decoder)	4.17	512	2.36	0.836	3.63
LACONIC-POOL (flow decoder)	4.17	1024	2.57	0.848	3.54
Sylber	3.86	768	3.64	0.394	4.14
FlexiCodec	3.13	24	6.07	0.708	4.01
LACONIC (flow decoder)	3.13	512	2.51	0.827	3.65
LACONIC-POOL (flow decoder)	3.13	1024	3.43	0.826	3.53
LACONIC (flow decoder)	2.00	512	2.98	0.808	3.64
LACONIC-POOL (flow decoder)	2.00	1024	20.63	0.749	3.33
LACONIC (flow decoder)	1.00	512	34.42	0.694	3.28
LACONIC-POOL (flow decoder)	1.00	1024	91.40	0.570	2.90
Seed-TTS EN					
Ground truth	–	–	2.18	0.734	3.52
Vocoder ceiling (mel → BigVGAN)	–	–	2.21	0.723	3.52
Mimi	12.5	32	2.46	0.696	3.40
SiTok†	12.5	4	2.80	0.660	3.46
SyllableLM	6.29	1	13.14	0.069	3.95
LACONIC (flow decoder)	6.25	512	2.40	0.617	2.96
LACONIC-POOL (flow decoder)	6.25	1024	2.47	0.633	2.79
LACONIC (flow decoder)	5.00	512	2.53	0.617	2.97
LACONIC-POOL (flow decoder)	5.00	1024	2.49	0.625	2.81
LACONIC (flow decoder)	4.17	512	2.36	0.615	2.96
LACONIC-POOL (flow decoder)	4.17	1024	2.44	0.618	2.81
Sylber	3.98	768	6.04	0.256	4.01
FlexiCodec	3.13	24	7.16	0.514	3.68
LACONIC (flow decoder)	3.13	512	2.48	0.610	2.99
LACONIC-POOL (flow decoder)	3.13	1024	3.21	0.607	2.82
LACONIC (flow decoder)	2.00	512	3.21	0.595	2.96
LACONIC-POOL (flow decoder)	2.00	1024	19.66	0.561	2.67
LACONIC (flow decoder)	1.00	512	43.50	0.528	2.62
LACONIC-POOL (flow decoder)	1.00	1024	104.00	0.454	2.29

tokens. Text and speech tokens enter through separate projections into the same width. For classifier-free guidance, the entire text segment is replaced by a learned null embedding with probability 0.15 during training.

Table 11: Resynthesis on LibriSpeech test-clean (1,237 utterances of 4–10 s) and the Seed-TTS EN test set (1,088 utterances). Each test set is scored with its own standard protocol. LibriSpeech: word-weighted WER from Whisper-large-v3 under its English text normalizer, and SIM between the resynthesis and the original recording. Seed-TTS: the benchmark's macro-averaged WER (`run_wer.py`) and SIM-o against the benchmark prompt, another recording of the same speaker, so ground truth itself reads 0.734. SIM uses ECAPA-TDNN over WavLM-Large features; UTMOS is UTMOSv1. *Hz* is tokens per second: for FlexiCodec the per-utterance target rate (matched within 2%), for Sylber and SyllableLM, whose segmentation is variable, the mean rate measured on that test set. *Dim./CB* is the feature dimension for continuous tokenizers (Sylber, LACONIC) and the number of codebooks for discrete ones. LACONIC uses d=512 and its ℓ_2 decoder; the vocoder ceiling renders ground-truth features through the same vocoder. †Values reported by SiTok, not scored by us.

System	Hz	Dim./CB	WER (%) ↓	SIM ↑	UTMOS ↑
LibriSpeech test-clean					
Ground truth	–	–	2.21	–	4.10
Vocoder ceiling (WavLM-L6 → HiFi-GAN)	–	–	2.35	0.949	4.05
Mimi	12.5	32	2.52	0.937	3.86
LACONIC (ℓ_2 decoder)	6.25	512	2.33	0.748	3.74
LACONIC-POOL (ℓ_2 decoder)	6.25	1024	2.81	0.661	3.69
SyllableLM	5.77	1	8.01	0.123	3.87
LACONIC (ℓ_2 decoder)	5.00	512	2.34	0.742	3.73
LACONIC-POOL (ℓ_2 decoder)	5.00	1024	2.69	0.657	3.68
LACONIC (ℓ_2 decoder)	4.17	512	2.35	0.737	3.74
LACONIC-POOL (ℓ_2 decoder)	4.17	1024	2.30	0.647	3.69
Sylber	3.86	768	3.64	0.394	4.14
FlexiCodec	3.13	24	6.07	0.708	4.01
LACONIC (ℓ_2 decoder)	3.13	512	2.22	0.727	3.73
LACONIC-POOL (ℓ_2 decoder)	3.13	1024	2.88	0.618	3.60
LACONIC (ℓ_2 decoder)	2.00	512	2.67	0.701	3.69
LACONIC-POOL (ℓ_2 decoder)	2.00	1024	13.89	0.498	3.06
LACONIC (ℓ_2 decoder)	1.00	512	29.50	0.559	2.73
LACONIC-POOL (ℓ_2 decoder)	1.00	1024	86.30	0.229	1.43
Seed-TTS EN					
Ground truth	–	–	2.18	0.734	3.52
Vocoder ceiling (WavLM-L6 → HiFi-GAN)	–	–	2.28	0.684	3.53
Mimi	12.5	32	2.46	0.696	3.40
SiTok†	12.5	4	2.80	0.660	3.46
SyllableLM	6.29	1	13.14	0.069	3.95
LACONIC (ℓ_2 decoder)	6.25	512	2.59	0.540	3.28
LACONIC-POOL (ℓ_2 decoder)	6.25	1024	3.90	0.483	3.23
LACONIC (ℓ_2 decoder)	5.00	512	2.58	0.537	3.29
LACONIC-POOL (ℓ_2 decoder)	5.00	1024	4.27	0.481	3.22
LACONIC (ℓ_2 decoder)	4.17	512	2.64	0.535	3.28
LACONIC-POOL (ℓ_2 decoder)	4.17	1024	2.96	0.480	3.26
Sylber	3.98	768	6.04	0.256	4.01
FlexiCodec	3.13	24	7.16	0.514	3.68
LACONIC (ℓ_2 decoder)	3.13	512	2.94	0.529	3.27
LACONIC-POOL (ℓ_2 decoder)	3.13	1024	3.92	0.462	3.15
LACONIC (ℓ_2 decoder)	2.00	512	3.45	0.512	3.21
LACONIC-POOL (ℓ_2 decoder)	2.00	1024	21.94	0.386	2.50
LACONIC (ℓ_2 decoder)	1.00	512	40.86	0.413	2.10
LACONIC-POOL (ℓ_2 decoder)	1.00	1024	105.11	0.196	1.37

Continuous next-token head. Local tokens are continuous, so the next-token distribution is modelled by a rectified-flow head rather than a softmax: a 6-block residual MLP of width 768 with adaLN-Zero modulation conditioned on the backbone state and a sinusoidal embedding of the flow

Table 12: The same frozen tokens rendered by two decoders, LibriSpeech test-clean (4–10 s), d=512. The flow decoder is trained on the same LibriSpeech train-960 audio as the tokenizer, resampled to 24 kHz for its mel target; all audio is scored at 16 kHz. Flow renders use guidance 3.0 and UTMOS is UTMOSv1, matching Tables 10 and 11. The two decoders trade off: the ℓ_2 decoder is slightly more intelligible and scores higher on UTMOS at every rate above 1 token/s, while the flow decoder preserves substantially more speaker identity at every rate. Neither ordering changes across the exchange regime; at 1 token/s both decoders fail.

	WER (%) ↓		UTMOSv1 ↑		SIM ↑	
r (tokens/s)	ℓ_2	flow	ℓ_2	flow	ℓ_2	flow
6.25	**2.33**	2.41	**3.74**	3.58	0.748	**0.841**
5.00	2.34	**2.28**	**3.73**	3.62	0.742	**0.839**
4.17	**2.35**	2.36	**3.74**	3.63	0.737	**0.836**
3.13	**2.22**	2.51	**3.73**	3.65	0.727	**0.827**
2.00	**2.67**	2.98	**3.69**	3.64	0.701	**0.808**
1.00	**29.50**	34.42	2.73	**3.28**	0.559	**0.694**

time. With $x_t = (1-t)\,\epsilon + t\,x$ and target velocity $x - \epsilon$, the head is trained with MSE plus a cosine term at weight 0.1, drawing t from a logit-normal distribution and averaging 4 independent noise/time samples per target token.

Duration. Because the number of local tokens *is* the duration, the generator needs no separate duration model: a single linear head predicts, per slot, whether that slot is the last, trained with per-slot binary cross-entropy at weight 1.0. Generation stops when its probability exceeds 0.5.

Training. We train on all three LibriTTS training splits (Zen et al., 2019): 354,780 utterances, 553.7 hours, 2,311 speakers. Text is converted to IPA phonemes with eSpeak-ng, one identifier per symbol over a 66-symbol inventory, truncated at 512 symbols. The prompt is drawn from another utterance of the same speaker and randomly cropped to between 10% and 100% of its length, which gives a mean prompt of about 3 seconds.

Optimization uses AdamW ($\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$) with weight decay 0.01 applied only to matrices, peak learning rate 4.5×10^{-4}, 4k linear warmup steps followed by cosine decay to 5% of peak, and 150k steps. Batches are formed by token budget rather than a fixed count (26,000 cost units, at most 128 utterances, length-bucketed), gradients are clipped at global norm 1.0, and the forward and backward passes run in bfloat16 autocast with fp32 parameters. We keep an exponential moving average at decay 0.999 and report the EMA weights of the final checkpoint. The LACONIC encoder and the speech decoder are frozen throughout.

Inference. Generation is deterministic: the flow is integrated with 10 Euler steps from an initial state of exactly zero, and the prompt's utterance tokens are copied verbatim into the generation rather than sampled. The generator itself uses *no* classifier-free guidance; the guidance scale of 3.0 quoted throughout the paper belongs to the mel flow-matching decoder that renders the generated tokens, not to the generator. The speech prompt at inference is the first 3 seconds of the reference utterance. Each reported number averages three decoder render seeds.

We train one generator for each (r, d) operating point. This intentionally differs from the single configurable LACONIC encoder: the goal here is not to build one configurable TTS model, but to measure how representation shape affects the difficulty and outcome of generation without confounding operating points inside a single generator.

F.2 COMPLETE GENERATION SURFACES

Figures 8–10 report WER, SIM-o and UTMOS across all tested rates and dimensionalities on LibriSpeech-PC, the E2 TTS suite and Seed-TTS EN. The qualitative structure is highly consistent across evaluation sets.

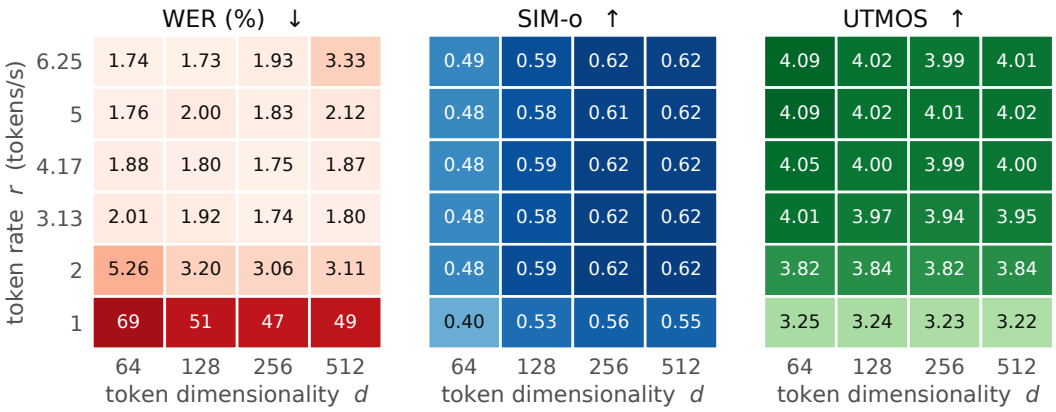


Figure 8: Zero-shot TTS on **LibriSpeech-PC** over the rate–dimension grid. One generator per cell. WER is colour-mapped on a log scale; SIM-o and UTMOS are linear.

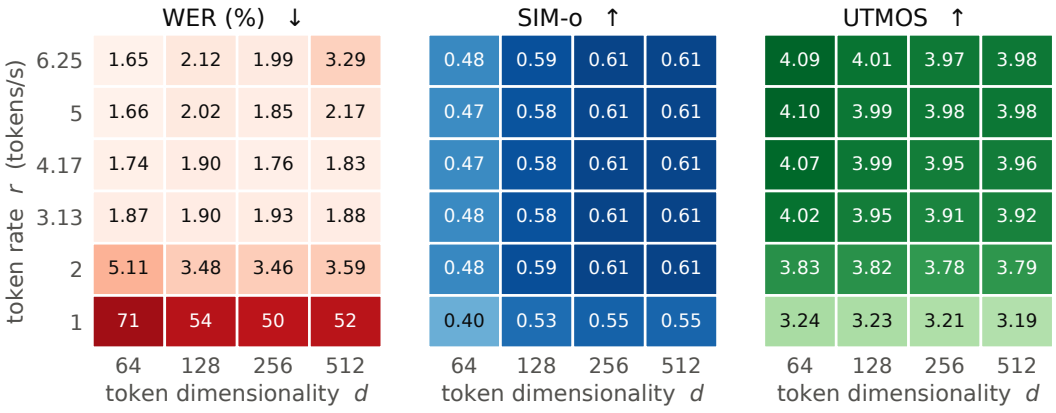


Figure 9: Zero-shot TTS on the **E2 TTS suite**, same construction as Figure 8.

Linguistic fidelity. WER remains low across a broad region between roughly 3.13 and 6.25 tokens per second. At 2 tokens per second dimensionality becomes increasingly important; at 1 token per second WER deteriorates severely at every tested dimensionality.

Speaker fidelity. SIM-o depends much more strongly on dimensionality than on token rate. Across the viable temporal regime, increasing d from 64 to 256 produces a large gain while $256 \rightarrow 512$ adds comparatively little.

Perceptual quality. UTMOS runs the other way: lower-dimensional representations often obtain higher predicted perceptual quality. The representation shape preferred for speaker fidelity is therefore not the shape preferred for perceptual naturalness or for linguistic generation.

F.3 Cross-dataset consistency

The rate–dimension interaction is not specific to Seed-TTS. On LibriSpeech-PC and E2 TTS, as on Seed-TTS, low-dimensional tokens work particularly well at high token rates whereas higher dimensionality becomes more useful as the temporal rate decreases, and the transition at 1–2 tokens per second appears on all three datasets.

Speaker similarity is even more consistent across datasets: SIM-o is nearly constant across token rate at fixed d over much of the useful regime, while changing d produces large differences. This parallels the speaker-probe results of Section 4.2 despite the substantially different downstream task.

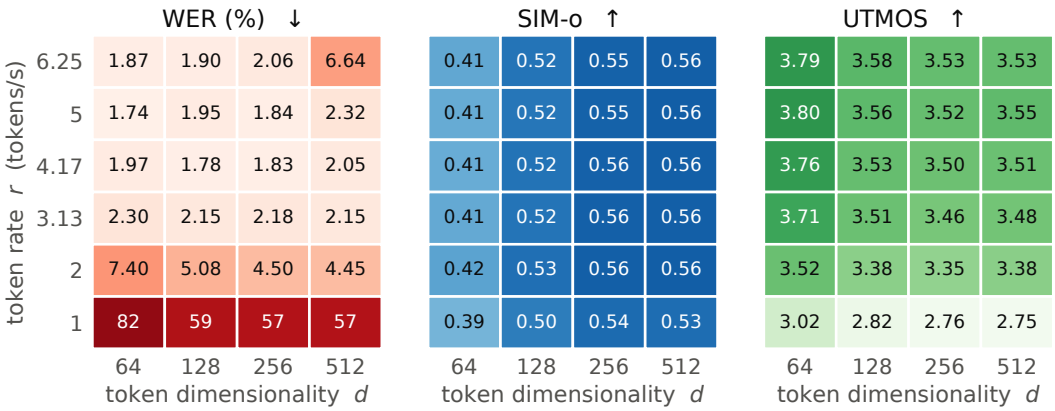


Figure 10: Zero-shot TTS on **Seed-TTS EN**, same construction as Figure 8. The reversal in the WER panel is the clearest instance of the rate–dimension interaction: at 6.25 tokens per second d=64 is best (1.87) and d=512 worst (6.64), while at 2 tokens per second the ordering inverts (7.40 against 4.45).

Table 13: The rate–dimension surface for generated speech with the flow-matching decoder (guidance 3.0, mean of three sampling seeds). One generator per cell. Bold marks the best WER in each row. The reference row gives the real recordings under the same protocol.

	WER (%) ↓				SIM-o ↑			
r (tokens/s)	d=64	128	256	512	d=64	128	256	512
LibriSpeech-PC (ground truth: WER 2.46, SIM-o 0.695)								
6.25	1.74	**1.73**	1.93	3.33	0.487	0.591	0.619	0.619
5.00	**1.76**	2.00	1.83	2.12	0.479	0.583	0.614	0.618
4.17	1.88	1.80	**1.75**	1.87	0.477	0.586	0.617	0.618
3.13	2.01	1.92	**1.74**	1.80	0.482	0.585	0.618	0.620
2.00	5.26	3.20	**3.06**	3.11	0.485	0.595	0.620	0.623
Seed-TTS EN (ground truth: WER 2.18, SIM-o 0.734)								
6.25	**1.87**	1.90	2.06	6.64	0.414	0.519	0.554	0.559
5.00	**1.74**	1.95	1.84	2.32	0.407	0.515	0.553	0.561
4.17	1.97	**1.78**	1.83	2.05	0.405	0.521	0.555	0.559
3.13	2.30	**2.15**	2.18	2.15	0.412	0.519	0.556	0.559
2.00	7.40	5.08	4.50	**4.45**	0.425	0.528	0.559	0.562
E2 TTS suite (ground truth: WER 2.15, SIM-o 0.695)								
6.25	**1.65**	2.12	1.99	3.29	0.480	0.588	0.612	0.611
5.00	**1.66**	2.02	1.85	2.17	0.474	0.580	0.606	0.610
4.17	**1.74**	1.90	1.76	1.83	0.474	0.584	0.609	0.610
3.13	**1.87**	1.90	1.93	1.88	0.477	0.584	0.612	0.613
2.00	5.11	3.48	**3.46**	3.59	0.481	0.592	0.612	0.614

F.4 Text Representation

We compare subword, character and phoneme conditioning at r=4.17 tokens per second and d=256, training one capacity-matched generator for each condition (Table 17).

The choice of text unit has little effect in domain: on LibriSpeech-PC all three obtain 1.74–1.89% WER, with SIM-o and UTMOS nearly identical. The difference is larger out of domain on Seed-TTS EN, where phoneme conditioning reduces WER from 2.91% with Vicuna subwords to 1.83% and character conditioning reaches 2.01%, while speaker similarity and UTMOS again barely move. We therefore use phonemes in the remaining TTS experiments. The choice primarily improves linguistic generalization rather than speaker or perceptual quality.

Table 14: LibriSpeech-PC: every cell of the four families. Ground truth: WER 2.46, SIM-o 0.695, UTMOS 4.10. Flow-decoder cells average three sampling seeds at guidance 3.0.

		WER (%)			SIM-o			UTMOS		
family	w	64	128	512	64	128	512	64	128	512
resynthesis, ℓ_2 decoder	6.25	2.47	2.38	2.37	0.545	0.548	0.548	3.75	3.74	3.75
	5.00	2.39	2.41	2.40	0.540	0.544	0.545	3.76	3.75	3.74
	4.17	2.46	2.45	2.43	0.537	0.542	0.543	3.75	3.75	3.74
	3.13	2.65	2.49	2.43	0.529	0.536	0.538	3.74	3.75	3.74
	2.00	3.42	2.90	2.94	0.506	0.519	0.522	3.64	3.70	3.70
	1.00	41.06	27.65	27.15	0.378	0.416	0.422	2.44	2.68	2.72
resynthesis, flow decoder	6.25	2.75	2.81	2.80	0.599	0.601	0.603	3.60	3.60	3.59
	5.00	2.83	2.71	2.76	0.597	0.600	0.603	3.64	3.63	3.62
	4.17	2.87	2.72	2.72	0.595	0.599	0.601	3.65	3.64	3.62
	3.13	3.02	2.93	3.02	0.591	0.596	0.598	3.65	3.66	3.65
	2.00	4.32	3.21	3.07	0.570	0.584	0.587	3.57	3.64	3.64
	1.00	57.30	39.37	33.33	0.473	0.504	0.513	3.11	3.24	3.27
TTS, ℓ_2 decoder	6.25	1.85	1.82	3.41	0.425	0.502	0.516	3.83	3.82	3.83
	5.00	1.88	1.95	2.04	0.415	0.492	0.518	3.84	3.83	3.84
	4.17	1.97	1.96	1.96	0.414	0.496	0.515	3.84	3.83	3.84
	3.13	2.21	2.22	1.93	0.415	0.491	0.513	3.79	3.80	3.80
	2.00	3.68	3.05	3.19	0.420	0.495	0.511	3.67	3.72	3.73
	1.00	47.90	33.83	34.43	0.313	0.416	0.426	2.08	2.35	2.38
TTS, flow decoder	6.25	1.74	1.73	3.33	0.487	0.591	0.619	4.09	4.02	4.01
	5.00	1.76	2.00	2.12	0.479	0.583	0.618	4.09	4.02	4.02
	4.17	1.88	1.80	1.87	0.477	0.586	0.618	4.05	4.00	4.00
	3.13	2.01	1.92	1.80	0.482	0.585	0.620	4.01	3.97	3.95
	2.00	5.26	3.20	3.11	0.485	0.595	0.623	3.82	3.84	3.84
	1.00	69.32	50.54	48.83	0.404	0.532	0.553	3.25	3.24	3.22

Table 15: Seed-TTS EN: every cell of the four families. Ground truth: WER 2.18, SIM-o 0.734, UTMOS 3.52. Flow-decoder cells average three sampling seeds at guidance 3.0.

		WER (%)			SIM-o			UTMOS		
family	w	64	128	512	64	128	512	64	128	512
resynthesis, ℓ_2 decoder	6.25	2.50	2.63	2.56	0.534	0.538	0.540	3.29	3.28	3.28
	5.00	2.71	2.63	2.58	0.530	0.534	0.537	3.30	3.29	3.29
	4.17	2.80	2.67	2.64	0.526	0.532	0.535	3.29	3.29	3.28
	3.13	3.00	2.91	2.94	0.518	0.527	0.529	3.27	3.28	3.27
	2.00	4.55	3.76	3.45	0.494	0.508	0.512	3.17	3.21	3.21
	1.00	56.86	42.09	40.87	0.371	0.404	0.413	1.92	2.06	2.10
resynthesis, flow decoder	6.25	2.45	2.44	2.39	0.611	0.615	0.617	2.97	2.97	2.96
	5.00	2.49	2.47	2.49	0.608	0.613	0.615	2.99	2.97	2.97
	4.17	2.55	2.45	2.44	0.605	0.611	0.614	2.98	2.97	2.97
	3.13	2.67	2.53	2.56	0.599	0.606	0.608	2.97	2.98	2.99
	2.00	4.88	3.32	3.25	0.579	0.592	0.595	2.90	2.95	2.97
	1.00	69.29	48.98	43.33	0.493	0.520	0.528	2.51	2.60	2.63
TTS, ℓ_2 decoder	6.25	2.06	2.63	9.19	0.350	0.428	0.456	3.55	3.50	3.42
	5.00	2.22	2.45	2.84	0.343	0.420	0.454	3.58	3.52	3.51
	4.17	2.28	2.21	2.53	0.339	0.423	0.452	3.58	3.52	3.50
	3.13	3.01	2.68	2.47	0.343	0.418	0.447	3.53	3.50	3.47
	2.00	5.72	4.68	5.09	0.349	0.418	0.441	3.37	3.38	3.38
	1.00	60.25	44.99	48.98	0.289	0.373	0.388	1.84	2.02	2.02
TTS, flow decoder	6.25	1.87	1.90	6.64	0.414	0.519	0.559	3.79	3.58	3.53
	5.00	1.74	1.95	2.32	0.407	0.515	0.561	3.80	3.56	3.55
	4.17	1.97	1.78	2.05	0.405	0.521	0.559	3.76	3.53	3.51
	3.13	2.30	2.15	2.15	0.412	0.519	0.559	3.70	3.51	3.48
	2.00	7.40	5.08	4.45	0.425	0.528	0.562	3.52	3.38	3.38
	1.00	81.79	58.51	57.13	0.388	0.503	0.532	3.02	2.82	2.75

F.5 Comparison with published TTS systems

Table 18 places selected LACONIC operating points alongside published zero-shot TTS systems and the pooled baselines.

Table 16: E2 TTS suite: every cell of the four families. Ground truth: WER 2.15, SIM-o 0.695, UTMOS 4.10. Flow-decoder cells average three sampling seeds at guidance 3.0.

		WER (%)			SIM-o			UTMOS		
family	w	64	128	512	64	128	512	64	128	512
resynthesis, ℓ_2 decoder	6.25	2.42	2.50	2.50	0.539	0.542	0.543	3.75	3.73	3.74
	5.00	2.44	2.45	2.47	0.534	0.537	0.539	3.75	3.74	3.74
	4.17	2.43	2.45	2.47	0.531	0.536	0.537	3.75	3.75	3.74
	3.13	2.53	2.47	2.47	0.524	0.530	0.532	3.74	3.74	3.73
	2.00	3.55	2.84	2.84	0.503	0.514	0.516	3.64	3.70	3.69
	1.00	44.30	30.25	29.29	0.378	0.415	0.420	2.44	2.68	2.72
resynthesis, flow decoder	6.25	2.36	2.38	2.36	0.601	0.603	0.603	3.60	3.60	3.59
	5.00	2.36	2.36	2.38	0.599	0.602	0.603	3.64	3.63	3.62
	4.17	2.41	2.33	2.37	0.596	0.600	0.602	3.65	3.64	3.62
	3.13	2.59	2.46	2.39	0.592	0.597	0.599	3.65	3.66	3.65
	2.00	4.10	2.85	2.77	0.571	0.586	0.589	3.57	3.64	3.64
	1.00	61.60	42.49	36.73	0.477	0.509	0.517	3.11	3.24	3.27
TTS, ℓ_2 decoder	6.25	1.54	2.08	3.15	0.416	0.493	0.505	3.86	3.85	3.86
	5.00	1.57	1.94	2.14	0.407	0.484	0.505	3.86	3.86	3.87
	4.17	1.76	1.93	1.79	0.408	0.486	0.503	3.85	3.85	3.87
	3.13	1.87	2.00	2.04	0.407	0.483	0.504	3.81	3.82	3.82
	2.00	3.59	3.20	3.57	0.412	0.488	0.500	3.67	3.74	3.73
	1.00	48.48	36.29	35.51	0.306	0.408	0.419	2.10	2.36	2.40
TTS, flow decoder	6.25	1.65	2.12	3.29	0.480	0.588	0.611	4.09	4.01	3.98
	5.00	1.66	2.02	2.17	0.474	0.580	0.610	4.10	3.99	3.98
	4.17	1.74	1.90	1.83	0.474	0.584	0.610	4.07	3.99	3.96
	3.13	1.87	1.90	1.88	0.477	0.584	0.613	4.02	3.95	3.92
	2.00	5.11	3.48	3.59	0.481	0.592	0.614	3.83	3.82	3.79
	1.00	70.75	53.59	51.64	0.400	0.531	0.547	3.24	3.23	3.19

Table 17: The text unit buys intelligibility and nothing else, and only out of domain: on Seed-TTS phonemes cut WER from 2.91 to 1.83, while in domain the three units are within noise. Text unit at 4.17 tokens/s and d=256. One generator per row, identical in every other respect (same recipe, 150k steps, and an embedding table of the same size, so the arms are capacity-matched). SIM-o, UTMOS and duration agree to three decimals across all three rows.

	LibriSpeech-PC			*E2 TTS suite*			*Seed-TTS EN*		
System	WER	SIM-o	UTMOS	WER	SIM-o	UTMOS	WER	SIM-o	UTMOS
Subword	1.89	0.614	3.98	2.06	0.608	3.95	2.91	0.554	3.49
Character	1.74	0.617	3.98	1.77	0.611	3.95	2.01	0.560	3.49
Phoneme	1.75	0.617	3.99	1.76	0.609	3.95	1.83	0.555	3.50

As with reconstruction, these results are operating-point context rather than a controlled leaderboard. Published systems differ substantially in model size, training data, representation, vocoder and evaluation procedure. Our purpose is to verify that the low-rate LACONIC representations support practically useful zero-shot generation; the controlled experiments within LACONIC are what establish the effect of representation shape.

G GENERATION CONTROLS

G.1 UTTERANCE-TOKEN DURATION LEAKAGE

During development we observed a failure mode caused by the utterance tokens. Because U is computed globally over the target utterance during training, it can encode the target duration. A generator trained with the target U can therefore learn when to stop from information that is unavailable at inference, where U instead comes from the prompt.

We test this with a 2×2 intervention at r=4.17 tokens per second and d=256 (Table 20). The first intervention resamples the target utterance tokens during training so that they cannot directly reveal the target duration; the second adds Gaussian noise to the conditioning prefix.

The result isolates the cause cleanly. With neither intervention, LibriSpeech-PC WER is 35.61% and generated utterances have only 0.68 times the ground-truth duration. Prefix noise alone does not fix it (35.29% WER, duration ratio 0.67). Resampling U alone reduces WER to 1.82% and restores the duration ratio to 0.94. Combining both gives 1.75% and the same ratio. The same pattern holds on E2 TTS and Seed-TTS EN.

We therefore use utterance-token resampling in all reported generation experiments. The intervention is required specifically because the utterance pathway is global; it is not evidence of a failure of the local low-rate representation.

H Downstream speech understanding

H.1 Setup

The main experiments deliberately use simple controlled probes in order to map information across rate and dimensionality. We additionally evaluate whether LACONIC remains useful under standard downstream protocols, covering automatic speech recognition, emotion recognition, speaker verification and keyword spotting. We consider the full representation $[U; L]$ as well as U-only and, where applicable, L-only inputs.

Protocols. Emotion recognition, speaker verification and keyword spotting follow DASB (Mousavi et al., 2024), using its ECAPA-TDNN probes on IEMOCAP, VoxCeleb1 (Chung et al., 2018) and Speech Commands v0.02 (Warden, 2018) respectively. The ASR column follows ALMTokenizer (Yang et al., 2025a): a LLaMA-3.2-1B decoder (Grattafiori et al., 2024) adapted with LoRA (rank 32, $\alpha = 16$, on the attention and linear projections), the audio tokens and text concatenated in one sequence, trained for 10 epochs and decoded with top-k sampling at $k = 30$; we report test-clean WER with the standard deviation over sampling runs. Continuous LACONIC tokens enter through an MLP projector in place of an embedding lookup.

Two deviations we state explicitly. First, ALMTokenizer trains its ASR reader on $2{,}000$ hours of LibriHeavy while we train on the 960 hours of LibriSpeech, so our ASR column is not directly comparable to the numbers that protocol reports for other tokenizers; we include it to compare LACONIC operating points against one another under a large frozen decoder, not to claim a win on that benchmark. Second, a reader trained only on LibriSpeech degrades sharply on utterances longer than about 17 seconds, which are almost absent from its training distribution; we therefore concatenate consecutive same-chapter utterances with probability 0.5 during training, up to 35 seconds, and split over-long test utterances at low-energy points. This is a property of the reader's training data, not of the representation: slicing the tokens of a whole utterance fails in exactly the same way.

These experiments serve a different purpose from the controlled sweeps: they test whether the compressed representations remain practically usable across heterogeneous speech tasks.

H.2 Results

At 6.25 tokens per second the full LACONIC representation obtains 3.73% ASR WER, 63.8% emotion recognition, 11.42 speaker-verification EER and 97.9% keyword-spotting accuracy. At 3.13 tokens per second the corresponding values are 4.66%, 64.8%, 10.40 and 97.6%.

The decomposition between U and L is informative. The utterance tokens alone reach 65.2/67.0% emotion recognition and 7.11/7.15 EER at 6.25/3.13 tokens per second, while still retaining more than 97% keyword-spotting accuracy. This supports the main-paper observation that globally organized information can be represented efficiently outside the temporally localized stream.

We include DAC, EnCodec, Mimi, WavTokenizer, SiTok, Sylber and SyllableLM as context. Because representation pretraining, downstream training and evaluation details differ across systems, these comparisons should not be read as a strict ranking. They establish that LACONIC's unusually low temporal rates do not prevent the representation from supporting a broad set of conventional speech-understanding tasks.

H.3 SPEECH TRANSLATION

ASR is a monotonic task: the reader may transcribe the token stream roughly in order. To test whether the dimensional ordering survives a task that requires reordering, we additionally evaluate English-to-German speech translation on CoVoST 2 (Wang et al., 2021).

Protocol. We use the full En–De split, discarding the examples whose translation field is marked `REMOVE` as the corpus requires, which leaves 289,179 training, 15,517 validation and 15,526 test utterances (429 hours of training audio, mean duration 5.3 s). Audio is decoded from the distributed 48 kHz MP3 and resampled to 16 kHz. The reader is the same 44M encoder–decoder used for the ASR probe, with the decoder alphabet replaced by a cased German character set, and is trained from scratch for 40k steps on each frozen representation. We report case-sensitive detokenized BLEU with sacreBLEU (Post, 2018).

Results. Table 22 compares four representations at a matched reader. Uncompressed WavLM reaches 19.47 BLEU. All three compressed representations lose roughly four BLEU against it, a larger relative cost than the ASR probe shows, so temporal compression is not free for translation. But among the compressed representations, how the budget is spent barely matters: LACONIC at r=3.13, d=32 reaches 15.54 BLEU, matching WAVLM-POOL at d=1024 (15.62) on a 32× smaller representation and matching its own d=512 read (15.22).

The dimensional ordering therefore holds on a task that the text objective was not trained for and that requires non-monotonic reordering: 100 scalar values per second carry as much translatable content as 3,200.

H.4 SUMMARY OF CONTROLS

The supplementary experiments distinguish two properties of LACONIC. First, the *rate–dimension geometry* is an empirical property: dedicated-rate and dedicated-dimensionality models reproduce the useful operating regime (Appendices C.2 and C.3), and similar temporal-compression behavior appears under alternative input features (Appendix C.1), under a different pretrained encoder (Appendix C.5) and under downstream consumers as different as translation (Appendix H.3). Second, the *coordinate ordering* is a learned property: the nested text objective deliberately makes linguistic information accessible from low-dimensional prefixes, and removing it costs 38.8 WER at r=3.13, d=32 (Appendix D.1). LACONIC combines the two to provide a single instrument for measuring how speech information changes as capacity is redistributed between time and dimensionality.

Table 18: Zero-shot TTS against published systems on three test sets, each scored with its benchmark's own protocol (Appendix B.4). Published rows are copied from the cited papers (†); *ground truth* is our measurement, with the published value in brackets. Data in hours; tok/s is the rate of the representation the model generates, where stated ("–": mel frames or not stated). LACONIC rows use the flow-matching decoder (guidance 3.0, mean of three sampling seeds). On the E2 TTS suite WER is corpus-level from HuBERT-large; on the other two it is the per-utterance mean. *The same generator recipe trained on Mimi's pre-RVQ features (512-d, 12.5 Hz) and rendered by Mimi's own decoder, sampled with noise scale 1.0: with the deterministic generation used for every other row it rarely stops (WER 26.8 / 9.2 / 31.3), since Mimi's causal features give the end predictor little to fire on. Rendering ground-truth Mimi features instead reads 2.52 / 2.45 / 2.25 WER. ‡The same generator recipe over FlexiCodec's continuous bottleneck at the same rate: its 512-d pre-RVQ acoustic residual plus the 5-d pre-FSQ semantic projection, one pair per merged token, with each token's span in 12.5 Hz frames as one further channel (the rate is dynamic, so a generator must predict duration too; rendering ground-truth features with a uniform span instead costs 29.2 WER). Rendering re-applies FlexiCodec's own FSQ and RVQ-24 and its decoder, which reproduces the codec's published resynthesis exactly: that ceiling reads $4.76/4.12/3.18$ WER. Generated deterministically like every other row (no utterance reaches the slot limit, unlike the Mimi baseline); sampling noise 1.0 would trade intelligibility for quality, reading $22.38/29.76/24.05$ WER.

System	Params	Data (h)	tok/s	WER (%) ↓	SIM-o ↑	UTMOS ↑
LibriSpeech-PC test-clean						
Ground truth [2.47 / 0.69 / 4.09]	–	–	–	2.46	0.695	4.10
F5-TTS†	336M	100k	–	2.42	0.66	3.90
F5-TTS small†	158M	0.6k	–	2.20	0.60	4.10
E2 TTS (F5 reprod.)†	333M	100k	–	2.95	0.69	3.56
MaskGCT†	1048M	100k	50	2.72	0.69	3.90
DiTAR†	600M	100k	–	2.39	0.67	4.22
CLEAR-Base†	439M	50k	7.8	2.21	0.59	4.22
CLEAR-Large†	686M	50k	7.8	1.88	0.59	4.22
SylFlow†	72M	0.6k	≈5	3.10	0.31	4.27
SylFlow†	72M	47.6k	≈5	2.35	0.36	4.33
LACONIC, d=64	75M	0.55k	6.25	1.74	0.487	4.09
LACONIC, d=256	75M	0.55k	4.17	1.75	0.617	3.99
LACONIC, d=512	75M	0.55k	3.13	1.80	0.620	3.95
LACONIC-POOL	75M	0.55k	6.25	1.87	0.399	4.21
LACONIC-POOL	75M	0.55k	4.17	1.90	0.390	4.18
LACONIC-POOL	75M	0.55k	3.13	2.91	0.365	4.14
Mimi prequant, same generator*	75M	0.55k	12.5	7.25	0.284	3.08
FlexiCodec prequant, same generator‡	75M	0.55k	4.20	12.25	0.220	1.98
Seed-TTS EN						
Ground truth [2.14 / 0.73 / –]	–	–	–	2.18	0.734	3.52
Seed-TTS ICL†	n/r	n/r	–	2.25	0.762	–
F5-TTS†	336M	100k	–	1.83	0.67	3.76
CosyVoice 2†	500M	170k	25	2.57	0.652	–
MaskGCT†	1048M	100k	50	2.62	0.717	–
DiTAR†	1B	280k	–	1.69	0.735	–
Spark-TTS†	500M	102k	50	1.98	0.584	–
SiTok-AR-TTS†	500M	100k	12.5	2.46	0.64	–
SylFlow†	72M	0.6k	≈5	2.62	0.31	4.19
SylFlow†	72M	47.6k	≈5	1.92	0.35	4.31
LACONIC, d=64	75M	0.55k	6.25	1.87	0.414	3.79
LACONIC, d=256	75M	0.55k	4.17	1.83	0.555	3.50
LACONIC, d=512	75M	0.55k	3.13	2.15	0.559	3.48
LACONIC-POOL	75M	0.55k	6.25	1.85	0.335	3.92
LACONIC-POOL	75M	0.55k	4.17	1.84	0.322	3.83
LACONIC-POOL	75M	0.55k	3.13	3.26	0.302	3.86
Mimi prequant, same generator*	75M	0.55k	12.5	9.09	0.268	2.88
FlexiCodec prequant, same generator‡	75M	0.55k	4.20	14.20	0.193	1.83
E2 TTS suite						
Ground truth [2.0 / 0.695 / –]	–	34	–	2.15	0.695	4.10
VALL-E†	n/r	60k	75	4.9	0.500	–
NaturalSpeech 3†	n/r	60k	–	2.6	0.632	–
Voicebox†	n/r	50k	–	2.2	0.667	–

Table 19: The same generated tokens rendered by the two decoders at 6.25 and 3.13 tokens/s; generation is deterministic, so the decoder is the only variable. WER (%). *Penalty* is the WER cost of reading d=512 instead of the best dimensionality in the row.

Test set	Decoder, r	d=64	128	256	512	penalty
LibriSpeech-PC	ℓ_2, 6.25	1.85	1.82	1.91	3.41	1.59
	flow, 6.25	1.74	1.73	1.93	3.33	1.60
	ℓ_2, 3.13	2.21	2.22	1.87	1.93	0.06
	flow, 3.13	2.01	1.92	1.74	1.80	0.06
Seed-TTS EN	ℓ_2, 6.25	2.06	2.63	4.26	9.19	7.13
	flow, 6.25	1.87	1.90	2.06	6.64	4.78
	ℓ_2, 3.13	3.01	2.68	2.64	2.47	0.00
	flow, 3.13	2.30	2.15	2.18	2.15	0.00
E2 TTS suite	ℓ_2, 6.25	1.54	2.08	1.95	3.15	1.61
	flow, 6.25	1.65	2.12	1.99	3.29	1.64
	ℓ_2, 3.13	1.87	2.00	1.88	2.04	0.16
	flow, 3.13	1.87	1.90	1.93	1.88	0.01

Table 20: Resampling the target's utterance tokens is the whole fix, and prefix noise on its own does nothing: resampling alone takes WER from 35.6 to 1.8, noise alone leaves it at 35.3. Utterance-token handling at 4.17 tokens/s and d=256: a 2×2 over resampling the target's utterance tokens during training and perturbing the input prefix with Gaussian noise. *dur.* is generated duration over the ground truth's. Without resampling the generator reads duration off the target's utterance tokens and, given the prompt's at inference, stops roughly a third early.

	LibriSpeech-PC		*E2 TTS suite*		*Seed-TTS EN*	
System	WER	dur.	WER	dur.	WER	dur.
Neither	35.61	0.68	45.29	0.69	32.39	0.83
Prefix noise only	35.29	0.67	42.44	0.71	32.40	0.82
Resampling only	1.82	0.94	1.90	0.94	1.85	0.94
Both (adopted)	1.75	0.94	1.76	0.93	1.83	0.95

Table 21: Understanding benchmarks. Rows above the first rule are as reported by SiTok (Wang et al., 2026), whose ASR column comes from Yang et al. (2025a); we reproduce their protocols. ASR: a LLaMA-3.2-1B (Grattafiori et al., 2024) model with LoRA on the attention and feed-forward projections and a trained projector over the tokens, trained on LibriSpeech (Panayotov et al., 2015) and scored on all 2,620 test-clean utterances with top-k (k=30) sampling, mean $\pm$ standard deviation over three seeds (greedy decoding gives 2.79 and 3.23). ER, SV and KS: the DASB probes (Mousavi et al., 2024), an ECAPA-TDNN (Desplanques et al., 2020) over frozen tokens, on IEMOCAP (Busso et al., 2008) four-class emotion (fold 1, 528 test utterances), VoxCeleb1-O (Chung et al., 2018) verification (37,611 trials, equal error rate) and Speech Commands (Warden, 2018) twelve-class keyword spotting (4,875 test clips). *Dim./CB* is the channels per token for continuous tokenizers and the number of codebooks for discrete ones. LACONIC reads d=512; U only and L only are the same tokens with one part withheld from the probe, so U only spends five tokens per utterance whatever its length. Deviations from the standard protocols: our 1B model sees 960 h of LibriSpeech rather than 2,000 h of LibriHeavy, with consecutive utterances joined during training and test utterances longer than 17 s tokenized piecewise (Appendix H.1); keyword spotting uses Speech Commands v0.02, which the DASB recipe downloads; speaker verification trains on the prepared 3 s chunks rather than fresh random crops. Single seed for ER, SV and KS. Rows marked *prequant* are our measurements under exactly the protocol used for LACONIC, on each model's continuous features before quantization: Mimi's encoder output before its RVQ, Sylber's syllable embeddings, SyllableLM's span-pooled Data2Vec2 features before their k-means, and FlexiCodec's pre-quantizer pair, its 512-d pre-RVQ acoustic residual concatenated with the 5-d pre-FSQ projection of its semantic latent, one vector per merged token at a fixed merging threshold calibrated so the measured rate is 4.16 tokens per second (the threshold is corpus-dependent); *Tokens/s* is their measured test-clean rate. The published Mimi row above reads its discrete codes instead. The prequant features are standardized per channel with LibriSpeech statistics before the reader, which LACONIC's tokens, already near unit scale, are not. Sylber finds no syllable in 4,162 Speech Commands clips, 3,761 of them *silence*, which gives that class away; its emotion probe skipped the few updates whose gradients were not finite.

System	Tokens/s	Dim./CB	ASR (%) ↓	ER (%) ↑	SV (%) ↓	KS (%) ↑
DAC	150	3	58.4	48.9	17.8	68.8
EnCodec	150	3	77.2	47.4	15.5	79.3
Mimi	100	8	23.1	54.3	19.7	92.2
WavTokenizer	40	1	45.6	51.1	19.4	65.3
GLM4-Voice	12.5	1	16.3	–	–	–
SiTok (CN=1)	12.5	1	4.95	63.5	13.8	96.9
SiTok (CN=4)	50	4	4.49	64.4	8.59	97.7
Mimi , prequant	12.59	512	8.31	58.1	4.82	98.2
Sylber , prequant	3.86	768	7.06	64.2	12.02	97.6
SyllableLM , prequant	5.79	768	4.13	63.4	16.93	96.1
FlexiCodec , prequant	4.16	517	12.76	56.1	14.95	97.0
LACONIC $[U; L_{:,:512}]$	6.25	512	3.73	63.8	11.42	97.9
U only	5/utt.	512	–	65.2	7.11	97.3
L only	6.25	512	–	–	–	97.7
LACONIC $[U; L_{:,:512}]$	3.13	512	4.66	64.8	10.40	97.6
U only	5/utt.	512	–	67.0	7.15	97.4
L only	3.13	512	–	–	–	97.7

Table 22: Speech translation, CoVoST 2 English→German. Case-sensitive detokenized BLEU on the 15,526-utterance test split, from the same 44M encoder–decoder reader trained from scratch on each frozen representation for 40k steps. Compression costs about four BLEU regardless of how it is done; among compressed representations, spending the budget on dimensions rather than time costs nothing.

Representation	r	d	numbers/s	BLEU ↑
Uncompressed WavLM	50	1024	51,200	**19.47**
WavLM-Pool	3.13	1024	3,200	15.62
Laconic, d=512	3.13	512	1,600	15.22
Laconic, d=32	3.13	32	**100**	15.54