2D LEARNED SPEECH COMPRESSION WITH CONTENT-ADAPTIVE TIME-FREQUENCY MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural audio codecs have relied on one-dimensional sequence representations, where speech is transformed into temporally ordered latent representations. However, this formulation does not explicitly model the inherent time-frequency structure of speech, treating frequency as a representation-dependent dimension and limiting natural adaptation across different sampling rates. To address this core limitation, we represent speech as a time-frequency spatial field via a complex short-time Fourier transform, preserving frequency and time as explicit spatial dimensions. Inspired by learned image compression, we then formulate speech coding as a two-dimensional learned compression problem over this spatial field, applying learned image compression models directly to compress the field. This enables a single model to operate on native 16, 24, and 48 kHz speech without sampling-rate-specific branches. We further apply content-adaptive token organization to the time-frequency field for sequence modeling, stabilize entropy skipping, and merge entropy-coded streams into a compact bitstream. Experiments show that CA2D, with only 69M parameters, achieves strong low-bitrate quality while maintaining a unified architecture across sampling rates, a notable achievement given its compact size. This single compact model operates across 16, 24, and 48 kHz and consistently provides competitive or superior objective and subjective quality compared with existing neural codecs: at around 800 bps, CA2D achieves comparable or better quality than 1 kbps-class models with over 100M parameters, such as BigCodec (159M parameters, 1040 bps), while the same model at 48 kHz is comparable to the 6 kbps operating point of FlowDec, a full-band codec. These results clearly demonstrate the effectiveness of treating speech as a two-dimensional spatial field for scalable neural audio compression with native multi-rate operation and low-bitrate perceptual quality.

1 INTRODUCTION

Neural audio codecs (NACs) have achieved progress in high-quality audio compression. Most existing approaches, such as SoundStream (Zeghidour et al., 2021), EnCodec (Défossez et al., 2022), DAC (Kumar et al., 2023), and BigCodec (Xin et al., 2024), represent speech as temporally ordered latent sequences under a one-dimensional formulation. This paradigm does not explicitly preserve the time-frequency structure of speech: frequency is treated as a representation-dependent dimension rather than an independent spatial axis, limiting flexibility across sampling rates. Figure 1 contrasts the two: conventional codecs collapse frequency into a representation-dependent sequence, whereas CA2D keeps time and frequency as explicit axes.

Frequency-domain codecs introduce spectral representations, such as MDCTCodec (Jiang et al., 2024), FreqCodec (Du et al., 2024), MelCap (Li et al., 2025), and UniSRCodec (Zhang et al., 2026), but frequency is typically used as a transform domain or input representation rather than a spatial dimension preserved throughout compression. To improve frequency scalability, we elevate frequency to an explicit axis parallel to time, representing speech as a two-dimensional time-frequency space, and adopt a learned image compression paradigm to compress this 2D representation. Thus, sampling rate changes only the frequency-axis resolution, not the model architecture.

We introduce CA2D, a two-dimensional learned speech compression framework that treats time and frequency as explicit spatial dimensions throughout analysis and synthesis. Based on a complex

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.

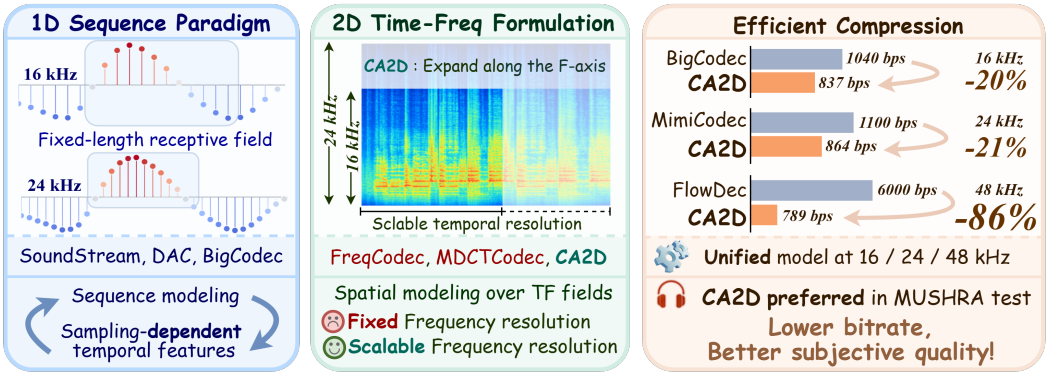


Figure 1: From 1D sequence compression to 2D time-frequency learning. The scalable F-axis lets CA2D unify multi-rate audio compression in one model, achieving lower bitrate at higher quality.

STFT representation, CA2D enables sampling rate to be viewed as an input resolution attribute rather than a model-specific design choice, allowing a single model to operate across native 16, 24, and 48 kHz speech without architecture change. The time and frequency axes are asymmetric and the frequency length varies with sampling rate, so fixed-order serialization is suboptimal and calls for a content-adaptive token organization scheme; low bitrates further make reliable entropy coding with compact bitstreams non-trivial.

Experiments show that CA2D achieves strong low-bitrate reconstruction quality with a unified architecture across sampling rates. At around 800 bps, it is comparable to 1 kbps-class codecs such as BigCodec and, at 48 kHz, to the 6 kbps operating point of FlowDec (Welker et al., 2025). Our contributions are as follows:

- **A 2D formulation for scalable audio compression.** We reformulate neural audio compression as two-dimensional compression over time-frequency fields, where sampling rate becomes an input resolution attribute rather than a model-specific design choice.
- **LIC-based compression of 2D time-frequency representations with content-adaptive token organization.** We leverage learned image compression models and content-adaptive token organization to compress the 2D time-frequency representation and show that it remains effective across substantially different frequency resolutions.
- **Efficient low-bitrate coding with strong multi-rate performance.** We introduce reliable entropy coding and compact bitstream merging, enabling CA2D to achieve strong objective and subjective quality at extremely low bitrates across native sampling rates.

2 Related Work

2.1 Neural Audio Compression

Neural audio compression has mainly followed waveform-domain temporal sequence modeling. SoundStream and EnCodec established temporal sequence compression with residual vector quantization (RVQ) and neural entropy models; DAC improved quantization and adversarial training; BigCodec targeted low-bitrate scenarios; FocalCodec (Della Libera et al., 2026) and WavTokenizer (Ji et al., 2025) explored tokenization. These methods demonstrate the effectiveness of learned latents and neural entropy models, but their time-order prior leaves frequency implicit, so sampling rate changes typically require adjusting frame rate, downsampling rate, or retraining.

Frequency-domain approaches introduce transforms explicitly. MDCTCodec, FreqCodec, MelCap, and UniSRCodec obtain spectral representations via STFT, MDCT, or mel transforms. Works like UniSRCodec also use 2D tokenizers, but their frequency axis is fixed by the spectral front end and tied to one sampling rate, rather than varying with the input rate. CA2D instead treats frequency as a native, variable-length axis parallel to time with constant physical spacing, applies learned image compression to this 2D representation, and thereby changes only frequency-axis resolution with sampling rate, not architecture (Figure 1).

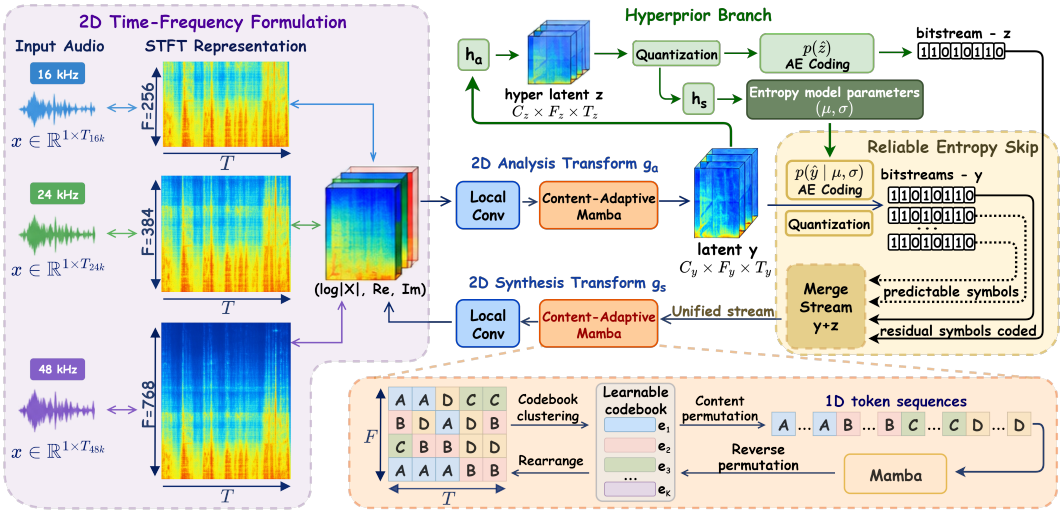


Figure 2: Overall architecture of the proposed 2D learned speech compression framework.

2.2 Learned Image Compression and Entropy Modeling

Learned image compression (LIC) provides a reference for compressing 2D time-frequency representations. Ballé et al. (2018) introduced the scale hyperprior and ELIC (He et al., 2022) developed spatial and channel-wise context models, showing that on explicit spatial axes, learned transforms with conditional probability modeling efficiently compress spatial latents. Content-adaptive token organization (Chen et al., 2025) indicates that clustering and reordering tokens via a learnable codebook exploits content-dependent dependencies better than fixed raster traversal. CA2D transfers this to speech time-frequency fields: frequency varies with sampling rate, so content-adaptive token organization avoids fixed traversal and enables unified modeling under variable frequency resolution. For entropy modeling, ECC (Xu et al., 2026) applies hyperpriors, context modeling, and entropy skipping to speech, but typically targets one-dimensional latents and fixed-sampling-rate sequences. CA2D instead implements reliable entropy coding, compact bitstream merging, and entropy skipping on 2D time-frequency latents with variable frequency resolution.

3 Method

3.1 2D Time-Frequency Formulation

We formulate learned speech compression in the 2D time-frequency domain and instantiate it as CA2D. Given a waveform x sampled at f_s, we compute its complex STFT $S = \mathrm{STFT}(x) \in \mathbb{C}^{F\times T}$ and represent it as a three-channel 2D field $X \in \mathbb{R}^{3\times F\times T}$ with log-magnitude, real, and imaginary parts. Frequency and time remain spatial axes. The three channels are coupled components of one complex spectrum: log-magnitude exposes perceptual structure, while the real and imaginary parts provide phase information that, together with log-magnitude, determines waveform reconstruction.

The representation is compressed by $y = g_a(X;\phi)$, quantized as $\hat{y} = Q(y)$, and reconstructed as $\hat{X} = g_s(\hat{y};\theta)$ followed by inverse STFT (Figure 2). Because the three channels are coupled, reconstruction uses waveform, perceptual, adversarial, and representation-level objectives rather than an image-style distortion on X. We adjust the STFT parameters with the sampling rate: when the sampling rate increases by a factor of k, the FFT size and hop length are increased by approximately the same factor, so that both the frequency spacing and the frame spacing remain approximately constant. The frequency length F therefore grows with f_s, while the temporal length T depends only on duration. Unlike 1D sequence codecs, where frequency is folded into a representation-dependent dimension, here it remains an explicit axis with shared weights. Sampling-rate changes are absorbed by the STFT resolution rather than by rate-specific branches: the same analysis transform, entropy model, and synthesis transform apply to all native rates with identical parameters.

3.2 CONTENT-ADAPTIVE MAMBA

The analysis and synthesis transforms adopt the CMIC architecture (Chen et al., 2025): convolutional components perform local spatial transformation and content-adaptive Mamba blocks (Gu & Dao, 2023) model long-range dependencies. Since Mamba operates on a 1D sequence, the 2D time-frequency latent must be serialized. Unlike the two approximately symmetric spatial axes in LIC, the time-frequency axes are asymmetric: the temporal axis describes the evolution of a 1D signal, while the frequency axis describes the spectrum; the frequency length varies with the sampling rate, and the time-frequency energy distribution is highly non-uniform, with different frequency bands contributing differently to perception. A fixed raster scan is therefore suboptimal, as it imposes an artificial ordering bias that cannot reflect the content-dependent dependencies in the time-frequency latent.

We adapt content-adaptive token organization to the time-frequency latent, so that the ordering is determined by latent content rather than by a fixed axis traversal, and remains effective under this asymmetric and variable-length time-frequency structure. As shown in Figure 2, for a latent representation $Y \in \mathbb{R}^{C\times F\times T}$, a trainable codebook defines content-dependent groups by assigning each element to its nearest entries based on similarity, and the assignments determine a permutation $\pi_k = \mathcal{O}_k(Y_k)$ for the k-th Mamba block. The block processes the reordered sequence as $Y'_k = \mathrm{Mamba}_k(\pi_k(Y_k))$, and the inverse permutation restores the spatial arrangement, $Y_{k+1} = \pi_k^{-1}(Y'_k)$.

3.3 RELIABLE ENTROPY SKIP

Following learned compression frameworks, the entropy model predicts Gaussian parameters (μ, σ) for each latent element through a hyperprior. The skip decision uses the predicted scale σ_n: for a latent element y_n with residual $r_n = y_n - \mu_n$, we set $s_n = \mathbb{1}(\sigma_n \le \tau_\sigma)$; when $s_n = 1$, the residual is omitted and the decoder sets it to zero. Since σ_n is available at both encoder and decoder, no skip mask needs to be transmitted. In ECC, the likelihood loss is directly masked by the skip status, so skipped variables are removed from both rate computation and likelihood learning:

$$R_{\text{skip}} = -\sum_n (1 - s_n) \log_2 p_{\hat{y}_n}(\hat{y}_n \mid \mu_n, \sigma_n).$$

We refer to this as the masked-likelihood model. In multi-rate joint training, we observe that some utterances cause bitrate spikes by several times. To locate the cause, we compare skip-enabled and skip-disabled inference using the same trained model. As shown in Figure 3(a), disabling entropy skipping exposes a subset of samples with abnormally high and fluctuating bitrates under masked-likelihood training.

Figure 3(b) further reveals the source of this instability from the latent–mean relationship. Under masked likelihood, the latent distribution y becomes misaligned with the predicted mean μ: the marginal distributions are shifted, the joint density $p(\mu, y)$ spreads away from the diagonal, and the residual $y - \mu$ exhibits a wider distribution. In contrast, full-likelihood training keeps y and μ closely aligned in all three views. These observations indicate that masked-likelihood training prevents the entropy model from accurately modeling the latent distribution, leading to unreliable entropy estimation and bitrate instability.

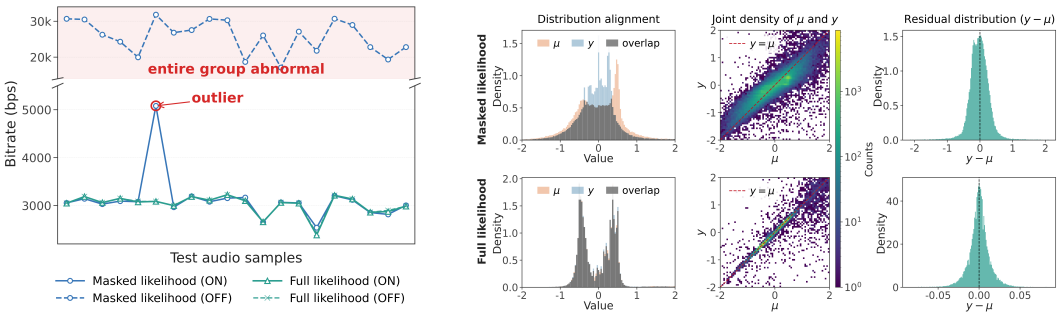
This misalignment can be explained by the gradient behavior under masked likelihood. In a conditional Gaussian model, the likelihood supervises both the mean and the scale:

$$\frac{\partial[-\log p(y_n \mid \mu_n, \sigma_n)]}{\partial \mu_n} = \frac{\mu_n - y_n}{\sigma_n^2}, \qquad \frac{\partial[-\log p(y_n \mid \mu_n, \sigma_n)]}{\partial \sigma_n} = \frac{1}{\sigma_n} - \frac{(y_n - \mu_n)^2}{\sigma_n^3}.$$

The former provides the force that pulls μ_n toward y_n, while the latter vanishes at $\sigma_n = |y_n - \mu_n|$. Once the likelihood of skipped positions is masked, both μ_n and σ_n lose supervision, i.e., **missing likelihood supervision**. Consequently, a small σ_n can persist even as the residual grows, causing the entropy model to underestimate uncertainty and eventually producing unreliable entropy skipping.

To avoid this, we decouple likelihood learning from actual coding and always compute the full likelihood for all latent positions, applying entropy skipping only when constructing the actual bitstream:

$$R_y = -\sum_n \log_2 p_{\hat{y}_n}(\hat{y}_n \mid \mu_n, \sigma_n).$$



(a) (b)

Figure 3: Failure analysis of entropy skipping under masked-likelihood training. (a) Disabling skipping exposes bitrate instability. (b) Latent–mean misalignment, corrected by full-likelihood training.

Skipping a position then only omits its residual and reconstructs from μ_n, without affecting probability-model supervision. As shown in Figure 3(b), full-likelihood training restores the alignment between y and μ and stabilizes the bitrate behavior in Figure 3(a): the full-likelihood curves with skip on and skip off almost coincide. Skipping is effective once training is stable, with about **98%** of latent elements skipped (App. A.2).

Full likelihood resolves the missing supervision caused by masking, but the timing of enabling skipping still needs control. Enabling skipping before the entropy model has sufficiently generalized changes the optimization objective too early, making it difficult for the shared entropy model to learn the complete latent distribution. We therefore adopt a **generalization-before-skip** strategy: skipping is disabled during model adaptation, allowing the entropy model to first generalize across sampling rates and data distributions, and is enabled only after stabilization. Together, full-likelihood training and generalization-before-skip decouple probabilistic modeling from transmission decisions and resolve unreliable entropy skipping.

3.4 Efficient Bitstream Merging

Following multi-stream entropy coding designs, latent components are commonly encoded as separate streams in the CompressAI codebase (Bégaint et al., 2020). The original implementation encodes different latent groups as 11 independent arithmetic streams, each requiring a 32-bit header for its length. For image benchmarks such as Kodak and CLIC (Kodak, 1993; CLIC, 2021), this overhead is negligible: a 768×512 image incurs only 0.0009 bpp against the 0.1 bpp at which image codecs typically operate. However, short speech segments contain far fewer samples, making the same overhead non-negligible at low bitrates. For a 2.55-s, 16-kHz speech segment, the original 11-stream implementation introduces approximately 138 bps of fixed overhead, corresponding to 13.8% of a 1-kbps bitstream. We therefore merge the independent streams into a single bitstream with only one compact 8-bit header, reducing the overhead to approximately 3.14 bps without modifying the learned probability model or its autoregressive dependencies.

4 Experiments

We evaluate low-bitrate rate-distortion at 16 kHz, operation along the two axes of the time-frequency field (sampling rate and input duration) with shared parameters, and controlled ablations of content-adaptive organization and entropy skipping. Datasets are described in App. A.3 and the training procedure in App. A.4.

4.1 Experimental Setup

Evaluation metrics. For 16 kHz evaluation we report LSD, MCD, PESQ, ViSQOL-S, F0 correlation, F0 RMSE, STOI, and ESTOI (Rix et al., 2001; Hines et al., 2015; Taal et al., 2011; Jensen & Taal, 2016). For 24 and 48 kHz we additionally report high-frequency LSD, ViSQOL-A, PEAQ-ODG, and 2fmodel (Hines et al., 2015; Thiede et al., 2000; Torcoli et al., 2021). LSD, MCD, and

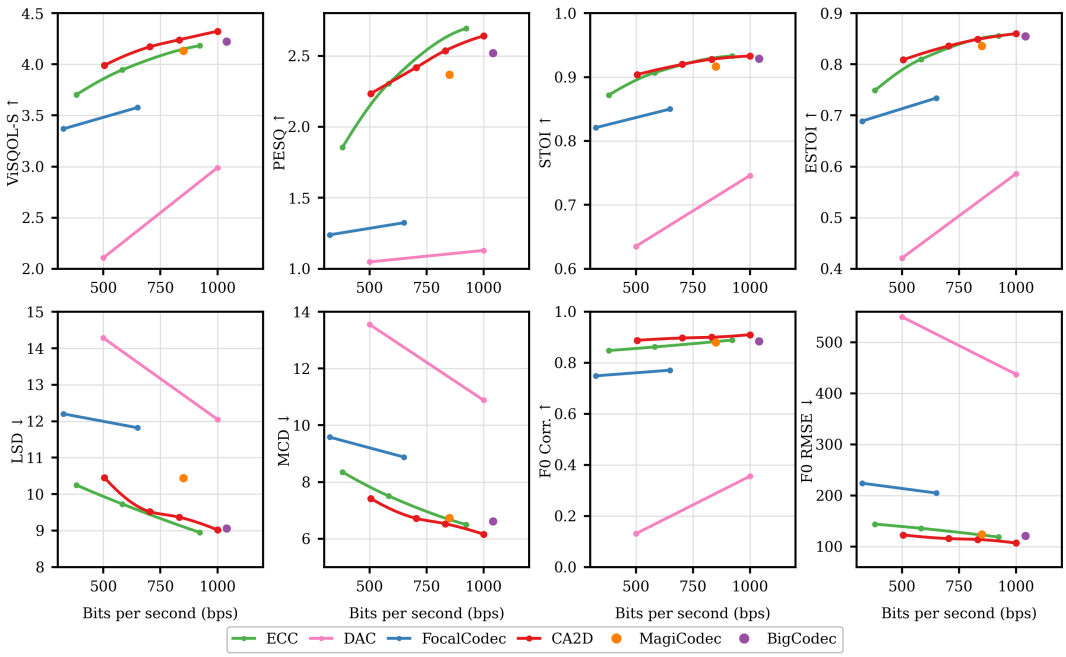


Figure 4: Rate–distortion/quality curves on the full 16 kHz test set.

F0 RMSE are lower-is-better; the rest are higher-is-better. Full metric configurations are provided in App. A.5.

Baselines. We compare against neural audio codecs, including waveform-domain codecs (DAC, BigCodec, MagiCodec, FocalCodec, WavTokenizer, EnCodec, MimiCodec, and FlowDec) (Kumar et al., 2023; Xin et al., 2024; Song et al., 2026; Della Libera et al., 2026; Ji et al., 2025; Défossez et al., 2022; 2024; Welker et al., 2025), and transform-domain methods (ECC and MDCTCodec) (Xu et al., 2026; Jiang et al., 2024). Operating points and the force-encoding protocol are listed in App. A.6. All CA2D bitrates are measured from the actual arithmetic-coded bitstream.

4.2 Main Rate–Distortion Results (16 kHz)

We first evaluate on the full 16 kHz LibriTTS (Zen et al., 2019) test set. Full numerical results are reported in Table A4 (App. A.7), and the rate–distortion (RD) curves are shown in Figure 4. CA2D establishes a new operating point for very-low-bitrate speech coding, reaching roughly 2.2 PESQ and 4.0 ViSQOL-S at about 0.5 kbps, and 2.6 PESQ and 4.3 ViSQOL-S at about 1 kbps, while at comparable low bitrates achieving the best ViSQOL-S, MCD, F0 Corr., F0 RMSE, and ESTOI among all evaluated systems.

At comparable or up to 17–20% lower bitrates, CA2D consistently outperforms MagiCodec and matches or surpasses BigCodec and FocalCodec on the majority of metrics, with clear advantages on PESQ, ViSQOL-S, and the spectral and prosodic measures. The gain is most pronounced on F0 Corr. and F0 RMSE—at 832 bps, CA2D already exceeds DAC at roughly 3.6× its bitrate. This indicates that the benefit of the proposed 2D representation extends beyond waveform reconstruction to the stability of the pitch contour, which is critical for perceptual naturalness at very low bitrates.

Against ECC, which is also based on a probabilistic entropy model, the two systems trade wins: ECC holds a slight edge on PESQ, whereas CA2D leads on ViSQOL-S and both F0 metrics at lower rates, with the remaining metrics essentially on par. Using DAC as the anchor, CA2D reduces bitrate by 81.37% for PESQ and 75.66% for ViSQOL-S, versus 81.61% and 72.36% for ECC.

4.3 Frequency Axis: Multi-Sampling-Rate Operation

The proposed 2D formulation predicts that the sampling rate sets only the frequency-side resolution without touching any learned parameter. Tables A5 (24 kHz) and A6 (48 kHz) verify this prediction,

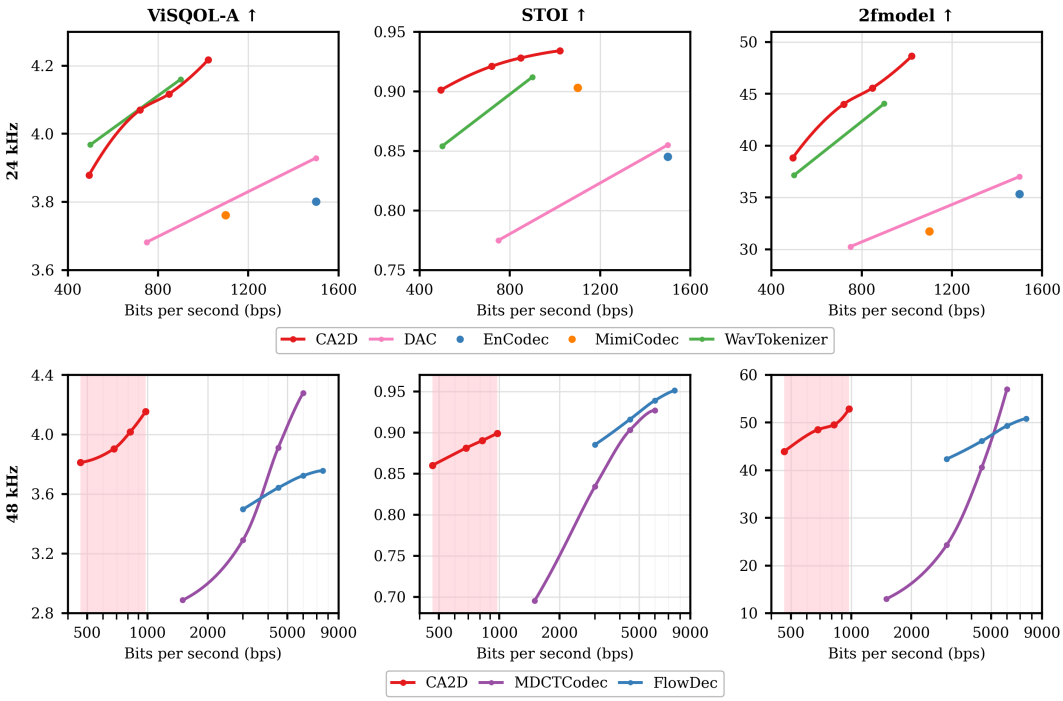


Figure 5: Rate–quality curves of CA2D and baselines at 24 kHz (top) and 48 kHz (bottom). Rows: sampling rates; columns: metrics; shaded region: CA2D bitrate range.

and Figure 5 visualizes the corresponding rate–quality curves. At 24 kHz, CA2D at 494–1022 bps surpasses DAC, EnCodec, and MimiCodec at all evaluated operating points below 1600 bps on almost every metric while using a lower bitrate; at 48 kHz, CA2D reaches 4.154 ViSQOL-A and 52.91 on 2fmodel at 978 bps, exceeding every FlowDec operating point over 3–7.5 kbps in these two metrics and approaching MDCTCodec at 6000 bps with roughly one-sixth of its bitrate. BigCodec and MagiCodec are not suitable for this axis: although their objective scores appear acceptable, the decoded speech is severely distorted in subjective listening at both 24 and 48 kHz—because the temporal modeling of waveform-based codecs is tightly coupled to the sampling rate, and a change in sampling rate introduces frequency warping that significantly shifts formant positions and bandwidths. This contrasts with CA2D, whose advantage becomes more pronounced at higher sampling rates, reflecting a direct benefit of the 2D formulation.

4.4 Temporal Axis: Performance Across Input Durations

We vary the input duration while keeping everything else fixed, evaluating 1,000 LibriTTS clips at 16 kHz over 2.55–7.67 s (full five-duration results in Table A7). CA2D maintains strong performance across all evaluated durations. Its measured bitrate remains stable, and its reconstruction quality changes only marginally as the segment length varies (Figure 6). Against BigCodec at 1040 bps, CA2D at around 830 bps, i.e., roughly 20% lower bitrate, achieves comparable or better performance on the evaluated perceptual metrics, including PESQ, ViSQOL-S, and MCD, across the evaluated durations. These results show that CA2D provides competitive quality at a substantially lower bitrate when the input length varies.

4.5 Subjective Evaluation (MUSHRA)

We conduct a MUSHRA listening test (Schoeffler et al., 2018) with 10 human listeners, including the hidden reference and a low-quality anchor at each sampling rate. Figure 7 summarizes the results: the horizontal axis is bitrate, the vertical axis is the MUSHRA score, and the bubble size is proportional to the model parameter count. At 16 kHz, CA2D reaches 92.50 at 837.6 bps and ECC reaches 92.00 at 952.3 bps; both are comparable to the hidden reference (92.42) and outperform BigCodec (89.67), MagiCodec (83.28), FocalCodec (61.89), and DAC (50.50), while CA2D

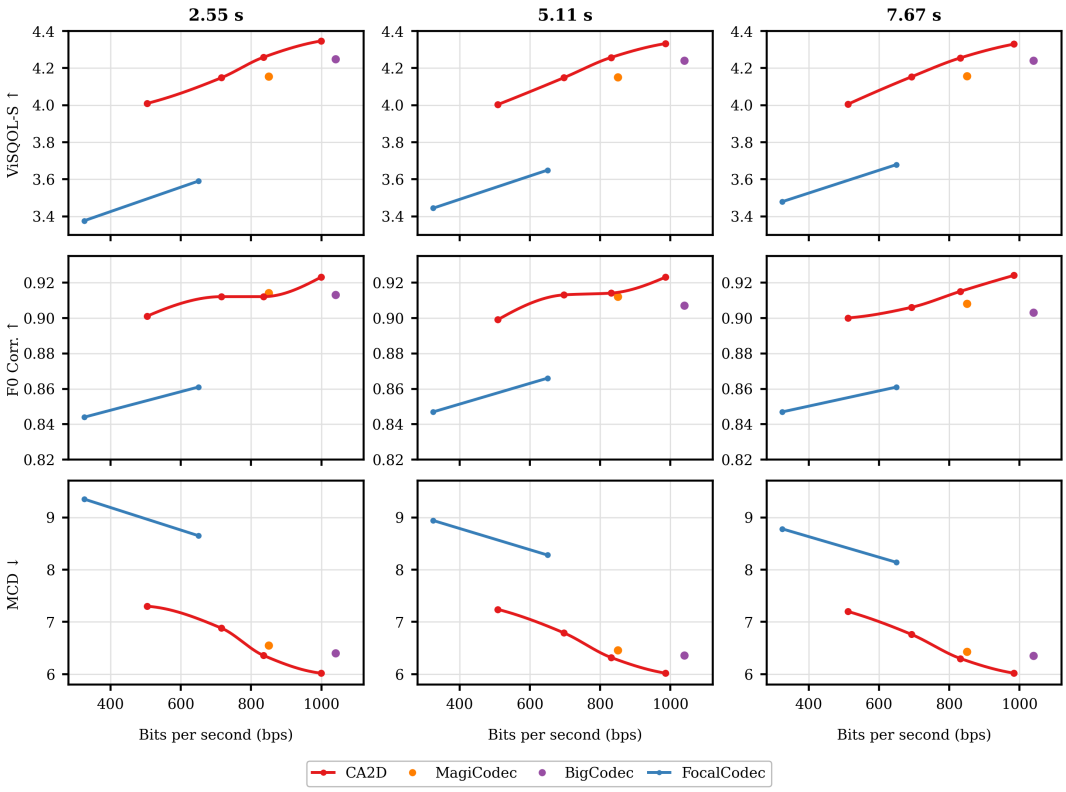


Figure 6: CA2D and baselines vs. input duration at 16 kHz. Rows: metrics; columns: durations. CA2D maintains strong performance across durations.

uses only 69M parameters. At 24 kHz, CA2D obtains 91.92 at 853.5 bps, clearly outperforming MimiCodec (75.22), WavTokenizer (67.50), DAC (63.33), and EnCodec (47.33). At 48 kHz, CA2D reaches 93.39 at 789 bps, close to MDCTCodec at 6000 bps (94.53) and above FlowDec at 6000 bps (90.25) and 3000 bps (67.56). Across all rates, CA2D lies on the low-bitrate frontier with a single shared-parameter model, confirming the subjective benefit of the proposed 2D formulation; its parameter count, MACs, and inference memory across sampling rates are reported in Table A8 (App. A.8).

4.6 Ablation: Entropy Skipping and Token Organization

We isolate the two components with two separate ablations.

Entropy skipping. We compare CA2D with and without entropy-skip training, both sharing the identical 2D transform (Table A9). The skip-disabled variant uses the same base model and the same 30k fine-tuning, with skipping off (App. A.4). As shown in Figure 8, enabling skip training improves both PESQ and ViSQOL at every operating point—up to $+0.10$ PESQ at the lowest rate and $+0.09$ at 1 kbps—while the bitrate stays essentially unchanged. This indicates that skip training omits redundant residuals and allows the model to devote more representational capacity to reducing reconstruction distortion at a similar bitrate. This gain is built on the entropy-model supervision maintained by full-likelihood training. Figure 3(b) compares latent–mean alignment under different training strategies, showing that full-likelihood training preserves the consistent relationship between y and μ, whereas masked-likelihood training breaks this alignment. Since μ and σ are predicted by the same entropy model, a stable latent–mean alignment allows the skip decision, which is based on predicted uncertainty, to reliably omit redundant residuals. Table 1 shows that on the same skip-trained model, turning inference-time skipping from a 98% skip ratio to 0% changes the bitrate by only about 1% and leaves the reconstruction quality essentially unchanged. The difference here comes from the training stage—masked likelihood versus full likelihood—rather than from whether skipping is enabled at inference; for a trained model, toggling skip on/off does not change its latent–mean alignment.

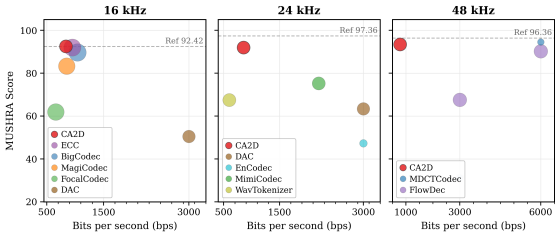


Figure 7: MUSHRA listening test results. Bubble size: parameter count; dashed line: hidden reference.

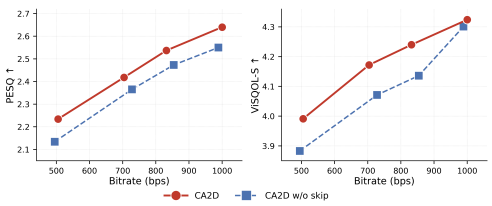


Figure 8: Ablation of entropy skipping: bitrate–PESQ (left) and bitrate–ViSQOL (right) for CA2D with and without skipping.

Compared with masked-likelihood training, full-likelihood training avoids the problem where skipped positions lose probabilistic-model supervision. This training strategy is not specific to CA2D. We further apply the same full-likelihood fix to a reproduction of ECC to verify whether the problem originates from ECC-style masked likelihood itself. Retaining ECC's original masked likelihood causes the bitrate to rise from about 3 kbps to over 23 kbps when skipping is disabled at inference, while reconstruction quality also degrades; switching to full-likelihood training eliminates this failure mode entirely (App. A.1, Table A1). Thus, full-likelihood training addresses a general defect of ECC-style masked likelihood under multi-rate joint training, rather than a patch tailored to the CA2D architecture.

Content-adaptive token organization. We examine the organization strategy at 48 kHz, where the larger frequency dimension makes serialization particularly important: the backbone and training objective are unchanged, and only the organization strategy varies. Table 2 reports the comparison. Both the fixed-order baseline and CA2D are evaluated without entropy skip, avoiding skip interference and isolating token organization. Fixed frequency-first traversal obtains 3.967 ViSQOL-A and 0.876 STOI at 819 bps; content-adaptive organization reaches 3.900 ViSQOL-A and 0.879 STOI at the lower operating point of 724 bps, achieving comparable quality with substantially fewer bits—the fixed traversal spends roughly 95 bps of extra rate for only a marginal quality gain. At the higher operating point of 852 bps, content-adaptive organization further improves to 4.011 and 0.890, clearly outperforming the fixed-order baseline. These results support the motivation for content-adaptive organization: a single fixed traversal imposes identical neighborhoods across the entire field, whereas content-adaptive organization follows feature-dependent structure.

Table 1: Entropy-skip inference toggle at 16 kHz (1,000 clips, 2.55 s).

Skip	Skip ratio	Bitrate	MCD↓	PESQ↑	ViSQOL-S↑	STOI↑
ON	98.58%	704	6.66	2.468	4.176	0.922
OFF	0	712	6.65	2.474	4.180	0.922
ON	98.07%	1000	6.12	2.683	**4.337**	**0.935**
OFF	0	1006	**6.11**	**2.689**	**4.337**	**0.935**

Table 2: Ablation of token organization at 48 kHz (1,000 clips, 2.55 s).

Model	Bitrate	MCD↓	ViSQOL-A↑	STOI↑
Fixed-order	819	5.64	3.967	0.876
CA2D w/o skip	852	5.49	**4.011**	**0.890**
CA2D w/o skip	724	**5.48**	3.900	0.879

5 Conclusion

We have presented CA2D, a neural audio codec that formulates compression as a two-dimensional learned compression problem over complex STFT representations. A single set of parameters covers all three native rates, with the sampling rate absorbed by the frequency-axis resolution rather than by rate-specific branches. We further introduced content-adaptive time-frequency organization that adapts to the asymmetric axes and variable frequency length of the time-frequency latent, analyzed and resolved a masked-likelihood supervision failure in entropy skipping, and merged the entropy-coded streams into a single bitstream. Experiments demonstrate strong low-bitrate rate-distortion performance, competitive perceptual metrics, graceful multi-sampling-rate scaling, and competitive performance across input durations. That these properties follow from treating the time-frequency plane as a spatial field suggests that the learned image compression toolbox, long underexplored for audio, is a promising direction for low-bitrate and multi-rate neural audio coding. Future work may extend the framework to general audio beyond speech, explore entropy models tailored to the time-frequency geometry, and investigate real-time streaming implementations.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the development and debugging of experimental code. For the other tasks requiring disclosure, we have not used generative AI tools, or they are not applicable to this work. Additionally, we used generative AI tools for language polishing, LaTeX formatting, retrieval and discovery of related literature, and discovery of open-source codec implementations and pretrained weights. We have reviewed all AI-assisted work: all AI-assisted text was manually checked and edited by the authors; all AI-assisted code was tested and verified by the authors; and all retrieved resources were manually inspected. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.

Yunuo Chen, Zezheng Lyu, Bing He, Hongwei Hu, Qi Wang, Yuan Tian, Li Song, Wenjun Zhang, and Guo Lu. Cmic: Content-adaptive mamba for learned image compression. *arXiv e-prints*, pp. arXiv–2508, 2025.

CLIC. Workshop and challenge on learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil 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. *Advances in Neural Information Processing Systems*, 38:23742–23767, 2026.

Zhihao Du, Shiliang Zhang, Kai Hu, and Siqi Zheng. Funcodec: A fundamental, reproducible and integrable open-source toolkit for neural speech codec. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 591–595. IEEE, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5708–5717. IEEE, 2022.

Andrew Hines, Jan Skoglund, Anil C Kokaram, and Naomi Harte. Visqol: an objective speech quality model. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):13, 2015.

Jesper Jensen and Cees H Taal. An algorithm for predicting the intelligibility of speech masked by modulated noise maskers. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2009–2022, 2016.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Ruiqi Li, Ziang Zhang, et al. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. In *International Conference on Learning Representations*, volume 2025, pp. 93809–93826, 2025.

Xiao-Hang Jiang, Yang Ai, Rui-Chen Zheng, Hui-Peng Du, Ye-Xin Lu, and Zhen-Hua Ling. Mdct-codec: A lightweight mdct-based neural audio codec towards high sampling rate and low bitrate scenarios. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 540–547. IEEE, 2024.

Eastman Kodak. Kodak lossless true color image suite (photocd pcd0992), 1993. Available from http://r0k.us/graphics/kodak/.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in neural information processing systems*, 36:27980–27993, 2023.

Jingyi Li, Zhiyuan Zhao, Yunfei Liu, Lijian Lin, Ye Zhu, Jiahao Wu, Qiuqiang Kong, and Yu Li. Melcap: A unified single-codebook neural codec for high-fidelity audio compression. 2025.

Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7):1877–1884, 2016.

Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pp. 749–752. IEEE, 2001.

Michael Schoeffler, Sarah Bartoschek, Fabian-Robert Stöter, Marlene Roess, Susanne Westphal, Bernd Edler, and Jürgen Herre. webmushra—a comprehensive framework for web-based listening tests. *Journal of open research software*, 6(1):8–8, 2018.

Yakun Song, Jiawei Chen, Xiaobin Zhuang, Chenpeng Du, Ziyang Ma, Jian Wu, Jian Cong, Dongya Jia, Zhuo Chen, Yuping Wang, et al. Magicodec: Simple masked gaussian-injected codec for high-fidelity reconstruction and generation. *IEEE Transactions on Audio, Speech and Language Processing*, 2026.

Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on audio, speech, and language processing*, 19(7):2125–2136, 2011.

Thilo Thiede, William C Treurniet, Roland Bitto, Christian Schmidmer, Thomas Sporer, John G Beerends, and Catherine Colomes. Peaq-the itu standard for objective measurement of perceived audio quality. *Journal of the Audio Engineering Society*, 48(1/2):3–29, 2000.

Matteo Torcoli, Thorsten Kastner, and Jürgen Herre. Objective measures of perceptual audio quality reviewed: An evaluation of their application domain dependence. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1530–1541, 2021.

Simon Welker, Matthew Le, Ricky TQ Chen, Wei-Ning Hsu, Timo Gerkmann, Alexander Richard, and Yi-Chiao Wu. Flowdec: A flow-based full-band general audio codec with high perceptual quality. In *International Conference on Learning Representations*, volume 2025, pp. 24526–24553, 2025.

Detai Xin, Xu Tan, Shinnosuke Takamichi, and Hiroshi Saruwatari. Bigcodec: Pushing the limits of low-bitrate neural speech codec. *arXiv preprint arXiv:2409.05377*, 2024.

Jun Xu, Zhengxue Cheng, Fengxi Zhang, Yuhan Liu, Li Song, and Wenjun Zhang. Benchmarking neural speech compression from a rate-distortion perspective. *arXiv preprint arXiv:2606.11631*, 2026.

Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). *The Rainbow Passage which the speakers read out can be found in the International Dialects of English Archive:(http://web. ku. edu/˜ idea/readings/rainbow. htm).*, 2019.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco 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. *arXiv preprint arXiv:1904.02882*, 2019.

Zhisheng Zhang, Xiang Li, Yixuan Zhou, Jing Peng, Shengbo Cai, Guoyang Zeng, and Zhiyong Wu. Unisrcodec: Unified and low-bitrate single codebook codec with sub-band reconstruction. *arXiv preprint arXiv:2601.02776*, 2026.

A Appendix

A.1 Entropy-Skip Diagnostic: Masked-Likelihood versus Full-Likelihood Training

This section reports the diagnostic summarized in Sec. 4.6 and illustrated in Figure 3. These experiments are conducted on a reproduction of the ECC model. Both variants are obtained by training the same pretrained model with two different skip schemes at the same high-rate operating point (1,000 clips, 2.55 s, 16 kHz, 80 epochs); "skip off" at inference writes every residual into the bitstream. In Table A1, **bold** marks the best result in each column and underlining marks the second-best result; the Bitrate and skip columns are excluded from this comparison.

Table A1: Entropy-skip toggle under masked-likelihood and full-likelihood training.

Training mode	Skip at inference	Bitrate	LSD ↓	MCD ↓	PESQ ↑	ViSQOL-S ↑	F0 Corr. ↑	F0 RMSE ↓	STOI ↑	ESTOI ↑
Masked-likelihood	ON	3033	8.41	5.24	3.303	4.542	**0.945**	**79.4**	0.962	0.917
Masked-likelihood	OFF	23225	8.53	5.40	3.258	4.516	0.942	80.6	0.959	0.913
Full-likelihood (ours)	ON	3023	8.31	**5.05**	3.328	4.562	0.941	84.6	**0.964**	**0.920**
Full-likelihood (ours)	OFF	3029	**8.30**	**5.05**	**3.329**	**4.563**	0.941	84.4	**0.964**	**0.920**

A.2 Entropy-Skip Inference Toggle

This section reports the inference-time skip toggle at three operating points (Sec. 4.6). All three models are fine-tuned with entropy skipping. At inference, we toggle skipping on and off to analyze model behavior: disabling skipping writes every residual into the bitstream and leaves both bitrate and quality nearly unchanged. In Table A2, **bold** marks the best result in each column and underlining marks the second-best result; the Bitrate and skip-ratio columns are excluded from this comparison.

Table A2: Entropy skipping ON vs. OFF at inference (16 kHz, 1,000 clips of 2.55 s).

Skip	Skip ratio	Bitrate	LSD↓	MCD↓	PESQ↑	ViSQOL-S↑	STOI↑	ESTOI↑
ON	98.58%	704	9.55	6.66	2.468	4.176	0.922	0.839
OFF	0	712	9.51	6.65	2.474	4.180	0.922	0.839
ON	98.31%	832	9.34	6.47	2.584	4.248	0.929	0.852
OFF	0	838	9.32	6.46	2.588	4.250	0.929	0.852
ON	98.07%	1000	9.01	6.12	2.683	**4.337**	**0.935**	**0.863**
OFF	0	1006	**9.00**	**6.11**	**2.689**	**4.337**	**0.935**	**0.863**

A.3 Datasets and Test Set Construction

Training data. The model is trained on LibriTTS (Zen et al., 2019) for the 16 and 24 kHz settings and VCTK (Yamagishi et al., 2019) for 48 kHz. For LibriTTS, we follow the official split. For the VCTK train/test split, we follow the partition strategy of the `badayvedat/VCTK` dataset[1].

[1] `https://huggingface.co/datasets/badayvedat/VCTK`

Full 16 kHz test set. LibriTTS test-clean and test-other contain 9,957 utterances in total. Because the model applies up to $128\times$ temporal downsampling, the spectrogram must be padded to $128n$ frames; consequently, our model requires more padding than models with lower downsampling factors. To ensure a fair comparison, we crop each utterance to $1.28n - 0.01$ s for compatibility with the spatial transform and use the same cropped utterances for all models. This yields 8,866 variable-length utterances. The full test set refers to these 8,866 cropped utterances.

Multi-sampling-rate test set. For the cross-rate experiment we evaluate 1,000 clips of 2.55 s at each of 16, 24, and 48 kHz. The 16 and 24 kHz clips are corresponding LibriTTS utterances and the 48 kHz clips are from VCTK. With a fixed duration, we can observe the bitrate and quality variations of CA2D, BigCodec, and MagiCodec across sampling rates, while also computing MACs/s uniformly for each model.

Input-duration test set. We first select 1,000 LibriTTS utterances longer than 7.67 s and crop a 7.67 s segment from each. The shorter test clips of 6.39, 5.11, 3.83, and 2.55 s are then obtained by successively truncating the same 7.67 s segment. All five durations therefore share identical source content and differ only in length. These durations correspond to $1.28n - 0.01$ s with $n = 2, \ldots, 6$, and the spectrograms of $128n$ frames.

A.4 Training Configuration

STFT configuration. The model input uses a complex STFT with a Hann window. For 16, 24, and 48 kHz speech, the FFT size and hop length are $(510, 160)$, $(766, 240)$, and $(1534, 480)$, respectively. This keeps the frequency spacing at approximately 31.3 Hz and the frame spacing at 10 ms across all sampling rates, matching the physical spacing assumption in the 2D formulation.

The model is trained in two stages: a 16 kHz pretraining stage on LibriTTS for 400k iterations, which gives a stable initialization for the 2D analysis transform and entropy model, followed by multi-sampling-rate joint training on native 16, 24, and 48 kHz speech for a total of 90k iterations. UniSRCodec suggests assigning larger loss weights to low-frequency components to ensure better reconstruction quality. Following this idea, each training cycle in our multi-sampling-rate setup contains an equal number of mini-batches per rate in random order, and the per-rate batch size is GPU-memory-bound and larger at lower rates. Since lower sampling rates predominantly capture low-frequency content, lower-rate data naturally receive more sample-level exposure, thereby assigning a larger effective loss weight to low frequencies. In this way, mixed-rate training implicitly applies a descending loss weight from low to high frequencies, while parameter updates remain comparable across rates.

The training objective follows the rate-distortion formulation of the underlying learned compression framework:

$$\mathcal{L} = R + \lambda D + \lambda_{\mathrm{ortho}} L_{\mathrm{ortho}},$$

where R denotes the estimated latent rate, L_{ortho} the orthogonality regularization associated with the content-adaptive representation, and D the reconstruction objective. Following FreqCodec, we define

$$D = \lambda_t \mathcal{L}_t + \lambda_f \mathcal{L}_f + \lambda_{\mathrm{adv}} \mathcal{L}_{\mathrm{adv}} + \lambda_{\mathrm{feat}} \mathcal{L}_{\mathrm{feat}},$$

where $\mathcal{L}_t$ is the time-domain L1 loss, $\mathcal{L}_f$ is the multi-scale log-power and Mel spectral L1/L2 loss, $\mathcal{L}_{\mathrm{adv}}$ is the hinge adversarial loss, and $\mathcal{L}_{\mathrm{feat}}$ is the feature-matching loss. We set $\lambda_t = 1.0$, $\lambda_f = 1.0$, $\lambda_{\mathrm{adv}} = 1/9$, and $\lambda_{\mathrm{feat}} = 100/9$, following (Du et al., 2024). We use $\lambda \in \{0.08, 0.15, 0.2, 0.3\}$ for the rate-distortion trade-off and $\lambda_{\mathrm{ortho}} = 0.1$. Here, *a single unified model for different sampling rates* should be understood per rate-distortion operating point. Specifically, for each λ we train one model whose parameters are shared across all native sampling rates; different λ values correspond to different bitrate points and therefore yield different models. Within each trained model, the same analysis transform, entropy model, and synthesis transform are applied to 16, 24, and 48 kHz without sampling-rate-specific branches. Thus, the unified aspect is across sampling rates at a fixed operating point, not across all bitrate points.

Optimization uses Adam with a learning rate of 3×10^{-4} for both the generator and the discriminator, with betas $(0.5, 0.9)$. A multi-scale STFT discriminator (FreqCodec) is adapted to the target rate.

Finally, following the generalization-before-skip strategy described in Sec. 3.3, entropy skipping is enabled only after the base model has converged. Starting from this same converged base model, we

fine-tune two variants for an additional 30k iterations under identical optimization settings. The *with entropy skipping* variant enables the skip mechanism during this stage, whereas the *without entropy skipping* variant keeps skipping disabled for the same 30k iterations. Hence, the two variants share the same initialization, training budget, and optimization/data configuration; the only difference is whether entropy skipping is enabled. In Sec. 4.6, these two variants are compared to isolate the effect of entropy skipping from that of additional fine-tuning.

A.5 Evaluation Metric Configurations

To ensure reproducibility, we provide the key implementation details and hyperparameters used for all objective metrics. Throughout all tables, bitrate values are reported in bits per second (bps) when no unit is specified. For entropy-model-based codecs, CA2D and ECC report the actual bitstream length produced by our optimized encoder, whereas all other codecs use their reported bitrates.

Our encoder writes bits directly rather than bytes or 16-bit words. The decoder of the CompressAI range coder must read a fixed 32-bit word to initialize its state before decoding the first symbol, which introduces an unavoidable 4-byte header. In our encoder, the state bits required for decoder initialization are filled directly by the valid coded bits of the bitstream itself, so no separate header is needed. Only a termination bit is written at the end, and the final byte is padded to an 8-bit boundary. Since arithmetic decoding inherently assumes zero bits beyond the end of the stream, this zero-padding matches the encoder-side termination convention exactly, so at most 8 bits of termination information suffice to uniquely determine all symbols and ensure correct decoding.

STFT-based metrics. LSD, LSD-HF, and MCD are computed from STFT-based spectral representations. The STFT parameters, listed in Table A3, are scaled with the sampling rate to maintain a consistent physical frequency resolution of approximately 31.3 Hz/bin:

Table A3: STFT parameters used for LSD, LSD-HF, and MCD.

Sampling rate	n_{fft}	Window length	Hop length
16 kHz	512	512	160
24 kHz	768	768	240
48 kHz	1536	1536	480

A Hann window is used. For LSD, the magnitude spectrum is converted to the logarithmic domain with $\epsilon = 10^{-7}$, and the distance is reported in dB using $20/\ln(10)$. LSD-HF is computed only over the high-frequency band of 8–12 kHz for 24 kHz speech and 8–24 kHz for 48 kHz speech. The frequency bins are selected according to their actual frequencies rather than fixed bin indices.

MCD. We use an 80-channel Slaney mel filterbank spanning 0 to the Nyquist frequency, followed by $\ln(\cdot + 10^{-7})$ and an orthonormal DCT-II. The first 34 cepstral coefficients (c_1–c_{34}) are retained, while the energy coefficient c_0 is discarded. MCD is computed frame by frame without DTW:

$$\text{MCD} = \frac{1}{T}\sum_{t}\sqrt{2\sum_{d=1}^{34}(c_{t,d}^{\text{ref}} - c_{t,d}^{\text{deg}})^2}.$$

F0 metrics. Fundamental frequency is estimated using the DIO algorithm in PyWORLD (Morise et al., 2016) with an F0 range of 71–400 Hz, a frame period of 10 ms, and speed parameter 5. F0 correlation and RMSE are evaluated only on jointly voiced frames. Correlation is computed in the logarithmic F0 domain using Pearson correlation. The F0 RMSE is reported in cents and computed as

$$\text{RMSE}_{\text{cent}} = \text{RMSE}(\log F_0)\frac{1200}{\ln 2}.$$

These F0 metrics therefore characterize pitch tracking only on jointly voiced frames, not voiced/unvoiced decisions or overall prosodic quality.

PESQ. PESQ is evaluated only for the 16 kHz setting using the native 16 kHz wideband configuration.

ViSQOL. For 16 kHz speech, we use the speech-scoring mode with the pretrained SVR speech model. For 24 and 48 kHz speech, we use the audio-scoring mode. Since the audio mode operates at 48 kHz, 24 kHz signals are first resampled to 48 kHz using `soxr`, while 48 kHz signals are evaluated at their native sampling rate. Throughout the tables we abbreviate these two configurations as **ViSQOL-S** (speech-scoring mode, used for all 16 kHz results) and **ViSQOL-A** (audio-scoring mode, used for all 24 and 48 kHz results); the two are not directly comparable because they use different scoring models and different internal sampling rates.

PEAQ-ODG and 2f-model. PEAQ is computed using the `peaqb-fast` Basic implementation, which requires 48 kHz input. Therefore, 24 kHz signals are first upsampled to 48 kHz, while 48 kHz signals are processed natively. We report the resulting Objective Difference Grade (ODG). In addition, the two MOVs `AvgModDiff1b` and `ADBb` are used to compute the 2f-model score:

$$S_{2\mathrm{f}} = \frac{56.1345}{1 + \left(-0.0282\,\mathrm{AvgModDiff1b} - 0.8628\right)^2} - 27.1451\,\mathrm{ADBb} + 86.3515,$$

which is clipped to $[0, 100]$, with higher scores indicating better quality.

STOI and ESTOI. Both metrics are computed using their standard implementations and reported as intelligibility scores, where higher values indicate better intelligibility.

MUSHRA. We conducted a MUSHRA listening test with 10 listeners. For each of 16, 24, and 48 kHz, listeners rated four 2.55-s reconstructed samples. The test included a hidden reference and a low-quality anchor. Results are reported in Sec. 4.5 and Figure 7.

A.6 Baseline Configurations

We evaluate the proposed method against strong baselines across three sampling rates. The choice of baselines is summarized below.

16 kHz. FocalCodec (325/650 bps), DAC (500–6000 bps), ECC (eight rate points within its 224–4875 bps range), MagiCodec (850 bps), and BigCodec (1040 bps) are used. Among them, ECC is the strongest dedicated 16 kHz baseline.

24 kHz. MagiCodec, and BigCodec are also evaluated at 24 kHz. For DAC, we use its official 24 kHz model. In addition, we include EnCodec (1500–24000 bps), WavTokenizer (500/900 bps), and MimiCodec (1100/2200 bps).

48 kHz. We compare with the diffusion-based full-band FlowDec (3000–7500 bps) and MDCT-Codec (1500–6000 bps).

Cross-rate stress-test references. BigCodec and MagiCodec are additionally evaluated at 24 and 48 kHz by force-encoding higher-rate speech with their 16 kHz-trained models. These cross-rate results serve only as stress-test references and are not recommended operating points.

A.7 Full Results on All Evaluation Sets

This section reports the complete rate–distortion results underlying Sec. 4.2, Sec. 4.3 and Sec. 4.4: the full 16 kHz test set (8,866 LibriTTS utterances), the 24 kHz and 48 kHz comparisons (1,000 clips of 2.55 s each), and the variable-length evaluation at five controlled durations (1,000 LibriTTS clips per duration). In all tables, **bold** marks the best result in each column and underlining marks the second-best result; the Model and Bitrate columns are excluded from this comparison.

Table A4: Rate-distortion performance on the full 16 kHz LibriTTS test set (8,866 utterances).

Model	Bitrate	LSD↓	MCD↓	PESQ↑	ViSQOL-S↑	F0 Corr.↑	F0 RMSE↓	STOI↑	ESTOI↑
FocalCodec	325	12.20	9.58	1.239	3.369	0.749	224.3	0.821	0.689
FocalCodec	650	11.82	8.88	1.324	3.578	0.771	205.1	0.850	0.734
DAC	500	14.28	13.55	1.048	2.107	0.131	549.8	0.635	0.421
DAC	1000	12.05	10.88	1.129	2.989	0.356	437.4	0.746	0.586
DAC	1500	11.08	9.32	1.262	3.487	0.567	324.4	0.808	0.679
DAC	3000	9.55	6.77	1.976	4.205	0.826	167.0	0.907	0.827

Continued on next page

Table A4 – continued from previous page

Model	Bitrate	LSD↓	MCD↓	PESQ↑	ViSQOL-S↑	F0 Corr.↑	F0 RMSE↓	STOI↑	ESTOI↑
DAC	6000	8.46	4.46	3.919	4.691	0.943	80.9	0.973	0.945
ECC	224	10.80	9.29	1.394	3.331	0.723	213.0	0.806	0.646
ECC	381	10.25	8.35	1.856	3.702	0.848	144.3	0.872	0.749
ECC	583	9.73	7.51	2.304	3.946	0.862	136.0	0.907	0.810
ECC	923	8.95	6.50	2.693	4.183	0.889	118.9	0.933	0.856
ECC	1497	8.72	6.00	3.028	4.327	0.908	104.8	0.949	0.886
ECC	2056	8.42	5.44	3.367	4.457	0.919	98.4	0.960	0.910
ECC	3716	7.89	4.47	3.885	4.671	0.938	84.6	0.976	0.946
ECC	4875	**7.64**	**4.05**	**4.016**	**4.756**	**0.944**	**79.0**	**0.981**	**0.958**
MagiCodec	850	10.44	6.74	2.368	4.134	0.880	124.1	0.917	0.836
BigCodec	1040	9.06	6.61	2.521	4.225	0.885	121.6	0.929	0.855
CA2D	505	10.45	7.42	2.234	3.991	0.888	123.1	0.904	0.809
CA2D	704	9.52	6.72	2.418	4.172	0.897	115.9	0.920	0.836
CA2D	832	9.37	6.53	2.537	4.240	0.900	114.2	0.928	0.849
CA2D	1000	9.02	6.16	2.640	4.324	0.910	107.2	0.933	0.860

Table A5: Full rate-distortion comparison at 24 kHz (1,000 clips, 2.55 s).

Model	Bitrate	LSD↓	LSD-HF (8–12 kHz)↓	MCD↓	ViSQOL-A↑	F0 Corr.↑	F0 RMSE↓	STOI↑	ESTOI↑	PEAQ-ODG↑	2fmodel↑
DAC	750	11.49	12.16	9.20	3.682	0.635	292.7	0.775	0.605	-3.895	30.26
DAC	1500	10.62	11.60	7.80	3.929	0.838	157.2	0.855	0.735	-3.859	37.02
DAC	3000	9.84	10.94	6.37	4.140	0.930	92.5	0.915	0.838	-3.781	43.92
DAC	6000	9.19	10.47	4.96	4.310	0.953	70.3	0.956	0.914	-3.611	51.91
DAC	9000	8.81	10.27	4.16	4.411	0.964	59.9	0.973	0.946	-3.407	58.26
DAC	12000	8.57	10.16	3.60	4.473	0.966	55.8	0.982	0.965	-3.170	64.05
DAC	24000	**8.02**	9.84	**2.44**	**4.585**	**0.980**	**38.4**	**0.995**	**0.990**	**-2.369**	**80.34**
EnCodec	1500	11.43	12.19	8.68	3.801	0.793	179.2	0.845	0.723	-3.866	35.35
EnCodec	3000	10.80	11.74	7.41	4.010	0.876	133.0	0.899	0.817	-3.819	41.08
EnCodec	6000	10.39	11.44	6.52	4.174	0.916	100.5	0.937	0.882	-3.741	47.08
EnCodec	12000	10.11	11.27	5.82	4.278	0.936	84.8	0.961	0.925	-3.618	53.14
EnCodec	24000	9.97	11.20	5.40	4.317	0.947	75.8	0.972	0.945	-3.505	56.59
WavTokenizer	500	9.79	10.13	7.11	3.968	0.887	122.8	0.854	0.737	-3.858	37.15
WavTokenizer	900	9.25	**9.81**	6.14	4.160	0.918	100.7	0.912	0.826	-3.795	44.06
MagiCodec	1275	9.94	10.83	7.07	3.992	0.876	129.1	0.890	0.780	-3.826	39.90
MimiCodec	1100	11.67	12.48	7.69	3.761	0.893	117.8	0.903	0.807	-3.833	31.75
BigCodec	1560	10.49	12.55	6.86	4.153	0.831	166.5	0.907	0.807	-3.783	44.16
MimiCodec	2200	10.96	11.84	6.73	3.913	0.926	93.9	0.938	0.871	-3.780	35.73
CA2D	494	10.78	11.76	7.38	3.879	0.895	116.5	0.901	0.803	-3.824	38.82
CA2D	720	10.00	11.00	6.50	4.070	0.910	103.9	0.921	0.839	-3.781	44.01
CA2D	848	10.12	11.42	6.46	4.117	0.914	104.3	0.928	0.851	-3.753	45.58
CA2D	1022	9.28	10.18	5.73	4.217	0.922	96.9	0.934	0.862	-3.718	48.64

Table A6: Full rate-distortion comparison at 48 kHz (1,000 clips, 2.55 s).

Model	Bitrate	LSD↓	LSD-HF (8–24 kHz)↓	MCD↓	ViSQOL-A↑	F0 Corr.↑	F0 RMSE↓	STOI↑	ESTOI↑	PEAQ-ODG↑	2fmodel↑
FlowDec	3000	9.15	9.18	5.97	3.498	0.949	55.0	0.885	0.760	-3.803	42.35
FlowDec	4000	8.80	8.90	5.16	3.671	0.956	48.4	0.926	0.841	-3.722	48.92
FlowDec	4500	8.87	8.97	5.19	3.642	0.958	49.4	0.916	0.821	-3.745	46.13
FlowDec	6000	8.73	8.89	4.66	3.724	0.965	44.0	0.939	0.867	-3.677	49.30
FlowDec	7500	8.70	8.89	4.43	3.758	**0.967**	**40.4**	**0.951**	**0.893**	-3.640	50.86
BigCodec	3120	10.51	11.19	6.89	3.757	0.703	229.6	0.802	0.575	-3.859	39.13
MagiCodec	2550	9.54	9.73	7.08	3.742	0.626	296.4	0.784	0.549	-3.866	37.66
MDCTCodec	1500	12.74	11.29	11.15	2.888	0.883	84.5	0.695	0.482	-3.903	13.01
MDCTCodec	3000	10.96	10.08	8.50	3.291	0.936	62.1	0.834	0.684	-3.878	24.29
MDCTCodec	4500	9.01	8.78	5.87	3.911	0.949	55.5	0.903	0.796	-3.780	40.61
MDCTCodec	6000	8.28	8.36	**4.30**	**4.279**	0.946	54.9	0.927	0.838	**-3.567**	**56.96**
CA2D	463	9.02	8.97	5.99	3.812	0.931	67.3	0.860	0.743	-3.792	43.95
CA2D	681	8.74	8.73	5.50	3.905	0.935	65.6	0.881	0.764	-3.751	48.50
CA2D	821	8.60	8.57	5.37	4.018	0.934	61.3	0.890	0.774	-3.729	49.54
CA2D	978	**8.10**	**8.11**	4.59	4.154	0.938	61.6	0.899	0.785	-3.668	52.91

Table A7: CA2D and baselines at five controlled input durations (1,000 LibriTTS clips each).

Duration	Model	Bitrate	LSD↓	MCD↓	PESQ↑	ViSQOL-S↑	F0 Corr.↑	F0 RMSE↓	STOI↑	ESTOI↑
2.55 s	FocalCodec	325	12.06	9.35	1.227	3.376	0.844	198.9	0.826	0.697

Continued on next page

Table A7 – continued from previous page

Duration	Model	Bitrate	LSD↓	MCD↓	PESQ↑	ViSQOL-S↑	F0 Corr.↑	F0 RMSE↓	STOI↑	ESTOI↑
	FocalCodec	650	11.69	8.65	1.315	3.590	0.861	177.3	0.856	0.742
	MagiCodec	850	10.34	6.55	2.432	4.154	0.914	105.6	0.922	0.845
	BigCodec	1040	8.89	6.40	2.572	4.248	0.913	105.6	0.935	0.864
	CA2D	504	10.34	7.30	2.297	4.009	0.901	116.3	0.910	0.818
	CA2D	716	9.75	6.88	2.544	4.148	0.912	107.0	0.925	0.846
	CA2D	835	9.16	6.36	2.619	4.258	0.912	107.1	0.932	0.857
	CA2D	999	**8.87**	**6.02**	**2.708**	**4.346**	**0.923**	**97.4**	**0.937**	**0.867**
3.83 s	FocalCodec	325	11.88	9.09	1.254	3.420	0.842	191.2	0.830	0.703
	FocalCodec	650	11.47	8.40	1.350	3.628	0.863	170.3	0.859	0.746
	MagiCodec	850	10.26	6.49	2.457	4.154	0.907	113.4	0.923	0.844
	BigCodec	1040	**8.88**	6.37	2.600	4.244	0.910	110.9	0.935	0.863
	CA2D	503	10.47	7.26	2.295	4.006	0.897	123.7	0.909	0.815
	CA2D	702	9.78	6.80	2.543	4.153	0.910	110.8	0.924	0.843
	CA2D	829	9.20	6.33	2.601	4.258	0.913	108.4	0.931	0.853
	CA2D	991	**8.88**	**6.01**	**2.707**	**4.341**	**0.924**	**100.4**	**0.937**	**0.864**
5.11 s	FocalCodec	325	11.77	8.94	1.267	3.444	0.847	186.4	0.834	0.706
	FocalCodec	650	11.37	8.28	1.368	3.649	0.866	168.8	0.862	0.749
	MagiCodec	850	10.23	6.46	2.475	4.151	0.912	114.2	0.923	0.843
	BigCodec	1040	**8.87**	6.36	2.625	4.240	0.907	116.6	0.935	**0.862**
	CA2D	509	10.52	7.24	2.293	4.003	0.899	125.7	0.908	0.813
	CA2D	697	9.86	6.79	2.511	4.148	0.913	111.2	0.923	0.840
	CA2D	831	9.22	6.32	2.599	4.257	0.914	110.1	0.931	0.853
	CA2D	986	8.93	**6.02**	**2.685**	**4.332**	**0.923**	**105.2**	**0.936**	**0.862**
6.39 s	FocalCodec	325	11.72	8.85	1.275	3.462	0.847	186.2	0.835	0.707
	FocalCodec	650	11.33	8.20	1.387	3.671	0.863	166.2	0.863	0.751
	MagiCodec	850	10.20	6.44	2.487	4.155	0.909	116.5	0.923	0.843
	BigCodec	1040	**8.87**	6.35	2.624	4.242	0.905	118.0	0.934	**0.861**
	CA2D	510	10.52	7.21	2.294	4.006	0.902	123.4	0.908	0.812
	CA2D	693	9.93	6.78	2.498	4.151	0.908	114.0	0.921	0.837
	CA2D	831	9.24	6.31	2.591	4.257	0.916	110.5	0.930	0.851
	CA2D	983	8.95	**6.03**	**2.681**	**4.331**	**0.926**	**104.2**	**0.935**	**0.861**
7.67 s	FocalCodec	325	11.67	8.78	1.286	3.479	0.847	183.2	0.836	0.710
	FocalCodec	650	11.29	8.14	1.399	3.679	0.861	165.3	0.864	0.753
	MagiCodec	850	10.18	6.43	2.484	4.156	0.908	117.8	0.923	0.843
	BigCodec	1040	**8.87**	6.35	2.629	4.241	0.903	120.7	0.934	**0.861**
	CA2D	512	10.57	7.20	2.285	4.005	0.900	124.8	0.907	0.812
	CA2D	693	9.93	6.76	2.487	4.153	0.906	116.8	0.921	0.837
	CA2D	832	9.22	6.30	2.581	4.255	0.915	112.6	0.930	0.851
	CA2D	984	8.92	**6.02**	**2.671**	**4.329**	**0.924**	**105.2**	**0.935**	**0.861**

A.8 Model Complexity and Computational Scaling

This section reports the complexity measurements underlying the scaling analysis in Sec. 4.5: parameter counts, per-second MACs, and GPU memory. All measurements are performed on an NVIDIA 4090 and averaged over 1,000 clips of 2.55 s. CA2D shares the same architecture and parameter count across all sampling rates, while its computational cost grows with the time-frequency resolution.

CA2D contains 69M parameters, and this count remains identical at 16, 24, and 48 kHz. Its computational cost, however, grows with the spatial resolution of the STFT. This is the primary computational trade-off of the spatial formulation: higher sampling rates process a larger time-frequency field without introducing additional parameters. At 48 kHz, the cost remains well below that of FlowDec, whose configurations require 4.5–5.4T MACs. Figure A1 illustrates the parameter–memory trade-off across all three rates: CA2D maintains a constant parameter count while requiring less inference GPU memory than baselines with comparable or larger parameter counts.

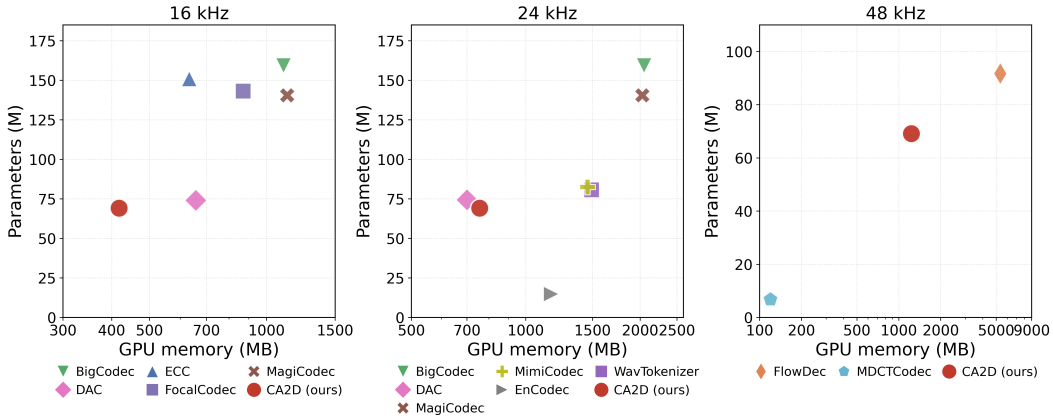


Figure A1: Parameters versus inference GPU memory for the compared codecs at 16, 24, and 48 kHz; CA2D is highlighted.

Table A8: Model complexity at all three sampling rates.

Rate	Model	Params (M)	MACs/s (G)	GPU mem (MB)
16 kHz	BigCodec	159	61	1104
	DAC	74	56	658
	ECC	151	17	633
	FocalCodec_50	142	9	866
	FocalCodec_25	144	9	876
	MagiCodec	141	7	1128
	CA2D	69	29	418
24 kHz	BigCodec	159	92	2049
	DAC	74	84	700
	MagiCodec	141	11	2026
	MimiCodec	81	10	1454
	EnCodec	15	6	1166
	WavTokenizer	81	3	1490
	WavTokenizer	81	6	1490
	CA2D	69	44	757
48 kHz	FlowDec_75m	88	4485	4846
	FlowDec_25s	95	5378	5860
	MDCTCodec	7	7	120
	CA2D	69	88	1229

A.9 CA2D versus CA2D without Skip on the Complete 16 kHz Test Set

This section reports the full ablation behind Sec. 4.6. CA2D and CA2D w/o skip share the same base model and 30k fine-tuning; only skip differs. This is a different condition from the inference toggle of Table 1, where a single trained model is decoded with skipping switched off, so the two sets of numbers are not expected to coincide.

Table A9: CA2D versus CA2D without skip on the full 16 kHz test set (8,866 utterances).

Model	Bitrate	LSD↓	MCD↓	PESQ↑	ViSQOL-S↑	F0 Corr.↑	F0 RMSE↓	STOI↑	ESTOI↑
CA2D w/o skip	495	10.87	7.81	2.134	3.883	0.877	135.6	0.900	0.802
CA2D	505	10.45	7.42	2.234	3.991	0.888	123.1	0.904	0.809
CA2D w/o skip	728	9.54	6.76	2.365	4.071	0.897	118.3	0.919	0.835
CA2D	704	9.52	6.72	2.418	4.172	0.897	115.9	0.920	0.836
CA2D w/o skip	854	9.45	6.75	2.473	4.136	0.905	109.6	0.928	0.851
CA2D	832	9.37	6.53	2.537	4.240	0.900	114.2	0.928	0.849
CA2D w/o skip	989	9.00	6.22	2.550	4.301	0.906	109.9	0.932	0.857
CA2D	1000	9.02	6.16	2.640	4.324	0.910	107.2	0.933	0.860