AUDIO TOKENS AS A BUDGETED RESOURCE: MARGINAL-UTILITY ALLOCATION FOR SCALABLE AUDIO REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete audio tokens are widely used as a representation interface, yet fixed-depth RVQ tokenizers allocate equal capacity to every frame despite varying refinement value. We introduce UniAdapt, which learns marginal utility of RVQ refinements on a frozen codec and allocates them under exact serialized-bit budgets. A rate-independent causal controller predicts acoustic utility, while an optional semantic head supports speech-only utterance-level allocation; measured acoustic and semantic marginal gains on speech have a correlation of 0.42. For causal allocation, a primal–dual allocator selects prefix-valid depths, while an exact guard constrains each sequence prefix to its matched fixed-depth serialized budget. Under utterance-level allocation, UniAdapt reduces Log-STFT distortion by 1.07–4.39% across speech, music, and environmental audio without larger budgets. Causally, it improves three of four speech rates with zero violations across 800 utterance–rate evaluations and runs faster than real time. A 20-listener utterance-level MUSHRA study shows a significant 3.52-point speech improvement, with no significant differences on music or environmental audio. These results support separating utility prediction from budget enforcement for scalable, budget-conditioned audio representations.

1 INTRODUCTION

Discrete audio tokens are increasingly used as a common interface for speech and audio generation, understanding, and interactive systems (Zhang et al., 2024; Huang et al., 2024; Ji et al., 2025; Défossez et al., 2024). Residual vector quantization (RVQ) is particularly attractive because it exposes a hierarchy of discrete refinements (Zeghidour et al., 2022; Défossez et al., 2023; Kumar et al., 2023). Yet fixed-depth RVQ activates the same number of layers at every frame for a given operating point, implicitly allocating equal capacity over time. Real audio violates this assumption: an extra refinement can be valuable near a phonetic transition or transient event but largely redundant in a stationary region. This raises a simple question: *given a limited representation budget, where is the next RVQ refinement worth spending?*

Existing work addresses different parts of this problem. Variable-rate and content-adaptive codecs relax uniform allocation (Chae et al., 2025; Zhang et al., 2025; Li et al., 2026), while semantic-aware tokenizers show that acoustic fidelity and semantic preservation need not favor the same representation (Zhang et al., 2024; Huang et al., 2024; Li et al., 2025; Zhao et al., 2026b). Most closely related, BAMU (Zhao et al., 2026a) already introduced rate-independent marginal-utility prediction and exact realized-container budget enforcement for frozen speech codecs, using a non-causal predictor and full-utterance allocation under an exact serialized-size budget. SPG-Codec (Zhao et al., 2026b) showed that the benefit of frozen semantic priors depends strongly on bitrate, but incorporated semantics during codec training rather than post-hoc token allocation. UniStream (Zhao & Wu, 2026) provides the frozen causal 48-kHz RVQ hierarchy used here for speech, music, and environmental audio, but does not determine how its capacity should vary over time. Building on these directions, UniAdapt focuses on causal depth decisions under exact per-prefix budgets and on objective-dependent allocation over the same frozen hierarchy. This distinction matters in streaming: decisions are irreversible, while the true transmitted cost includes code indices, depth

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.

maps, expert side information, headers, run-length coding, and byte padding, so an utterance-level budget does not ensure prefix feasibility.

Our central idea is to treat audio tokens as a *budgeted resource*: predict token value independently of the target rate, choose the representation objective, and introduce the resource constraint only when capacity is allocated. Rather than predicting a bitrate-specific depth, UniAdapt estimates the marginal utility of activating one additional RVQ refinement at each frame and layer. On speech, measured acoustic and HuBERT-based semantic marginal gains have an overall correlation of 0.4205, falling to near zero or slightly negative values in deeper layers, which motivates separate objective-dependent utility models. The causal streaming system uses acoustic utility only; an optional semantic head is used for speech-only, utterance-level objective-dependent allocation.

We instantiate this principle on a frozen causal universal audio codec. A lightweight causal controller predicts frame- and layer-wise acoustic marginal utilities without receiving the target rate. For online operation, a causal primal–dual allocator converts these utilities into prefix-valid RVQ depths under token-bucket and switching controls. Before committing each depth, an exact feasibility guard evaluates the realized serialized prefix cost against the matched fixed-depth prefix budget and additionally checks that the next frame can still transmit the mandatory minimum depth $d = 1$. Depth and expert maps, the header, and byte padding are all included in the accounting. The committed depths define active prefixes of the same frozen RVQ hierarchy, yielding a fully decodable dynamic representation with the same codebooks and decoder rather than separately trained rate-specific tokenizers.

Experiments keep utterance-level and causal settings separate. Under utterance-level exact-budget allocation, one three-domain acoustic controller reduces Log-STFT distortion by 1.07–4.39% across speech, music, and environmental audio without exceeding matched fixed-depth budgets. In the causal setting, UniAdapt improves three of four speech rates, produces zero budget violations across 800 utterance–rate evaluations, and runs complete waveform-to-waveform streaming faster than real time. A 20-listener MUSHRA study using utterance-level allocation shows a significant 3.52-point speech improvement, with no significant difference on music or environmental audio. Frozen-token evaluations show significant CER reductions under a shared multi-rate CTC probe, higher FSD50K macro-mAP, and lower speaker-verification EER at depths 3, 4, and 6. At lower realized bitrates, the main MTG-Jamendo tagging and Clotho mean-recall metrics show no statistically significant differences. On LibriSpeech, dynamic allocation improves PESQ and STOI over the same frozen backbone at a slightly lower rate, although EnCodec retains higher absolute reconstruction quality at a similar bitrate. These results support content-adaptive allocation under explicit serialized-bit constraints rather than a claim of reconstruction state of the art.

Contributions. Our contributions are fourfold:

- We extend marginal-utility allocation on frozen RVQ hierarchies from utterance-level optimization to **causal, budget-conditioned token allocation**, using a rate-independent causal predictor across operating rates.
- We model **objective-dependent marginal utility** on the same frozen hierarchy, with acoustic utility for causal operation and optional semantic utility for speech-only utterance-level allocation.
- We develop a **causal exact-budget allocator** combining primal–dual rate control with serialized per-prefix feasibility checks, keeping each committed prefix within its matched fixed-depth budget while ensuring next-frame minimum-depth feasibility.
- We validate the resulting **budget-conditioned frozen-token representation** across multiple rates and audio domains, downstream tasks, human listening, and real-time waveform streaming.

2 Related Work

2.1 Neural Audio Codecs and Adaptive Tokenization

RVQ-based codecs such as SoundStream (Zeghidour et al., 2022), EnCodec (Défossez et al., 2023), and DAC (Kumar et al., 2023) establish hierarchical residual quantization for neural audio coding.

Related codec architectures explore multi-scale quantization, low token rates, and causal universal audio coding (Siuzdak et al., 2024; Défossez et al., 2024; Ji et al., 2025; Zhao & Wu, 2026). These methods primarily develop representation architectures and their supported operating points. UniAdapt instead studies how existing RVQ refinements should be allocated across frames under explicit serialized-bit budgets. Adaptive methods vary token usage through active RVQ depth or temporal resolution (Chae et al., 2025; Zhang et al., 2025; Li et al., 2026). BAMU (Zhao et al., 2026a) is most closely related: it learns rate-independent acoustic marginal utilities on frozen speech codecs using a non-causal predictor and full-utterance allocation under an exact serialized-size budget. UniAdapt extends this formulation to causal, irreversible per-frame decisions under per-prefix budgets, while separately studying objective-dependent utility. UniStream (Zhao & Wu, 2026) supplies the causal universal RVQ hierarchy used here but does not address content-adaptive per-frame depth.

2.2 Semantic-Aware Audio Representations

Semantic-aware tokenizers incorporate linguistic information through different mechanisms. SpeechTokenizer (Zhang et al., 2024) organizes semantic and acoustic information across RVQ layers. RepCodec (Huang et al., 2024) learns discrete semantic tokens by reconstructing continuous representations from pretrained speech encoders, whereas DualCodec (Li et al., 2025) integrates self-supervised speech features and waveform representations through dual-stream encoding. SPG-Codec (Zhao et al., 2026b) further shows that the usefulness of frozen semantic priors varies with bitrate, but incorporates semantic constraints during codec training. UniAdapt instead keeps the codec and RVQ hierarchy frozen and uses semantic preservation only to define the marginal value of transmitting an additional existing refinement. This semantic branch is restricted to speech and utterance-level allocation; causal streaming uses acoustic utility only. UniAdapt therefore studies objective-dependent allocation rather than semantic disentanglement.

2.3 Causal and Exact-Budget Allocation

Streaming turns adaptive tokenization into an irreversible online resource-allocation problem. Nominal bitrate or average RVQ depth alone does not determine the exact serialized size. In the dynamic container used here, the cost includes active code indices, run-length-coded depth and expert maps, the header, and byte padding. BAMU (Zhao et al., 2026a) enforces an exact utterance-level container budget, but its full-sequence optimization does not constrain every intermediate prefix. UniAdapt instead requires each committed prefix to remain within its matched fixed-depth serialized budget while retaining minimum-depth $d = 1$ feasibility for the next frame.

Together, these lines motivate allocation over a frozen RVQ hierarchy with causal acoustic-utility prediction and exact per-prefix budget enforcement, alongside an offline speech setting for objective-dependent acoustic–semantic allocation. UniAdapt targets this intersection rather than a new reconstruction backbone.

3 Method

3.1 Frozen Hierarchy and Problem Formulation

As shown in Fig. 1(a), UniAdapt operates on a frozen causal RVQ codec. The encoder produces latent frames

$$z = E(x) = \{z_t\}_{t=1}^{T}, \qquad z_t \in \mathbb{R}^C, \tag{1}$$

and the frozen quantizer input projection initializes the residual,

$$v_t = P_{\text{in}}(z_t) \in \mathbb{R}^{C_q}, \qquad r_{t,0} = v_t, \qquad r_{t,k} = r_{t,k-1} - q_{t,k}, \tag{2}$$

where C_q is the quantization-space dimensionality. UniAdapt selects $d_t \in \{1, \dots, K\}$ and uses $\tilde{v}_t(d_t) = \sum_{k=1}^{d_t} q_{t,k}$. The encoder, input projection, codebooks, routing modules, and decoder remain frozen. We use the reversible Top-1 shared-expert mode: expert assignments are derived from the frozen router, shared across acoustic RVQ layers within fixed temporal blocks, and serialized rather than optimized by the depth allocator.

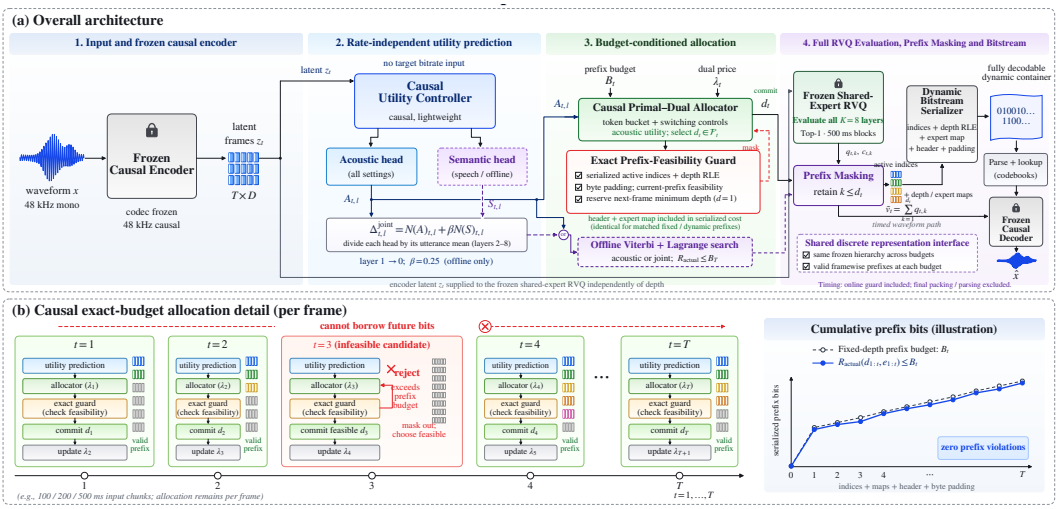

Figure 1: **UniAdapt framework.** (a) A frozen causal RVQ codec provides a shared hierarchy of audio refinements. A rate-independent causal controller predicts acoustic marginal utilities for all settings, with an optional semantic head for speech-only utterance-level allocation. A budget-conditioned allocator selects framewise prefix depths under exact serialized-prefix feasibility. The reported real-backbone path evaluates all RVQ layers and applies the selected prefix mask before serialization and frozen causal decoding. (b) Causal allocation is irreversible: each committed prefix must satisfy its matched fixed-depth serialized budget, while reserving minimum-depth feasibility for the next frame.

Let $R_{\text{actual}}(d_{1:t}, e_{1:t})$ denote the size obtained by serializing a sequence prefix, including active indices, depth and expert maps, the header, and byte padding. For matched fixed depth D, using the same expert map,

$$B_t = R^{\text{fixed}}_{\text{actual}}(D, t). \tag{3}$$

We consider

$$\max_{\{d_t\}} \sum_{t=1}^{T} U_t(d_t) - \rho \sum_{t=2}^{T} \mathbf{1}[d_t \neq d_{t-1}] \quad \text{s.t.} \quad R_{\text{actual}}(d_{1:\tau}, e_{1:\tau}) \le B_\tau, \ \forall \tau. \tag{4}$$

Equation (4) defines the constrained objective. Given the supplied budget reference and target rate, online decisions depend only on $z_{\le t}$ and persistent allocator state; the utility predictor itself receives no target-rate input.

3.2 Rate-Independent Acoustic and Semantic Marginal Utility

The controller predicts refinement value rather than bitrate-specific depth. Acoustic utility is measured in the projected RVQ space:

$$D^{\text{ac}}_{t,k} = \frac{1}{C_q} \|r_{t,k}\|_2^2, \qquad \Delta^{\text{ac}}_{t,k} = \left[D^{\text{ac}}_{t,k-1} - D^{\text{ac}}_{t,k}\right]_+, \quad k = 1, \dots, K. \tag{5}$$

For speech, a frozen HuBERT teacher Φ provides semantic targets. Let $\hat{x}^{(k)}$ be reconstruction at uniform depth k, and let j index the HuBERT frames:

$$D^{\text{sem}}_{j,k} = 1 - \cos\left(\Phi_j(x), \Phi_j(\hat{x}^{(k)})\right). \tag{6}$$

For $k \ge 2$, differencing and clipping precede linear interpolation to the codec grid:

$$\Delta^{\text{sem}}_{t,k} = \mathcal{I}_{J \to T}\left(\left[D^{\text{sem}}_{\cdot,k-1} - D^{\text{sem}}_{\cdot,k}\right]_+\right)_t, \qquad \Delta^{\text{sem}}_{t,1} = 0. \tag{7}$$

Here $\mathcal{I}$ uses linear interpolation with `align_corners=False`. These teacher computations are offline.

A causal trunk predicts non-negative utilities for all K layers:

$$h_t = f_\theta(z_{\le t}), \qquad \hat{\Delta}^o_{t,k} = \mathrm{softplus}(g_o(h_t)_k), \quad o \in \{\mathrm{ac}, \mathrm{sem}\}, \tag{8}$$

without target-bitrate input. Layer 1 is mandatory; its predicted contribution is identical across candidate depths and can therefore be omitted from the allocation objective without changing depth selection.

Main acoustic-only and causal experiments use raw $\hat{\Delta}^{\mathrm{ac}}$. Semantic-aware allocation is speech-only and utterance-level, with each branch normalized separately:

$$s_o = \max\left(\frac{1}{T(K-1)}\sum_{t=1}^{T}\sum_{k=2}^{K}\hat{\Delta}^o_{t,k}, 10^{-8}\right), \qquad \bar{\Delta}^o_{t,k} = \begin{cases} 0, & k = 1, \\ \hat{\Delta}^o_{t,k}/s_o, & k = 2, \dots, K. \end{cases} \tag{9}$$

Joint utility is

$$\Delta^{\mathrm{joint}}_{t,k} = \bar{\Delta}^{\mathrm{ac}}_{t,k} + \beta\bar{\Delta}^{\mathrm{sem}}_{t,k}, \qquad \beta = 0.25. \tag{10}$$

In semantic-weight sweeps, $\beta = 0$ denotes the normalized acoustic reference, not the main raw-acoustic system. Equation (9) uses complete-utterance statistics and is never used in causal streaming. For the selected utility,

$$U_t(d) = \sum_{k=2}^{d} u_{t,k}, \qquad U_t(1) = 0. \tag{11}$$

3.3 Causal Exact-Budget Allocation

At frame t, candidate depth d receives

$$S_t(d) = U_t(d) - \lambda_t c_t(d) - \rho\,\mathbf{1}[d \neq d_{t-1}], \tag{12}$$

where $c_t(d) = 10d$ is the code-index cost used as a rate-control surrogate in the evaluated online configuration. The target per-frame cost is $c = 1000R_{\mathrm{target}}/f_{\mathrm{frame}}$, where R_{target} is in kbps. Evaluation supplies this rate from the matched fixed-depth serialized reference. The dual update is

$$\lambda_{t+1} = \left[\lambda_t + \eta\frac{c_t(d_t) - c}{\max(c, 1)}\right]_+. \tag{13}$$

A token bucket, dwell constraint, transition penalty, and hysteresis control short-term rate and switching. No approximate depth-side cost is added to $c_t(d)$ because the exact guard already accounts for depth-map RLE.

Exact serializer-consistent accounting uses

$$R_{\mathrm{actual}} = H + 8\left\lceil\frac{R_{\mathrm{code}}}{8}\right\rceil + 8\left\lceil\frac{R^{\mathrm{RLE}}_{\mathrm{depth}}}{8}\right\rceil + 8\left\lceil\frac{R^{\mathrm{RLE}}_{\mathrm{expert}}}{8}\right\rceil, \tag{14}$$

where H is the fixed container-header cost and $R_{\mathrm{code}}(d_{1:t}) = 10\sum_{i=1}^{t} d_i$ for 1024-entry codebooks. The three variable payloads are independently byte aligned. Matched fixed and dynamic prefixes have identical header and expert-map costs. Consequently, the guard compares exact code-plus-depth payload sizes without knowing the next expert assignment, while still enforcing the complete serialized budget.

Let $\tilde{R}$ denote this code-plus-depth payload and $\tilde{B}_t = \tilde{R}^{\mathrm{fixed}}(D, t)$. Before commitment,

$$\mathcal{F}_t = \left\{d \in \{1, \dots, K\} : \tilde{R}(d_{1:t-1}, d) \le \tilde{B}_t,\ \tilde{R}(d_{1:t-1}, d, 1) \le \tilde{B}_{t+1}\right\}. \tag{15}$$

The second condition checks payload feasibility for a hypothetical next frame at minimum depth $d = 1$. Bucket affordability and dwell further restrict admissible actions; minimum depth is exempt from the bucket mask. The highest-scoring admissible depth is selected, with hysteresis retaining d_{t-1} when it remains admissible and the score improvement is below its margin. Dwell is released when retaining the previous depth would violate exact feasibility. Exact feasibility therefore takes precedence over all soft switching controls.

Utterance-level experiments instead use full-sequence Lagrange–Viterbi allocation with raw acoustic or normalized joint utility, followed by exact serialized-budget checking. This offline branch is not used for causal streaming.

3.4 Prefix Representation and Dynamic Bitstream

A committed depth defines the active RVQ prefix. With fixed expert assignments, prefix early exit is numerically equivalent to full RVQ evaluation followed by masking. Our reported real-backbone evaluations use full RVQ evaluation and the selected prefix mask; runtime results therefore do not rely on computational early exit.

The dynamic container stores the fixed header, RLE-compressed depth and block-shared expert maps, and packed active indices. The frozen decoder reconstructs

$$\hat{x} = G\big(\{\tilde{v}_t(d_t)\}_{t=1}^{T}\big)\,. \tag{16}$$

All metadata costs are included in R_{actual}. Different budgets select framewise prefixes of the same frozen hierarchy rather than separate rate-specific tokenizers.

Depth decisions are causal per latent frame. Shared-expert quantization buffers 75 frames (500 ms), using the majority Top-1 router assignment within each block. Encoder, controller, allocator, and decoder states persist across input chunks, so allocation does not reset at chunk boundaries. Reported streaming timings include online exact-budget accounting but exclude final byte packing and container parsing.

3.5 Learning Objective and Training Protocol

The codec and HuBERT teacher remain frozen. Utilities are trained directly against non-negative marginal labels:

$$\mathcal{L}_o = \frac{1}{|\Omega|} \sum_{(t,k)\in\Omega} \mathrm{SmoothL1}\left(\hat{\Delta}^{o}_{t,k}, \Delta^{o}_{t,k}\right), \quad o \in \{\mathrm{ac}, \mathrm{sem}\}, \tag{17}$$

where Ω contains non-padded positions and all K outputs.

We first train the acoustic controller. For semantic-head training, we initialize from this checkpoint, freeze the shared causal trunk and acoustic and speech-probability heads, and optimize only the semantic head. Its regression loss is weighted by 20, distinct from the allocation weight $\beta = 0.25$. HuBERT is absent from token extraction and deployment. Architecture, data sizes, allocator hyperparameters, and streaming settings are reported in the Appendix and experimental setup.

4 Experiments

4.1 Experimental Setup

Codec and controllers. We use the frozen 48-kHz causal UniStream codec (Zhao & Wu, 2026), with eight RVQ layers, 1024 entries per codebook, and a latent rate of 150 Hz. We use reversible Top-1 shared-expert quantization with one expert assignment per 500-ms block. The codec remains frozen and only the utility controllers are trained. We evaluate target depths $D \in \{2, 3, 4, 6\}$, corresponding to approximately 3.1, 4.6, 6.1, and 9.1 kbps under fixed-depth coding. All UniAdapt rates are measured from the actual serialized container, including active indices, depth RLE, expert RLE, the header, and byte padding.

The speech acoustic controller uses 5,000 LibriSpeech train-clean-100 utterances (Panayotov et al., 2015). The three-domain controller uses 3,000 examples, equally drawn from LibriSpeech, MTG-Jamendo (Bogdanov et al., 2019), and FSD50K (Fonseca et al., 2022). Semantic targets are generated from 1,000 LibriSpeech utterances using frozen HuBERT. The three-domain offline evaluation uses 200 LibriSpeech test-clean utterances, 50 MUSDB18-HQ (Rafii et al., 2019) test examples, and 50 FSD50K test examples. Online allocation, online ablations, and streaming use the speech 5K controller; the same 200 LibriSpeech test-clean utterances are evaluated at all four online rates.

Evaluation and statistics. Reconstruction is evaluated using multi-resolution Log-STFT distance; speech additionally uses PESQ and STOI (Rix et al., 2001; Taal et al., 2011). Downstream probes use five independent seeds (2040–2044), with paired 95% Student-t intervals over seeds ($df = 4$). These intervals characterize training-seed variability on a fixed evaluation set. Frozen external ASR

Table 1: Relative Log-STFT change of learned dynamic allocation versus fixed depth under matched serialized-bit budgets. Results use utterance-level offline allocation with the three-domain 3K controller. Percentages are computed from the ratio of mean distortions; lower is better.

Domain	$D = 2$	$D = 3$	$D = 4$	$D = 6$
Speech	-3.82%	-4.39%	-4.22%	-3.34%
Music	-1.43%	-1.16%	-1.64%	-1.77%
Environment	-1.53%	-1.07%	-1.61%	-1.34%

Table 2: Causal online allocation relative to matched fixed-depth coding. Percentages are averages of per-utterance relative changes. Negative rate and Log-STFT changes indicate lower bitrate and lower distortion, respectively.

Target	**Rate** Δ	**Log-STFT** Δ	**Switches/s**	**Violations**
$D = 2$	-3.56%	-0.70%	12.45	0/200
$D = 3$	-2.73%	$+0.55\%$	17.14	0/200
$D = 4$	-2.86%	-0.91%	9.10	0/200
$D = 6$	-3.13%	-1.01%	8.11	0/200

uses 10,000 paired bootstrap resamples clustered by speaker, while external-codec comparisons use 10,000 utterance-paired bootstrap resamples. MUSHRA uses the listener as the paired statistical unit. Unless stated otherwise, confidence intervals are pointwise and unadjusted for multiple comparisons. Full dataset splits, probe architectures, secondary metrics, and implementation details are reported in the Appendix.

4.2 Utterance-Level Exact-Budget Allocation

Table 1 shows improvements at all four rates in all three domains: 3.34–4.39% for speech, 1.16–1.77% for music, and 1.07–1.61% for environmental audio. Every dynamic condition uses no more serialized bits than its matched fixed-depth reference. On speech, the three-domain controller captures 95.4–97.1% of the latent-utility oracle improvement under this offline allocation protocol. This gain-capture statistic is computed from mean distortions and should not be conflated with the utterance-level causal statistic below.

4.3 Causal Online Allocation and Streaming

Under causal commitment, UniAdapt improves Log-STFT at $D = 2, 4, 6$, while $D = 3$ incurs a 0.55% increase. All four operating points use fewer serialized bits, with zero prefix-budget violations across 800 utterance–rate evaluations.

At $D = 4$, learned online allocation without dwell improves Log-STFT by 1.21%, compared with 4.67% for the offline latent oracle; the mean utterance-level oracle-gain capture is 21.05%. This differs from the 95.4–97.1% offline statistic both in allocation constraints and aggregation: the offline value is derived from mean distortions, whereas the online value averages utterance-level gain ratios.

Streaming. On an NVIDIA A100, with the sequential allocator running on CPU, we evaluate 50 utterances at four rates and 100-, 200-, and 500-ms chunks. Chunked and full causal execution produce identical transmitted indices, expert maps, and depth maps; decoded waveforms agree within numerical tolerance. Mean waveform-path RTF is 0.491–0.655, so all evaluated configurations run faster than real time. The timed path includes online exact-budget accounting but excludes final byte packing and container parsing. For 500-ms chunks, estimated startup latency—500-ms expert buffering plus measured first-output computation—is 735.9–745.2 ms, and side information occupies 1.89–6.25% of the serialized container.

Ablations. The online controls trade reconstruction quality, switching, and budget utilization. The "no smoothing" configuration jointly disables the transition penalty, hysteresis, and dwell controls;

Table 3: Three-domain MUSHRA results using reconstructions produced by utterance-level offline allocation. Confidence intervals use the listener as the paired statistical unit.

Domain	Fixed	Dynamic	Δ	95% CI
Speech	72.88	76.40	$+3.52$	$[+2.12, +4.92]$
Music	65.33	65.84	$+0.51$	$[-0.96, +1.97]$
Environment	66.06	64.85	-1.22	$[-2.71, +0.28]$
Overall	68.09	69.03	$+0.94$	$[-0.20, +2.07]$

Table 4: Semantic-aware allocation relative to the corresponding normalized acoustic-only reference at $\beta = 0$. Results use development subsets; oracle and learned evaluations use separately sampled 50-utterance subsets. Negative semantic change is better, whereas positive Log-STFT change denotes an acoustic cost.

Utility	Depth	Semantic Δ	Log-STFT Δ
Oracle	2	-9.38%	$+0.16\%$
Oracle	3	-10.25%	$+0.46\%$
Oracle	4	-16.71%	$+0.71\%$
Learned	2	-2.94%	$+0.47\%$
Learned	3	-5.62%	$+0.48\%$
Learned	4	-13.47%	$+0.61\%$

it is not a separate temporal filtering operation. Switching then increases from 9.10 to 37.05 transitions/s. Removing dwell improves the $D = 4$ Log-STFT point estimate from -0.91% to -1.21%, showing that dwell primarily controls trajectory stability rather than directly improving reconstruction.

With dwell reduced to one frame, learned acoustic utility improves Log-STFT by 1.21%, while latent oracle utility improves it by 1.39%. Three random permutations of the flattened frame–layer utility array instead degrade Log-STFT by 1.75–1.80%, while uniform utility recovers fixed-depth behavior. These results support the importance of assigning predicted utilities to their corresponding frame–layer refinements rather than introducing depth variation alone.

We further compare exact-prefix enforcement with an approximate-side-cost configuration that disables the exact guard, restores a 3-bit/frame depth charge, and uses dwell $= 1$. This alternative produces no final serialized budget violations in the evaluated samples but underuses the matched budget by 6.50–12.81%. The per-prefix hard guarantee follows from the feasibility constraint in Sec. 3.3; this finite-sample comparison instead supports improved budget utilization with exact accounting. Complete ablation results are provided in the Appendix.

4.4 Subjective and Objective-Dependent Evaluation

We conduct a MUSHRA-style study following ITU-R BS.1534 (International Telecommunication Union, 2015) using reconstructions produced by utterance-level offline allocation rather than the causal streaming allocator. Twenty listeners each complete 30 trials (10 speech, 10 music, and 10 environmental), yielding 2,400 ratings; all satisfy the predefined hidden-reference quality criterion. Each trial contains a hidden reference, fixed $D = 4$, dynamic $D = 4$, and a 3.5-kHz low-pass anchor. Speech improves by 3.52 points with a confidence interval excluding zero. Music, environmental audio, and overall comparisons have intervals containing zero.

On 20 LibriSpeech dev-clean utterances, signed HuBERT refinement gains used for this diagnostic have an overall frame–layer correlation of 0.4205 with the non-negative acoustic marginal gains. At $D = 4$, learned semantic-aware allocation reduces HuBERT distortion by 13.47% at a 0.61% Log-STFT cost. The semantic predictor has pooled frame–layer Pearson correlation 0.733; this should not be interpreted as a per-layer correlation. These experiments are speech-only and utterance-level; causal streaming uses acoustic utility only.

Table 5: Representative frozen-token results at $D = 4$. Values are five-seed means. Dynamic denotes acoustic utterance-level allocation; MTG uses the artist-disjoint Top-50 downstream protocol. All rates include depth and expert metadata.

Task	Metric	Fixed	Dynamic	Rate Δ
CTC	CER $\downarrow$	0.5993	0.5924	-0.57%
Speaker	EER $\downarrow$	0.1782	0.1616	-0.57%
FSD50K	macro-mAP $\uparrow$	0.0681	0.0697	-1.38%
MTG	macro-mAP $\uparrow$	0.1006	0.1007	-0.26%
Clotho	mean recall $\uparrow$	0.0230	0.0239	-0.60%

4.5 Scalable Frozen-Token Representations

The shared cross-rate CTC probe is trained on dynamic multi-rate tokens and yields lower CER at all four rates, with 5/5 seed wins and paired intervals excluding zero. Because its greedy WER remains near one, we treat CER as a lightweight cross-rate representation diagnostic rather than a replacement for external ASR.

The speaker probe is likewise trained on dynamic multi-rate tokens. Dynamic tokens significantly reduce EER at $D = 3, 4, 6$, whereas $D = 2$ is not statistically conclusive; the improvement does not extend uniformly to the secondary minDCF metric. For FSD50K, macro-mAP improves in all five seeds by $+0.001633$, with 95% CI $[+0.001204, +0.002062]$, while using 1.381% fewer serialized bits.

For artist-disjoint MTG-Jamendo and Clotho (Drossos et al., 2020), the primary metric point estimates are positive at lower realized bitrates, but their paired intervals include zero. We therefore report no statistically significant difference on these metrics; the comparisons do not establish equivalence or non-inferiority. For MTG, artist-disjoint refers specifically to the downstream probe split.

Frozen external ASR. At $D = 4$, fixed, acoustic-dynamic, and semantic-joint reconstructions obtain WERs of 2.762%, 2.630%, and 2.594%, respectively, using frozen wav2vec2 (Baevski et al., 2020). The joint condition gives a 6.11% relative WER reduction in the point estimate, but the 10,000-resample speaker-cluster bootstrap interval reaches zero. We therefore report this as a strong trend rather than a statistically conclusive recognition improvement.

4.6 External Codec Positioning

On all 2,620 LibriSpeech test-clean utterances, the official 24-kHz EnCodec checkpoint (Défossez et al., 2023) obtains PESQ/STOI 2.747/0.938 at 6.113 kbps, compared with 1.913/0.871 at 6.076 kbps for UniAdapt fixed and 2.043/0.889 at 6.043 kbps for UniAdapt dynamic. Dynamic allocation significantly improves the same frozen UniAdapt backbone at a slightly lower rate, while EnCodec retains higher absolute reconstruction quality. Because the systems differ in architecture, sample rate, training objective, and quantization design, this comparison is contextual rather than architecture-controlled.

5 Conclusion

We introduced UniAdapt, which treats audio tokens as a budgeted representation resource by learning frame- and layer-wise marginal utilities and allocating RVQ capacity causally under the true serialized-bit budget. The same frozen hierarchy supports multiple rates and acoustic or semantic objectives without retraining the codec.

Across speech, music, and environmental audio, UniAdapt improves utterance-level reconstruction under no-larger serialized budgets. In the causal setting, it improves three of four speech rates with zero budget violations and runs faster than real time. Human evaluation shows a significant speech-quality improvement, while frozen-token experiments show selective downstream gains and support cross-rate use.

AI USE STATEMENT

We used generative AI tools to assist with code debugging, experiment bookkeeping, figure preparation, and language drafting and editing. The authors designed the research questions, methods, experimental protocols, and statistical analyses. The authors manually reviewed all AI-assisted text, verified reported numerical results against the corresponding experiment files, tested AI-assisted code for correctness, and checked citations against the original sources. Generative AI tools did not generate experimental measurements or listener ratings. The authors take responsibility for the final content of the paper and the reported results.

ETHICS STATEMENT

This work uses publicly available speech, music, and environmental audio datasets under their respective licenses. The subjective evaluation involved human listeners who were informed about the listening task and participated voluntarily. No personally sensitive information was collected beyond the responses required for the listening test.

Neural audio coding can support efficient communication and audio representation, but reconstructed speech and speaker representations may also raise privacy concerns. We use speaker verification only as a representation benchmark and do not perform demographic or sensitive-attribute inference.

REPRODUCIBILITY STATEMENT

The paper and Appendix specify the frozen codec and reversible operating mode, utility targets and controller architecture, training procedure, exact serialized-bit accounting, online allocation configuration, operating rates, streaming protocol, evaluation settings, random seeds, and statistical procedures. The Appendix additionally reports implementation details, correctness and equivalence checks, controller training configurations, and the checkpoints used for the reported experiments.

REFERENCES

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*, volume 33, 2020.

Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The MTG-Jamendo dataset for automatic music tagging. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning*, 2019.

Yunkee Chae, Woosung Choi, Yuhta Takida, Junghyun Koo, Yukara Ikemiya, Zhi Zhong, Kin Wai Cheuk, Marco A. Martínez-Ramírez, Kyogu Lee, Wei-Hsiang Liao, and Yuki Mitsufuji. Variable bitrate residual vector quantization for audio coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10889508.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=ivCd8z8zR2.

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.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 736–740, 2020. doi: 10.1109/ICASSP40776.2020.9052990.

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. FSD50K: An open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:829–852, 2022. doi: 10.1109/TASLP.2021.3133208.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman 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. doi: 10.1109/TASLP.2021.3122291.

Zhichao Huang, Chutong Meng, and Tom Ko. RepCodec: A speech representation codec for speech tokenization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 5777–5790, 2024. doi: 10.18653/v1/2024.acl-long.314.

International Telecommunication Union. Method for the subjective assessment of intermediate quality level of audio systems. Recommendation BS.1534-3, ITU-R, October 2015.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, Ziang Zhang, Xiaoda Yang, Rongjie Huang, Yidi Jiang, Qian Chen, Siqi Zheng, and Zhou Zhao. WavTokenizer: An efficient acoustic discrete codec tokenizer for audio language modeling. In *International Conference on Learning Representations*, 2025.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved RVQGAN. In *Advances in Neural Information Processing Systems*, volume 36, pp. 27980–27993, 2023.

Jiaqi Li, Xiaolong Lin, Zhekai Li, Shixi Huang, Yuancheng Wang, Chaoren Wang, Zhenpeng Zhan, and Zhizheng Wu. DualCodec: A low-frame-rate, semantically-enhanced neural audio codec for speech generation. In *Interspeech*, pp. 4883–4887, 2025. doi: 10.21437/Interspeech.2025-468.

Jiaqi Li, Yao Qian, Yuxuan Hu, Leying Zhang, Xiaofei Wang, Heng Lu, Manthan Thakker, Jinyu Li, Sheng Zhao, and Zhizheng Wu. FlexiCodec: A dynamic neural audio codec for low frame rates. In *International Conference on Learning Representations*, 2026.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. MUSDB18-HQ: An uncompressed version of MUSDB18, 2019.

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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pp. 749–752, 2001. doi: 10.1109/ICASSP.2001.941023.

Hubert Siuzdak, Florian Grötschla, and Luca A. Lanzendörfer. SNAC: Multi-scale neural audio codec. *arXiv preprint arXiv:2410.14411*, 2024.

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. doi: 10.1109/TASL.2011.2114881.

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, 2022.

Hanglei Zhang, Yiwei Guo, Zhihan Li, Xiang Hao, Xie Chen, and Kai Yu. Unlocking temporal flexibility: Neural speech codec with variable frame rate. In *Interspeech*, pp. 5003–5007, 2025. doi: 10.21437/Interspeech.2025-1289.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. SpeechTokenizer: Unified speech tokenizer for speech language models. In *International Conference on Learning Representations*, 2024.

Mingyu Zhao and Zhiyong Wu. UniStream: Multi-expert residual vector quantization for 48 kHz causal streaming audio coding. *arXiv preprint arXiv:2609.09866*, 2026.

Mingyu Zhao, Zijian Lin, Yutang Feng, Jiatao Chen, Fan Wang, Jiehui Luo, Yuhao Ding, Jinchao Zhang, and Zhiyong Wu. BAMU: Bitstream-aware marginal-utility allocation for frozen pre-trained neural speech codecs. *arXiv preprint arXiv:2608.08432*, 2026a.

Mingyu Zhao, Zijian Lin, Kun Wei, and Zhiyong Wu. SPG-Codec: Exploring the role and boundaries of semantic priors in ultra-low-bitrate neural speech coding. In *IEEE International Conference on Multimedia and Expo*, 2026b. arXiv:2604.26296.

A Implementation and Reproducibility Details

This section provides implementation details omitted from the main paper. We distinguish between utterance-level acoustic or semantic-aware allocation and causal acoustic depth allocation. Joint utility uses complete-utterance normalization, whereas the causal acoustic controller uses raw predicted utilities. Causality of the online allocation policy is conditional on the supplied budget reference and target rate; shared-expert quantization additionally requires block buffering.

A.1 Frozen Codec and Reversible Operating Mode

All experiments use a frozen UniStream backbone (Zhao & Wu, 2026). The codec operates at 48 kHz and produces 150 latent frames per second. Its quantizer contains $K = 8$ RVQ layers: one shared first layer and seven expert-specific acoustic layers. Each codebook contains 1024 entries, so each active RVQ index requires 10 bits. The encoder, quantizer input projection, codebooks, routing modules, and decoder remain frozen.

The encoder produces $z_t \in \mathbb{R}^C$, and the quantizer input projection initializes the residual:

$$v_t = P_{\text{in}}(z_t) \in \mathbb{R}^{C_q}, \qquad r_{t,0} = v_t, \qquad r_{t,k} = r_{t,k-1} - q_{t,k}. \tag{1}$$

The controller consumes encoder latents z_t, while acoustic utility is measured in the projected quantization space.

For expert-specific layers, we derive frame-level Top-1 assignments from the first acoustic router and take their majority within each block of 75 latent frames (500 ms). The selected expert is shared across the acoustic RVQ layers in that block. Expert assignments are explicitly serialized and are not decision variables of the depth allocator. The backbone has four acoustic experts; the utility controller uses a single depth-action expert dimension because it predicts depth utility under this externally determined expert map.

Serialized experiments use the reversible Top-1/shared-expert mode. The legacy Top-2 forward path mixes expert outputs using continuous routing weights, but its stored discrete representation does not contain sufficient information to reproduce that mixture exactly. We therefore do not use that path for UniAdapt bitstreams.

The main fixed reference depths are

$$D \in \{2, 3, 4, 6\},$$

with realized fixed-container rates of approximately 3.1, 4.6, 6.1, and 9.1 kbps. Dynamic conditions are compared against the corresponding serialized fixed-depth reference, including its side information, rather than against the nominal $150 \times D \times 10$ index rate.

A.2 Utility Targets and Controller

Acoustic targets. For frame t and layer k, the target is the non-negative reduction in channel-normalized residual energy:

$$\Delta^{\text{ac}}_{t,k} = \left[\frac{\|r_{t,k-1}\|_2^2}{C_q} - \frac{\|r_{t,k}\|_2^2}{C_q} \right]_+, \qquad k = 1, \ldots, K. \tag{2}$$

The implementation predicts all K layers. Because layer 1 is mandatory, its contribution is identical across candidate depths and can be omitted from the allocation objective without changing depth selection.

Semantic targets. For speech, we use the frozen `facebook/hubert-base-ls960` teacher (Hsu et al., 2021), taking its layer-9 hidden states. Waveforms are resampled from 48 to 16 kHz before teacher inference. Let $\hat{x}^{(k)}$ denote reconstruction at uniform RVQ depth k. Original and reconstructed waveforms are restricted to their common sample length, and teacher outputs are restricted to their common frame length J.

On the HuBERT time grid, semantic distortion is cosine distance:

$$D^{\text{sem}}_{j,k} = 1 - \cos\left(\Phi_j(x), \Phi_j(\hat{x}^{(k)})\right), \qquad j = 1, \ldots, J. \tag{3}$$

We first difference adjacent-depth distortions and clip negative gains, then interpolate to the codec grid:

$$\Delta^{\mathrm{sem}}_{t,k} = \mathcal{I}_{J\to T}\left(\left[D^{\mathrm{sem}}_{\cdot,k-1} - D^{\mathrm{sem}}_{\cdot,k}\right]_+\right)_t, \quad k = 2,\dots,K, \qquad \Delta^{\mathrm{sem}}_{t,1} = 0. \tag{4}$$

Interpolation uses `mode="linear"` and `align_corners=False`. This order of differencing, clipping, and interpolation matches label generation. HuBERT is used offline for semantic targets and semantic-distortion evaluation; it is absent from token extraction and deployed inference.

Prediction and training loss. The controller directly predicts non-negative marginal utilities:

$$h_t = f_\theta(z_{\le t}), \qquad \hat{\Delta}^{o}_{t,k} = \mathrm{softplus}(g_o(h_t)_k), \quad o \in \{\mathrm{ac}, \mathrm{sem}\}. \tag{5}$$

Targets are used in their original utility domain. No logarithmic target transformation or inverse transformation is applied. Training uses masked Smooth-L1 regression:

$$\mathcal{L}_o = \frac{1}{|\Omega|}\sum_{(t,k)\in\Omega} \mathrm{SmoothL1}\left(\hat{\Delta}^{o}_{t,k}, \Delta^{o}_{t,k}\right), \tag{6}$$

where Ω contains non-padded positions and all K layer outputs.

Causal architecture. The shared trunk contains three left-padded temporal convolutions with kernel sizes 5, 3, and 3 and channel dimensions $256 \to 128 \to 128 \to 64$. Each convolution is followed by channel LayerNorm applied independently at each frame and a GELU activation. Separate 1×1 convolutional heads predict acoustic and semantic utilities, followed by softplus.

The three convolutions require 4, 2, and 2 previous latent frames, respectively. Exact chunked controller inference therefore retains eight raw latent frames, equivalent to 53.33 ms at 150 Hz. The target bitrate is never supplied to the predictor. A speech-probability head is present in the implementation, but its loss weight is zero in the reported configurations and its output is not used for acoustic or joint allocation.

Optimization. Controllers use AdamW with learning rate 3×10^{-4} and gradient norm clipping at 1.0. The speech acoustic controller uses 5,000 LibriSpeech train-clean-100 utterances. The balanced three-domain controller uses 1,000 examples each from LibriSpeech, MTG-Jamendo, and the official FSD50K training split; its training records are randomly cropped to at most 450 latent frames.

For semantic-head training, we initialize from the speech acoustic checkpoint and freeze the shared trunk, acoustic head, and speech-probability head. Only the semantic head is optimized, using 1,000 speech examples and a semantic regression weight of 20. This training weight is distinct from the allocation weight $\beta = 0.25$.

A.3 Exact Joint-Utility Normalization

Semantic-aware joint experiments normalize acoustic and semantic utilities separately before combining them. Let $U^{(m)}_{t,\ell}$ denote a predicted or oracle marginal utility for objective $m \in \{\mathrm{ac}, \mathrm{sem}\}$. For each utterance and branch, we compute one scalar over all time frames and optional layers:

$$s_m = \max\left(\frac{1}{T(K-1)}\sum_{t=1}^{T}\sum_{\ell=2}^{K} U^{(m)}_{t,\ell}, 10^{-8}\right). \tag{7}$$

Normalization is

$$\widetilde{U}^{(m)}_{t,\ell} = \begin{cases} 0, & \ell = 1, \\ U^{(m)}_{t,\ell}/s_m, & \ell = 2,\dots,K. \end{cases} \tag{8}$$

The two branches use distinct denominators s_{ac} and s_{sem}. Statistics are computed over all optional layers, not only those selected by the allocation.

The combined utility is

$$U^{\mathrm{joint}}_{t,\ell} = \widetilde{U}^{(\mathrm{ac})}_{t,\ell} + \beta\,\widetilde{U}^{(\mathrm{sem})}_{t,\ell}, \qquad \beta = 0.25. \tag{9}$$

Table 6: Initial dual prices used for the four online operating points.

Target depth	**2**	**3**	**4**	**6**
λ_{init}	0.20	0.10	0.08	0.02

There is no subsequent renormalization or division by $1+\beta$. When neither denominator is clamped, each normalized optional-layer branch has mean one and the joint mean is 1.25. Setting the first-layer utility to zero does not remove its code: layer 1 remains mandatory.

Causality and baseline distinction. Complete-utterance normalization is used only for offline joint token extraction and semantic oracle/learned-allocation experiments. In semantic-weight sweeps, $\beta = 0$ denotes the normalized acoustic reference. The main acoustic-only token experiments and causal streaming experiments instead consume raw predicted acoustic utilities. They neither compute utterance-global utility normalization nor combine semantic utility. No speech-probability or bitrate gate is applied in the reported joint experiments.

A.4 Online Allocator Configuration

The causal experiments use the speech-specific acoustic controller. The formal online configuration was selected on development data and fixed for test evaluation:

$$\begin{aligned} \eta &= 0.002, & \rho &= 1.0, \\ h &= 0.20, & m_{\text{dwell}} &= 5, \\ T_{\text{bucket}} &= 0.5\ \text{s}, & b_{\text{init}} &= 0. \end{aligned} \tag{10}$$

Here η is the dual step size, ρ the transition penalty, h the hysteresis margin, and m_{dwell} the dwell-counter threshold in latent frames.

Let R_{target} be the supplied target rate in kbps and $f_{\text{frame}} = 150$ Hz. In evaluation, R_{target} is the actual container bitrate of the matched fixed-depth reference for that utterance. The rate-control quantities are

$$c = \frac{1000 R_{\text{target}}}{f_{\text{frame}}}, \qquad c_t(d) = 10d, \qquad b_{\max} = 1000 R_{\text{target}} T_{\text{bucket}}. \tag{11}$$

The code-index cost $c_t(d)$ is used as a surrogate for total serialized cost. Additional estimated depth-side and expert-side charges are zero in the final online configuration; actual side-map costs remain included in the serialized-budget comparison.

For the utility $U_t(d)$ defined in the main text, the candidate score is

$$S_t(d) = U_t(d) - \lambda_t c_t(d) - \rho\, \mathbf{1}[d \neq d_{t-1}]. \tag{12}$$

Bucket affordability requires $c_t(d) \leq b_t + c$, with minimum depth $d = 1$ exempt from this bucket mask. Exact feasibility is enforced separately and is never bypassed.

The dwell counter is initialized to m_{dwell}, reset to zero after a depth change, and incremented when depth is retained. While the counter is below its threshold, the previous depth is retained if it remains exactly feasible. Otherwise the dwell lock is released. After applying admissibility masks, hysteresis retains the previous depth when it remains admissible and the score improvement is below h.

After choosing d_t, rate-control state is updated by

$$\begin{aligned} b_{t+1} &= \text{clip}\left(b_t + c - c_t(d_t),\, 0,\, b_{\max}\right), \\ \lambda_{t+1} &= \left[\lambda_t + \eta \frac{c_t(d_t) - c}{\max(c, 1)}\right]_+ . \end{aligned} \tag{13}$$

Initializing the bucket empty avoids granting initial rate credit. The exact prefix guard, rather than the bucket alone, provides the hard serialized-budget guarantee.

A.5 Exact Serialized-Bit Accounting

The container consists of a fixed header, a depth RLE payload, an expert RLE payload, and packed active indices. The header occupies $H = 256$ bits. Each variable payload is independently byte aligned.

For 1024-entry codebooks,

$$R_{\mathrm{code}}(d_{1:t}) = 10 \sum_{i=1}^{t} d_i. \tag{14}$$

A map run stores its value followed by an Elias-gamma-coded positive run length. For the eight-layer depth map, each depth value occupies four bits; for the four-expert map, each expert ID occupies two bits. Writing $\mathcal{R}_d$ and $\mathcal{R}_e$ for their runs,

$$\begin{aligned} R^{\mathrm{RLE}}_{\mathrm{depth}} &= \sum_{r \in \mathcal{R}_d} (4 + 2\lfloor \log_2 \ell_r \rfloor + 1), \\ R^{\mathrm{RLE}}_{\mathrm{expert}} &= \sum_{r \in \mathcal{R}_e} (2 + 2\lfloor \log_2 \ell_r \rfloor + 1). \end{aligned} \tag{15}$$

Consecutive expert blocks selecting the same expert form one run.

The exact size obtained by serializing a sequence prefix is

$$\begin{aligned} R_{\mathrm{actual}}(d_{1:t}, e_{1:t}) = H &+ 8 \lceil R_{\mathrm{code}}/8 \rceil \\ &+ 8 \left\lceil R^{\mathrm{RLE}}_{\mathrm{depth}}/8 \right\rceil \\ &+ 8 \left\lceil R^{\mathrm{RLE}}_{\mathrm{expert}}/8 \right\rceil. \end{aligned} \tag{16}$$

This is a sequence-prefix size definition, not a claim that each prefix is separately packetized and transmitted.

Matched fixed and dynamic prefixes use the same header format and expert map. Their header and expert-map costs are therefore identical. Define the exact code-plus-depth payload

$$\tilde{R}(d_{1:t}) = 8 \lceil R_{\mathrm{code}}(d_{1:t})/8 \rceil + 8 \left\lceil R^{\mathrm{RLE}}_{\mathrm{depth}}(d_{1:t})/8 \right\rceil, \tag{17}$$

and let $\tilde{B}_t = \tilde{R}(D, \ldots, D)$ for a length-t fixed-depth reference. The guard admits

$$\mathcal{F}_t = \left\{ d \in \{1, \ldots, K\} : \begin{array}{c} \tilde{R}(d_{1:t-1}, d) \le \tilde{B}_t, \\ \tilde{R}(d_{1:t-1}, d, 1) \le \tilde{B}_{t+1} \end{array} \right\}. \tag{18}$$

The second condition checks feasibility of a hypothetical next frame at minimum depth. Because common side costs appear on both sides, the guard does not need the next expert assignment to perform this comparison. All header and expert-map bits nevertheless remain included in the reported total bitrate.

The guard maintains incremental serializer-consistent state: the active-code count, completed depth-run bits, current depth, and current run length. It does not materialize every candidate bitstream at every frame. The implementation was additionally stress-tested using 200 random allocation trajectories per target depth $D \in \{2, 3, 4, 6\}$, each extending to 5,000 frames. No state without a feasible action was observed in these checks.

A.6 Streaming State and Chunked Inference

Encoder latents feed the cached causal utility controller and online allocator. Buffered latents also undergo full shared-expert RVQ evaluation. The committed depths select a prefix mask, and the selected latent vectors are passed to the cached waveform decoder. With fixed expert assignments, computational prefix early exit is an equivalent execution option; reported real-backbone runtime experiments use full RVQ evaluation followed by masking.

The encoder has receptive field 2254 waveform samples and stride 320. Its cached implementation retains 2560 stride-aligned waveform samples (53.33 ms).

Table 7: Input chunk sizes and corresponding latent-frame counts used for streaming evaluation.

Input chunk	**100 ms**	**200 ms**	**500 ms**
Latent frames	15	30	75

Table 8: Utility-controller training configurations. Speech acoustic and semantic labels are generated from excerpts of at most three seconds; the three-domain training loader randomly crops records to at most 450 latent frames. Semantic training initializes from the speech acoustic checkpoint and updates only the semantic head.

Controller	**Examples**	**Batch size**	**Steps**
Speech acoustic	5,000	32	10,000
Three-domain acoustic	3,000	16	10,000
Speech semantic head	1,000	32	5,000

Non-final input chunks are stride aligned. The controller retains eight raw latent frames. The allocator carries its dual price, bucket state, previous depth, and dwell counter across chunks, together with the incremental exact-budget state. Its generic action-expert field is not an optimization of the backbone expert map. The decoder retains eleven latent frames of context (73.33 ms). Shared-expert processing buffers 75 frames and flushes any remaining partial block at the end of the utterance.

The final runtime configuration places the sequential allocator on CPU and the neural codec and controller on the GPU, avoiding repeated per-frame GPU synchronization. Chunked execution is compared against full causal execution under the same configuration.

Reported streaming timings include encoding, utility prediction, online allocation and exact-budget accounting, full RVQ evaluation, prefix masking, and waveform decoding. Final byte packing and container parsing are excluded. Estimated startup latency is defined as the 500-ms expert-block buffer plus measured first-output computation.

A.7 Correctness and Equivalence Checks

We use explicit checks for chunked computation, depth feasibility, and dynamic serialization.

For the cached encoder, five test utterances evaluated with 100, 200, and 500 ms chunks produce exactly the same latent values as full encoding in the tested configuration, with zero maximum absolute difference. Chunked utility prediction agrees with full causal prediction within numerical tolerance, and chunked allocation produces the same depth maps as processing the causal sequence in one call.

Decoder checks cover five utterances, depths 2, 4, and 6, and latent chunk lengths 15, 30, and 75. The reported maximum waveform absolute difference is approximately 2.94×10^{-6}, with relative ℓ_2 difference approximately 5.49×10^{-7}.

End-to-end checks compare active code indices, block-shared expert maps, depth maps, serialized-bit counts, and decoded waveforms. Discrete quantities match exactly; waveforms agree within numerical tolerance. The formal streaming evaluation covers 50 utterances, four rates, and three chunk sizes, giving 600 conditions with no observed budget violations. These empirical checks complement the serializer-consistent feasibility rule; they are not a proof based solely on finite samples.

A.8 Training Sets, Checkpoints, and Random Seeds

The principal controller training configurations are summarized in Table 8.

Speech acoustic label sampling uses seed 2030, semantic label sampling uses seed 2038, and balanced three-domain subset sampling uses seed 2040. These are data-sampling seeds, distinct from the downstream probe training seeds 2040–2044.

Table 9: Online component ablations at $D = 4$ on the same 200 test utterances. Changes are relative to matched fixed-depth coding and use Eq. (19). All configurations in this table retain the exact prefix guard.

Configuration	Rate Δ	Log-STFT Δ	Switches/s	Violations
Full, $m = 5$	-2.860%	-0.906%	9.10	$0/200$
No smoothing	-0.411%	-0.674%	37.05	$0/200$
No transition penalty	-1.286%	-0.445%	18.93	$0/200$
No hysteresis	-2.653%	-0.843%	10.84	$0/200$
No dwell, $m = 1$	-3.060%	-1.206%	9.70	$0/200$
Penalty only	-2.665%	-1.212%	11.43	$0/200$
Hysteresis only	-1.164%	-1.142%	20.12	$0/200$
Dwell only	-0.987%	-0.231%	23.66	$0/200$
No dual update	-17.213%	-0.060%	7.16	$0/200$
Redundant side cost	-9.526%	$+1.085\%$	9.69	$0/200$

The formal online experiments use the speech acoustic controller at step 10,000. Semantic-aware experiments use the semantic-head checkpoint at step 5,000, and three-domain reconstruction uses the balanced acoustic controller at step 10,000. All conditions share the frozen UniStream codec checkpoint at step 200,000.

B Additional Allocation and Ablation Results

This section provides additional results underlying Sec. 4.3. Unless otherwise stated, experiments use the same 200 LibriSpeech test-clean utterances, target depth $D = 4$, the frozen speech-specific 5K acoustic controller, and 500-ms input chunks. Codec parameters and expert assignments remain fixed across allocation conditions.

For condition a, let L_i^a and R_i^a denote the Log-STFT distortion and serialized bitrate of utterance i. Percentage changes in this section are averages of per-utterance relative changes:

$$\delta_L^a = \frac{100}{N} \sum_{i=1}^{N} \left(\frac{L_i^a}{L_i^{\text{fixed}}} - 1 \right), \qquad \delta_R^a = \frac{100}{N} \sum_{i=1}^{N} \left(\frac{R_i^a}{R_i^{\text{fixed}}} - 1 \right). \tag{19}$$

Negative values indicate lower distortion or bitrate. Switching rates are averaged over utterances. The violation column counts utterances whose final serialized size exceeds the matched fixed-depth budget. Conditions using the exact guard additionally enforce the per-prefix constraint defined in Sec. 3.3; zero final violations alone do not establish prefix feasibility for unguarded conditions.

B.1 Online Allocator Component Ablations

The *Full* configuration is the locked online configuration used in the main experiments: transition penalty $\rho = 1.0$, hysteresis margin $h = 0.20$, dwell-counter threshold $m = 5$, dual step $\eta = 0.002$, bucket duration 0.5 s, an initially empty bucket, and the exact prefix-feasibility guard. At $D = 4$, the initial dual price is 0.08.

The "no smoothing" condition jointly sets $\rho = 0$, $h = 0$, and $m = 1$; it does not remove a separate temporal filter. "No dwell" denotes the experimental setting $m = 1$ throughout this section. "Penalty only," "hysteresis only," and "dwell only" retain only the named switching control, while leaving dual rate control and exact-budget enforcement active. "No dual update" sets $\eta = 0$ while retaining the initial dual price. "Redundant side cost" adds a 3-bit/frame estimated depth charge while retaining the exact guard.

Jointly disabling the switching controls raises transitions from 9.10 to 37.05/s. Removing only the transition penalty raises switching to 18.93/s and reduces the reconstruction improvement. These

Table 10: Development-set comparison at $D = 4$. Both conditions use the same controller, exact guard, transition penalty, and hysteresis; only the dwell threshold changes. Percentages average per-utterance changes.

Configuration	Rate Δ	Log-STFT Δ	Switches/s	Violations
Full, $m = 5$	-2.7554%	-1.0680%	9.562	$0/500$
No dwell, $m = 1$	-2.5693%	-1.3806%	9.971	$0/500$

Table 11: Utility-information controls at $D = 4$ with the same online allocator, exact prefix guard, and dwell threshold $m = 1$. Shuffled controls permute flattened frame–layer entries. Uniform and oracle inputs use privileged full-utterance information; causality here describes the allocator's decision recurrence, not every utility-generation procedure.

Utility	Rate Δ	Log-STFT Δ	Switches/s	Violations
Learned acoustic	-3.0596%	-1.2062%	9.697	$0/200$
Shuffled, seed 2040	-0.0092%	$+1.7947\%$	26.282	$0/200$
Shuffled, seed 2041	-0.0062%	$+1.7467\%$	26.824	$0/200$
Shuffled, seed 2042	-0.0117%	$+1.7978\%$	26.851	$0/200$
Uniform	0.0000%	0.0000%	0.000	$0/200$
Oracle latent utility	-3.1503%	-1.3892%	12.556	$0/200$

results show that switching controls regulate the temporal complexity of the depth map and its associated side-information cost.

Dwell exhibits a quality–stability trade-off. Reducing its threshold from five to one improves the Log-STFT change from -0.906% to -1.206%, while increasing switching from 9.10 to 9.70/s. Penalty-only and hysteresis-only configurations also have better reconstruction point estimates than Full, but higher switching rates. Thus, the controls do not each contribute a monotonic quality gain.

Without dual updates, the allocator uses 17.21% less bitrate than the matched reference and recovers almost no reconstruction gain. This demonstrates the importance of adapting the dual price under the tested initialization and configuration. Adding an estimated depth-side charge on top of the exact guard likewise leads to substantial budget under-utilization and worse reconstruction. The final online configuration therefore uses zero estimated depth-side charge while retaining exact depth-RLE accounting.

Development-set dwell comparison. We additionally compare the two dwell settings on the full 500-utterance development set.

The development results show the same qualitative trade-off: $m = 1$ improves reconstruction but increases switching. The paired change in relative Log-STFT distortion is -0.3126 percentage points, while switching increases by 0.4089/s. We retain the locked $m = 5$ configuration for the main streaming results and report $m = 1$ separately.

B.2 Utility-Information Baselines

We next hold the $D = 4$ allocator fixed with $m = 1$ and vary its utility input. Learned utility is the acoustic controller output. For each shuffled control, the frame–layer utility array is flattened, randomly permuted, and reshaped. This preserves the multiset of utility values within each utterance but disrupts both temporal and layer correspondence. Uniform utility assigns the utterance-wide mean predicted utility to every frame–layer entry.

The oracle input uses ground-truth non-negative marginal reductions in projected latent residual energy. It is a privileged diagnostic requiring full RVQ computations, not a deployable predictor or a guaranteed optimum for waveform distortion.

All three shuffled controls degrade Log-STFT by approximately 1.75–1.80%, whereas learned utility improves it by 1.21%. This supports the importance of assigning utilities to their corresponding

Table 12: Approximate-side-cost allocation without the exact guard. The allocator uses a 3-bit/frame depth-side charge and $m = 1$. Violations count final serialized-budget overruns; their absence does not establish that every intermediate prefix is feasible.

Target	Rate Δ	Log-STFT Δ	Switches/s	Violations
$D = 2$	-12.81%	$+0.02\%$	16.36	$0/200$
$D = 3$	-9.03%	$+0.75\%$	19.60	$0/200$
$D = 4$	-8.80%	$+0.04\%$	13.67	$0/200$
$D = 6$	-6.50%	-0.12%	12.33	$0/200$

Table 13: Learned and oracle-utility allocation at $D = 4$. Percentages average per-utterance distortion changes relative to fixed depth. The online oracle row supplies privileged latent utilities to the online allocator; the offline row uses full-utterance allocation. The latent oracle is a proxy reference, not a certified optimum for waveform Log-STFT distortion.

Condition	Log-STFT Δ	Switches/s
Learned online, $m = 1$	-1.2062%	9.697
Oracle utility, online allocator	-1.3892%	12.556
Offline latent oracle	-4.6748%	5.688

frame–layer refinements rather than introducing arbitrary depth variation. Because shuffling also changes layer correspondence, this comparison does not isolate temporal alignment alone.

Uniform utility recovers fixed-depth behavior in this configuration. Replacing predicted utility with ground-truth latent utility improves the relative Log-STFT change by 0.1830 percentage points. The gain is modest under the same allocator, although the realized bitrates and switching rates also differ. This comparison measures the effect of replacing the utility signal within the tested allocation procedure, rather than an isolated causal decomposition of all sources of reconstruction error.

B.3 Exact versus Approximate Budget Enforcement

We compare exact-prefix enforcement with an approximate-side-cost configuration. The alternative disables the exact guard, restores a depth-side charge of 3 bits/frame, and uses $m = 1$. Initial dual prices remain 0.20, 0.10, 0.08, and 0.02 for $D = 2, 3, 4, 6$, respectively; the remaining rate-control parameters follow the online configuration.

This is a comparison between budget mechanisms, not an ablation that changes only guard availability. In particular, the score and bucket costs change when the approximate side charge is restored.

No final serialized-budget overrun occurs across these 800 utterance–rate evaluations. Nevertheless, the alternative uses 6.50–12.81% less bitrate than the matched reference and provides no positive reconstruction point-estimate gain at $D = 2, 3, 4$.

The hard per-prefix guarantee of the guarded method follows from its feasibility constraint: every committed action must satisfy the exact current-prefix budget and next-frame minimum-depth payload condition. The finite-sample comparison instead provides empirical evidence of better budget utilization under exact accounting than under the tested approximate-side-cost mechanism. It does not show that removing the guard alone necessarily causes final-budget violations.

B.4 Online versus Offline Oracle

We compare learned and ground-truth latent utilities within the same online allocator and then contrast them with full-utterance latent-oracle allocation. All rows use the same 200 test utterances and matched $D = 4$ fixed-depth budgets. The online conditions use $m = 1$ and exact-prefix enforcement. The offline reference uses full-sequence Lagrange–Viterbi allocation with `switch_bits`$= 50$ and an exact final-container constraint.

Replacing learned utility with ground-truth latent utility within the same online allocator changes the relative Log-STFT improvement from 1.2062% to 1.3892%. The offline latent oracle achieves 4.6748%, leaving a considerably larger difference between allocation procedures.

There are two distinct ways to express the learned-online versus offline-oracle difference. Relative to the common fixed-depth baseline, the reported percentage changes differ by

$$(-1.2062) - (-4.6748) = 3.4686 \quad \text{percentage points.}$$

Alternatively, averaging the distortion ratio relative to the offline oracle gives

$$G_{\mathrm{rel}} = \frac{100}{N} \sum_{i=1}^{N} \left(\frac{L_i^{\mathrm{online}}}{L_i^{\text{offline-oracle}}} - 1 \right) = 3.6639\%. \tag{20}$$

The latter is a relative percentage, not a percentage-point difference between the two changes from fixed depth.

The per-utterance oracle-gain capture is

$$G_{\mathrm{capture}} = \frac{100}{|\mathcal{J}|} \sum_{i \in \mathcal{J}} \frac{L_i^{\mathrm{fixed}} - L_i^{\mathrm{online}}}{L_i^{\mathrm{fixed}} - L_i^{\text{offline-oracle}}} = 21.05\%, \tag{21}$$

where $\mathcal{J} = \{i : L_i^{\mathrm{fixed}} > L_i^{\text{offline-oracle}}\}$. This is a mean of utterance-level ratios, rather than a ratio of mean improvements.

The main three-domain offline evaluation reports 95.4–97.1% speech oracle-gain capture computed from mean distortions. That result uses a different controller and evaluation protocol, as well as different allocation constraints and aggregation. It should not be interpreted as the gain capture of the causal streaming system.

These comparisons indicate that improving utility prediction alone, while retaining the tested online allocator, recovers only a small part of the observed difference to the offline reference. However, the online and offline procedures also differ in lookahead, prefix constraints, switching treatment, and optimization strategy. The experiments therefore do not isolate an unavoidable cost of causality or establish an upper bound on achievable online quality. They identify online allocation as an important remaining source of improvement alongside utility modeling.

C Streaming and Bitstream Analysis

This section reports the streaming measurements and correctness checks underlying Sec. 4.3. All experiments use the frozen 48-kHz codec and speech-specific acoustic utility controller. Encoder latents feed the causal controller and online allocator, while buffered latents undergo full shared-expert RVQ evaluation. The committed depths select a prefix mask, and the selected latent vectors are reconstructed by the cached decoder.

The timed waveform-to-waveform path includes encoding, utility prediction, online allocation and exact-budget accounting, full RVQ evaluation, prefix masking, and waveform decoding. It excludes model loading, audio-file loading and preprocessing, final byte packing, and container parsing. The reported runtime results therefore do not rely on computational prefix early exit or measure network transmission latency.

The input chunk sizes are 100, 200, and 500 ms, corresponding to 15, 30, and 75 latent frames at 150 Hz. Streaming states persist across chunk boundaries. Shared-expert quantization buffers 75 frames and applies one expert assignment across the acoustic RVQ layers of each block; any remaining partial block is processed at the end of the utterance.

C.1 Chunked–Full Equivalence

We compare chunked execution with full-sequence causal computation under the same controller, allocator configuration, and supplied budget reference. Full-sequence causal allocation processes the utility sequence in one call while retaining the same sequential decision rule; it is distinct from the offline Lagrange–Viterbi allocator.

Table 14: Waveform-streaming performance over 50 utterances. RTF entries are mean ± across-utterance standard deviation. Timing includes online exact-budget accounting, full RVQ evaluation, and prefix masking, but excludes final byte packing and parsing. Allocation statistics are averaged over utterances and are identical across the three chunk sizes for each target.

	Waveform-path RTF			Allocation Statistics		
Target	**100 ms**	**200 ms**	**500 ms**	**Rate Δ**	**Avg. depth**	**Switches/s**
$D = 2$	0.655 ± 0.040	0.575 ± 0.013	0.491 ± 0.015	-2.327%	1.878	12.45
$D = 3$	0.649 ± 0.013	0.574 ± 0.013	0.492 ± 0.015	-2.203%	2.843	16.40
$D = 4$	0.636 ± 0.014	0.543 ± 0.014	0.510 ± 0.015	-2.699%	3.831	9.21
$D = 6$	0.643 ± 0.014	0.543 ± 0.013	0.515 ± 0.015	-2.976%	5.760	9.19

The encoder has a receptive field of 2254 waveform samples and total stride 320. The cached implementation retains 2560 stride-aligned waveform samples (53.33 ms). Across five test utterances and all three chunk sizes, the chunked and full encoder outputs have identical shapes and zero measured latent difference.

The utility controller caches eight raw latent frames (53.33 ms). Its chunked outputs agree with full causal outputs within numerical tolerance. The allocator preserves its dual price, token-bucket state, previous depth, dwell counter, and incremental exact-budget state across chunks. Its generic action-expert field does not optimize the backbone expert map. Chunked and one-shot causal allocation produce identical framewise depth trajectories in the evaluated conditions.

The cached decoder retains 11 latent frames (73.33 ms). Checks cover five utterances, depths $D \in \{2, 4, 6\}$, and latent chunk lengths 15, 30, and 75, giving 45 conditions. The observed maximum waveform absolute difference is approximately 2.94×10^{-6}, and the maximum relative ℓ_2 difference is approximately 5.49×10^{-7}.

End-to-end comparisons additionally check:

- encoder latents and predicted acoustic utilities;
- quantizer layer vectors and RVQ code indices;
- block-shared expert maps and framewise depth maps;
- total serialized-bit counts; and
- reconstructed waveforms.

Code indices, expert maps, depth maps, and serialized sizes match exactly; continuous tensors agree within numerical tolerance.

An initial evaluation covers 20 utterances, four rates, and three chunk sizes, giving 240 conditions. The final evaluation covers 50 utterances under the same rate–chunk combinations, giving 600 conditions. All pass the corresponding equivalence and serialized-budget checks.

C.2 End-to-End Real-Time Performance

The final runtime configuration uses an NVIDIA A100 GPU for the neural codec and controller and CPU execution for the sequential allocator. Rate–chunk configurations are measured serially to avoid contention from concurrent allocator processes.

For utterance i, waveform-path real-time factor is

$$\mathrm{RTF}_i = \frac{t_i^{\mathrm{processing}}}{T_i^{\mathrm{audio}}}, \tag{22}$$

where $t_i^{\mathrm{processing}}$ is elapsed time for the complete chunk-processing loop and T_i^{audio} is input duration. The waveform is loaded before timing, and chunks are processed without waiting for real-time input arrival. GPU synchronization is used in the timing procedure.

All twelve rate–chunk configurations have mean RTF below one. The ranges are 0.636–0.655 for 100-ms chunks, 0.543–0.575 for 200-ms chunks, and 0.491–0.515 for 500-ms chunks. Larger

Table 15: Separate 500-ms latency-profile runs over 50 utterances. Startup is estimated using Eq. (23). The processing interval is defined by Eq. (24); its mean and standard deviation are computed across utterance-level means. All ± entries are standard deviations, not confidence intervals. Overhead fractions and rates are means across utterances.

Target	Est. startup	Processing interval	Overhead	Overhead kbps	RTF
$D = 2$	744.8 ± 6.4 ms	243.5 ± 3.6 ms	6.25%	0.187	0.514 ± 0.015
$D = 3$	744.1 ± 6.4 ms	244.4 ± 3.3 ms	4.71%	0.211	0.515 ± 0.015
$D = 4$	745.2 ± 7.0 ms	243.1 ± 3.5 ms	2.79%	0.165	0.513 ± 0.015
$D = 6$	735.9 ± 7.4 ms	234.9 ± 10.4 ms	1.89%	0.166	0.496 ± 0.025

chunks require fewer calls and less repeated processing of cached context per unit of audio, consistent with their lower measured RTF.

Rate changes are averages of per-utterance relative differences against the matched fixed-depth serialized containers. These statistics use the 50-utterance streaming subset and therefore differ from the 200-utterance online-allocation statistics in the main paper. Every evaluated streaming condition satisfies its matched serialized budget.

C.3 Estimated Startup Latency and Output Processing Intervals

We separately profile first-output computation and subsequent output completion intervals using 500-ms chunks. Let $c_{i,b}$ denote elapsed processing time from the start of the chunk loop until output block b is completed for utterance i. Estimated startup latency is

$$\hat{\tau}_i^{\text{startup}} = 500 \text{ ms} + c_{i,1}, \tag{23}$$

with $c_{i,1}$ expressed in milliseconds. This adds the required expert-block accumulation time to measured first-output computation. It is not a measurement of model-loading latency or a complete capture–network–playback system.

For an utterance with $M_i > 1$ output blocks, we also compute

$$\bar{\tau}_i^{\text{processing}} = \frac{1}{M_i - 1} \sum_{b=2}^{M_i} (c_{i,b} - c_{i,b-1}) . \tag{24}$$

This statistic measures intervals between output completions while processing preloaded audio without input pacing. It can include the final partial block and should not be interpreted as a network packet interval or the output cadence of a real-time audio source.

Estimated startup ranges from 735.9 to 745.2 ms, while mean output processing intervals range from 234.9 to 244.4 ms. These quantify different aspects of execution: startup includes expert-block accumulation, whereas the processing interval excludes waiting for input arrival. The implementation achieves faster-than-real-time processing in these measurements while retaining substantial initial buffering.

The RTF values in Table 15 come from separate profiling runs and need not equal the 500-ms RTF estimates in Table 14.

C.4 Serialized Side-Information and Padding Overhead

The dynamic container explicitly stores the depth trajectory and shared-expert map. Reported side-information overhead includes the header, byte-aligned depth and expert maps, and code-payload padding. For utterance i, let p_i^{code} be the number of padding bits added to the packed active indices. The measured overhead is

$$\begin{aligned} S_i = H &+ 8 \left\lceil R_{\text{depth},i}^{\text{RLE}}/8 \right\rceil \\ &+ 8 \left\lceil R_{\text{expert},i}^{\text{RLE}}/8 \right\rceil + p_i^{\text{code}}. \end{aligned} \tag{25}$$

The reported fraction and bitrate are

$$\frac{100 S_i}{R_{\text{actual},i}} \quad \text{and} \quad \frac{S_i}{1000 T_i^{\text{audio}}}, \tag{26}$$

Table 16: Representative $D = 4$, 500-ms profile with CPU allocation. These are measurements for one profiling utterance, not the 50-utterance means in Table 14. End-to-end RTF also includes processing-loop overhead not assigned to an individual component.

Component	RTF
Cached encoder	0.0379
Utility controller	0.0071
CPU allocator and exact guard	0.0698
Full shared-expert RVQ	0.2756
Cached decoder	0.0992
Other processing-loop overhead	0.0120
End-to-end waveform path	0.5017

respectively, averaged across utterances. Thus, these values represent all bits beyond the raw active-code indices, not only the two side maps.

For the 500-ms profiles, overhead accounts for 1.89–6.25% of the serialized container, corresponding to 0.165–0.211 kbps. Its relative fraction is largest at the lowest evaluated rate, where fewer code-index bits are available to amortize the overhead.

The 500-ms expert block trades signaling cost against buffering. Sharing assignments reduces expert-map variation, and adjacent blocks with the same assignment can be represented by a single RLE run. The corresponding expert block must nevertheless be available before its quantized output is finalized. All reported rates include these costs; no packet-transport overhead is measured.

C.5 Runtime Placement and Component Profile

The allocator makes a small sequential decision at each latent frame. An initial implementation ran this loop on GPU and incurred repeated host–device synchronization. In a representative $D = 4$, 500-ms profile, total RTF was approximately 4.47, with allocator RTF approximately 4.04.

The final implementation runs this control loop on CPU while keeping the encoder, utility controller, RVQ, and decoder on GPU. This preserves the allocation rule; the corresponding executions are checked against their full causal references.

CPU placement reduces total RTF in these representative profiles from approximately 4.47 to 0.502. Full RVQ evaluation is the largest measured component in the final profile. The difference between end-to-end time and the sum of instrumented stage times accounts for additional loop operations, including prefix masking and buffering. The component totals may differ slightly after rounding.

These measurements characterize the evaluated implementation and hardware. They do not claim that CPU execution is universally preferable for all allocators or that the reported runtime includes a separately measured serialization pipeline.

C.6 Streaming Budget Correctness

The chunked implementation uses the same exact prefix-feasibility guard as the one-shot causal allocator. Before a depth is committed, it must satisfy the matched fixed-depth prefix budget and the next-frame minimum-depth payload condition.

The guard incrementally tracks the exact byte-aligned code and depth-map payloads. Matched fixed and dynamic prefixes use identical header and expert-map costs, so the payload comparison enforces the complete serialized-budget inequality without requiring future expert assignments. Header, expert-map, and padding bits remain included when the final container size is computed.

All

$$50 \text{ utterances} \times 4 \text{ target rates} \times 3 \text{ chunk sizes} = 600$$

final streaming conditions satisfy the matched serialized budget. Their chunked and full causal executions also agree on code indices, expert assignments, depth maps, and serialized sizes.

Table 17: Optional-layer semantic gains on 20 development utterances. Gain means and positive-frame fractions use signed HuBERT-grid differences; Pearson correlations use aligned codec-grid entries.

Layer	Mean gain	Positive frames	Acoustic correlation
2	0.182084	93.9%	0.2875
3	0.077078	92.9%	0.1147
4	0.033587	87.8%	0.0164
5	0.015633	83.7%	−0.0200
6	0.008651	80.1%	−0.0102
7	0.006105	76.2%	−0.0231
8	0.004106	72.4%	−0.0364

The empirical checks support chunked–full equivalence and consistency between guard accounting and the serializer on the evaluated data. The per-prefix budget guarantee follows from restricting committed actions to the exactly feasible set, rather than from the absence of violations in a finite test set. Here a serialized prefix means the size obtained by serializing the corresponding sequence prefix; these experiments do not measure independently packetized prefix transmission.

D Semantic and Objective-Dependent Analysis

This section examines objective-dependent refinement value and evaluates semantic-aware allocation on speech. All joint-allocation results use the utterance-level normalization in Appendix A.3. The codec remains frozen, and causal streaming uses acoustic utility only.

D.1 Marginal Gains Across RVQ Layers

We analyze 20 LibriSpeech dev-clean utterances using frozen HuBERT features (Hsu et al., 2021). Mean cosine distortion decreases from 0.3755 at uniform depth 1 to 0.0483 at depth 8, with diminishing average gains from additional layers.

For this diagnostic, semantic gains retain their signs:

$$\delta^{\mathrm{sem}}_{j,k} = D^{\mathrm{sem}}_{j,k-1} - D^{\mathrm{sem}}_{j,k}. \tag{27}$$

Unlike the non-negative training targets, these diagnostic gains are not clipped. Their means and positive-frame fractions are computed on the HuBERT grid and averaged over utterances. For correlation analysis, signed gains are linearly interpolated to the codec grid and compared with the non-negative acoustic marginal gains.

Pooling optional layers and frames gives Pearson correlation 0.4205, whereas within-layer correlations approach zero in deeper layers. The pooled statistic also reflects between-layer variation and should not be interpreted as uniformly strong frame-level agreement. These observations motivate modeling semantic value separately from acoustic residual reduction; they do not establish statistical independence or semantic disentanglement.

D.2 Oracle and Learned Semantic Allocation

Oracle allocation uses measured acoustic and semantic marginal utilities, while learned allocation uses controller predictions. Both normalize each objective separately and compare joint allocation with its corresponding normalized acoustic reference at $\beta = 0$. All conditions satisfy the same matched fixed-depth budget ceiling; their realized serialized sizes need not be identical.

Oracle and learned evaluations use separately sampled 50-utterance dev-clean subsets, with sampling seeds 2034 and 2039, respectively. Percentage changes are computed from mean distortions within each evaluation. The two subsets are not paired with each other.

Increasing β shifts oracle allocation toward lower HuBERT distortion at greater acoustic cost. We use $\beta = 0.25$ for the reported learned joint condition as a moderate trade-off, without claiming

Table 18: Speech semantic–acoustic trade-offs under utterance-level allocation. Each row is relative to the corresponding $\beta = 0$ reference. Negative semantic change is better; positive Log-STFT change is an acoustic cost. Oracle and learned rows use separate development subsets and are not a paired predictor comparison.

Utility	β	Depth	Semantic Δ	Log-STFT Δ
Oracle	0.25	2	-9.38%	$+0.16\%$
Oracle	0.25	3	-10.25%	$+0.46\%$
Oracle	0.25	4	-16.71%	$+0.71\%$
Oracle	0.50	2	-11.49%	$+0.87\%$
Oracle	0.50	3	-14.34%	$+1.35\%$
Oracle	0.50	4	-23.65%	$+1.50\%$
Learned	0.25	2	-2.94%	$+0.47\%$
Learned	0.25	3	-5.62%	$+0.48\%$
Learned	0.25	4	-13.47%	$+0.61\%$

Table 19: Frozen external ASR on 500 test-clean utterances. CER and WER are corpus-level error rates; bitrate is the mean serialized-container bitrate. Bold identifies the best point estimate among reconstructed conditions, not statistical significance.

Condition	CER (%)	WER (%)	kbps
Original waveform	0.5479	1.9180	–
Fixed $D = 4$	0.8447	2.7624	6.0823
Acoustic dynamic $D = 4$	0.8516	2.6297	6.0476
Semantic joint $D = 4$	**0.8379**	**2.5935**	**6.0424**

it is universally optimal. The oracle is a privileged utility reference, not a certified optimum for reconstructed waveform quality.

The semantic head is trained for 5,000 steps using 1,000 speech utterances, with the shared trunk and acoustic head frozen. On 80 held-out utility records, pooled prediction–target Pearson correlations are 0.9906 for acoustic utility and 0.7329 for semantic utility; the corresponding R^2 values are 0.9792 and 0.5357. These aggregate statistics include all layer outputs and do not imply comparable accuracy within each layer.

Despite weaker semantic prediction, learned joint allocation at $D = 4$ reduces mean HuBERT distortion from 0.125546 to 0.108630, a 13.47% reduction, with a 0.61% Log-STFT increase. Thus, objective-dependent allocation remains effective without modifying the frozen codec.

D.3 Frozen External ASR

Because HuBERT distortion also defines the semantic training target, we additionally evaluate reconstructions using a separate frozen ASR model, `facebook/wav2vec2-large-960h-lv60-self`, with greedy CTC decoding (Baevski et al., 2020). Evaluation uses 500 LibriSpeech test-clean utterances from 40 speakers. All reconstructed conditions use utterance-level allocation at $D = 4$.

Joint allocation reduces the WER point estimate by 0.1689 percentage points, or 6.11% relative to fixed depth, while using a lower bitrate. We assess uncertainty using 10,000 paired speaker-cluster bootstrap resamples, recomputing corpus error rates from the sampled speaker totals.

The pointwise 95% WER-difference intervals, in percentage points, are $[-0.3813, +0.1302]$ for dynamic minus fixed, $[-0.3439, 0.0000]$ for joint minus fixed, and $[-0.2641, +0.1858]$ for joint minus dynamic. All include zero, so the ASR results support a favorable point estimate but not a statistically conclusive improvement.

Direct HuBERT-distortion analyses use development subsets, whereas this external ASR evaluation uses held-out test utterances. Together, the results support a semantic–acoustic allocation trade-off,

Table 20: Cross-rate CTC results. CER and WER are fractions, not percentages; error rates are mean ± standard deviation across five seeds. Bitrates are means over test utterances and are shared across seeds.

Depth	Condition	CER	WER	kbps
2	Fixed	0.614976 ± 0.027256	1.008878 ± 0.006890	3.080988
	Dynamic	0.610068 ± 0.025802	1.008154 ± 0.006409	3.052069
	Joint	0.607274 ± 0.026936	1.009530 ± 0.006924	3.045245
3	Fixed	0.601594 ± 0.029039	1.009650 ± 0.006415	4.582023
	Dynamic	0.594388 ± 0.026446	1.009867 ± 0.008351	4.551451
	Joint	0.595302 ± 0.026673	1.010856 ± 0.007094	4.543908
4	Fixed	0.599269 ± 0.027084	1.005959 ± 0.006793	6.082280
	Dynamic	0.592434 ± 0.026855	1.008999 ± 0.009020	6.047587
	Joint	0.592233 ± 0.027896	1.011508 ± 0.007661	6.042355
6	Fixed	0.604717 ± 0.027794	1.008323 ± 0.006420	9.084434
	Dynamic	0.597712 ± 0.026994	1.008106 ± 0.006189	9.045284
	Joint	0.597580 ± 0.026467	1.007527 ± 0.006790	9.038676

Table 21: Paired CER differences across five seeds. Each ± value is the half-width of a pointwise 95% Student-t interval. All intervals exclude zero; both comparisons improve in all five seeds at every depth.

Depth	Dynamic − Fixed	Joint − Fixed
2	-0.004908 ± 0.002745	-0.007703 ± 0.000525
3	-0.007205 ± 0.003887	-0.006292 ± 0.003339
4	-0.006835 ± 0.001601	-0.007036 ± 0.001498
6	-0.007004 ± 0.003676	-0.007137 ± 0.002906

with more limited evidence of downstream recognition benefits. They do not establish semantic-aware causal streaming or generalization of the semantic head to non-speech audio.

E Full Frozen-Token Representation Results

We freeze the codec and utility controllers before downstream training. Probes consume active RVQ indices and their depth/expert metadata; inactive layers are masked. "Dynamic" denotes acoustic utterance-level allocation, while "joint" denotes speech-only semantic-aware utterance-level allocation. All rates include serialized side information.

Experiments use five probe-training seeds, 2040–2044. Condition summaries report mean ± standard deviation; paired differences report mean ± the half-width of a pointwise 95% Student-t interval with $df = 4$. These intervals characterize seed variability on fixed evaluation data and are unadjusted for multiple comparisons. Differences are calculated before rounding.

E.1 Cross-Rate Character Recognition

For each seed, one shared CTC probe is trained on dynamic tokens at $D \in \{2, 3, 4, 6\}$. The 5,000 source utterances yield 20,000 utterance–rate pairs. The encoder uses active-layer masking, aggregation normalized by the square root of active depth, $4\times$ temporal subsampling, and a two-layer Transformer. Training runs for 20,000 steps, with checkpoint selection based on mean development CER across the four dynamic conditions.

Each probe is evaluated on fixed, dynamic, and joint tokens at all four rates on the same 500 test utterances, giving 60 evaluations across five seeds.

Dynamic and joint tokens reduce CER at all four evaluated rates. However, the probe is trained on dynamic tokens at those same rates, so this is a shared multi-rate representation diagnostic, not a symmetric training comparison or a test of unseen-rate generalization. Greedy WER remains near

Table 22: Unseen-speaker verification results, mean ± standard deviation across five seeds. EER is a fraction; minDCF is normalized. Bitrates match the corresponding conditions in Table 20.

Depth	Condition	EER ↓	minDCF ↓
2	Fixed	0.166300 ± 0.009085	0.933680 ± 0.024685
	Dynamic	0.160640 ± 0.004255	0.938709 ± 0.024173
	Joint	0.160044 ± 0.005024	0.941122 ± 0.022184
3	Fixed	0.170977 ± 0.004921	0.921212 ± 0.027045
	Dynamic	0.160026 ± 0.005250	0.940062 ± 0.020172
	Joint	0.161162 ± 0.004663	0.937215 ± 0.023602
4	Fixed	0.178204 ± 0.004632	0.932332 ± 0.013108
	Dynamic	0.161632 ± 0.004057	0.949428 ± 0.023620
	Joint	0.163372 ± 0.004526	0.938332 ± 0.022727
6	Fixed	0.196291 ± 0.013163	0.942388 ± 0.017924
	Dynamic	0.163945 ± 0.004219	0.967044 ± 0.023801
	Joint	0.169228 ± 0.003234	0.959892 ± 0.026443

Table 23: Paired EER differences, mean ± pointwise 95% interval half-width. Intervals exclude zero at $D = 3, 4, 6$, with 5/5 seed wins for both comparisons. At $D = 2$, both have 4/5 wins but inconclusive intervals.

Depth	Dynamic − Fixed	Joint − Fixed
2	-0.005660 ± 0.009301	-0.006256 ± 0.008753
3	-0.010951 ± 0.002664	-0.009815 ± 0.001270
4	-0.016573 ± 0.004963	-0.014833 ± 0.007337
6	-0.032345 ± 0.013059	-0.027062 ± 0.012910

one; the frozen external ASR evaluation in Appendix D.3 provides a more interpretable recognition measure.

E.2 Unseen-Speaker Verification

The speech representation splits contain disjoint speaker sets: 251 training speakers and 40 speakers each in development and test. A shared probe is trained on dynamic tokens at all four rates using mean–standard-deviation pooling, a 256-dimensional normalized embedding, and AAM-Softmax supervision on training speakers. Checkpoint selection uses mean development EER across dynamic rates.

Test evaluation uses all unordered pairs of 500 utterances: 3,440 same-speaker and 121,310 different-speaker trials. Scores are cosine similarities. We report EER and normalized minDCF with target prior 0.01 and unit miss/false-alarm costs.

EER improves at $D = 3, 4, 6$, but minDCF does not improve uniformly. As with CTC, the probe is trained on dynamic conditions, so fixed-token evaluation involves a condition shift.

Joint allocation also exhibits trade-offs relative to acoustic allocation: at $D = 6$, joint-minus-dynamic EER is $+0.005283 \pm 0.004204$; at $D = 4$, joint-minus-dynamic minDCF is -0.011096 ± 0.004312. Both pointwise intervals exclude zero. These findings show task- and operating-point-dependent effects, rather than uniform improvement from semantic allocation.

E.3 FSD50K Environmental-Sound Classification

We use the official FSD50K train/development/test partitions containing 36,796, 4,170, and 10,231 examples. A shared 200-class multi-label probe is trained on fixed and dynamic $D = 4$ tokens. Checkpoint selection uses mean fixed/dynamic development macro-mAP.

Table 24: Five-seed FSD50K results. Quality entries are mean ± standard deviation; differences and intervals are paired across seeds.

Metric	Fixed	Dynamic	Difference	95% paired CI
Macro-mAP	0.068051 ± 0.000820	0.069684 ± 0.001025	$+0.001633$	$[+0.001204, +0.002062]$
Micro-mAP	0.201742 ± 0.001198	0.203521 ± 0.001463	$+0.001778$	$[-0.000327, +0.003883]$
Rate (kbps)	6.109820	6.025438	-0.084382	–

Table 25: Five-seed MTG-Jamendo Top-50 test results at $D = 4$. Values are means; confidence intervals apply to paired seed differences. Dynamic bitrate is 0.2582% lower.

Metric	Fixed	Dynamic	Difference	95% paired CI
Macro-mAP	0.100553	0.100682	$+0.000128$	$[-0.000542, +0.000799]$
Micro-mAP	0.211426	0.212286	$+0.000859$	$[-0.000375, +0.002093]$
Rate (kbps)	6.023859	6.008305	-0.015554	–

Dynamic macro-mAP improves in all five seeds while using 1.381% less serialized bitrate. Micro-mAP improves in four seeds, but its paired interval includes zero. The supported classification gain is therefore specific to macro-mAP.

E.4 Artist-Disjoint Music Tagging

For the MTG-Jamendo downstream probe, we combine the available training and validation token sets and repartition them by artist. The new train/development/test splits contain 44,518/5,600/5,583 recordings from 2,915/309/349 artists, with no artist overlap between these probe splits. The vocabulary comprises the 50 most frequent tags in the new training split. Each seed trains a shared fixed/dynamic multi-label probe.

This artist-disjoint property applies to downstream probe training. The split was constructed after upstream utility training and does not establish that every test recording or artist was absent from all upstream training data.

Macro-mAP improves in four seeds and micro-mAP in all five, but both intervals include zero. We report no statistically significant difference on these metrics; these comparisons do not establish performance equivalence or non-inferiority.

E.5 Clotho Audio–Text Retrieval

Clotho evaluation uses 3,839 training, 1,045 validation, and 1,045 test audio examples. The test set contains 5,225 captions. A shared fixed/dynamic audio–text probe is trained for each seed, with selection based on mean validation recall across conditions.

Mean recall is the unweighted average of audio-to-text and text-to-audio R@1, R@5, and R@10. Audio-to-text retrieval counts a query as successful when at least one associated caption is retrieved within the specified rank.

The dynamic mean-recall point estimate is 4.08% higher, with three seed wins and 0.6002% lower bitrate, but its interval includes zero. A2T R@10 improves in four seeds with one tie and is the only individual recall metric whose pointwise interval excludes zero. We treat it as a secondary finding, not evidence of an overall retrieval improvement.

E.6 Summary Across Tasks

The frozen-token results are task dependent. Shared multi-rate probes show lower CER and, at $D = 3, 4, 6$, lower speaker EER; a probe trained jointly on fixed/dynamic tokens also shows higher FSD50K macro-mAP. Speaker minDCF exhibits different trade-offs. The primary MTG and Clotho intervals include zero despite positive point estimates and lower bitrates. These results support selective benefits of adaptive allocation, without establishing universal downstream improvement.

Table 26: Clotho test results at $D = 4$, averaged across five seeds. Intervals are pointwise and unadjusted across retrieval metrics.

Metric	Fixed	Dynamic	Difference	95% paired CI
Mean recall	0.022960	0.023898	$+0.000938$	$[-0.000859, +0.002734]$
A2T R@10	0.042297	0.045550	$+0.003254$	$[+0.000043, +0.006464]$
Rate (kbps)	6.025979	5.989811	-0.036168	–

Table 27: Mean serialized bitrate of source token sequences used for the listening set. Playback excerpts are cropped after reconstruction. Percentage changes are computed from the domain mean rates.

Domain	Fixed (kbps)	Dynamic (kbps)	Rate change
Speech	6.0567	6.0353	-0.3528%
Music	6.0238	6.0090	-0.2465%
Environment	6.0308	5.9900	-0.6768%

F Subjective Evaluation and External Codec Comparison

The listening study evaluates perceptual changes within the same frozen backbone, while the external comparison contextualizes absolute speech reconstruction quality. Both use acoustic utterance-level allocation, not the causal streaming allocator.

F.1 Three-Domain MUSHRA Protocol

We conduct a MUSHRA-style study based on the reference-and-anchor procedure of ITU-R BS.1534 (International Telecommunication Union, 2015). The set contains 30 trials: 10 speech, 10 music, and 10 environmental examples. Speech excerpts last 5 s; music and environmental excerpts last 10 s. All stimuli are stored as 48-kHz mono PCM16.

Each trial provides a visible reference and four hidden conditions: the hidden reference, fixed $D = 4$ reconstruction, dynamic $D = 4$ reconstruction, and a low-pass anchor generated with a biquad filter at 3.5 kHz. A common peak-scaling factor is applied to all conditions within a trial. Hidden-condition labels and trial order are randomized when generating the listening set; the same generated set is used for all listeners.

Twenty listeners complete all 30 trials, yielding $20 \times 30 \times 4 = 2{,}400$ ratings. All pass the applied hidden-reference quality check, so none are excluded. For each comparison, ratings are first averaged within each listener over the relevant trials, and the resulting listener-level differences form the paired statistical observations.

F.2 Listening-Set Bitrates

Table 27 reports the container bitrates recorded for the source token sequences used to generate listening stimuli. The reconstructions are subsequently cropped to the 5- or 10-second listening excerpts. These rates therefore describe the source representations, not separately encoded containers for the cropped playback excerpts.

F.3 MUSHRA Results

The hidden reference and anchor receive overall mean scores of 97.05 and 32.32, respectively. Fixed and dynamic reconstruction receive 68.09 and 69.03.

The overall paired difference is $+0.937$ points, with positive differences for 15 of 20 listeners. Its 95% Student-t interval is $[-0.195, +2.068]$, the participant-bootstrap interval is $[-0.117, +1.950]$, and the paired t-test gives $p = 0.0994$.

Speech improves by 3.52 points with an interval excluding zero. Music, environmental audio, and the overall comparison remain inconclusive; the environmental point estimate favors fixed coding.

Table 28: MUSHRA-style results for utterance-level allocation. Intervals are paired listener-level 95% Student-t intervals ($df = 19$), unadjusted for multiple comparisons.

Domain	Fixed	Dynamic	Difference	95% paired CI
Speech	72.88	76.40	$+3.52$	$[+2.12, +4.92]$
Music	65.33	65.84	$+0.51$	$[-0.96, +1.97]$
Environment	66.06	64.85	-1.22	$[-2.71, +0.28]$
Overall	68.09	69.03	$+0.94$	$[-0.20, +2.07]$

Table 29: Reconstruction on 2,620 test-clean utterances with a common 20-second duration cap. Values are means over utterances. Rates include each system's actual container overhead.

System	PESQ ↑	STOI ↑	kbps
Official EnCodec	2.747038	0.937710	6.112894
UniAdapt fixed $D = 4$	1.912894	0.870564	6.075926
UniAdapt dynamic $D = 4$	2.042566	0.888684	6.043176

Thus, the study supports a speech-specific subjective benefit, without establishing a general three-domain improvement or equivalence in the other domains. Listener-level inference is conditional on the selected listening items.

F.4 External EnCodec Comparison

We compare official 24-kHz EnCodec at its nominal 6-kbps, eight-codebook operating point with fixed and dynamic $D = 4$ UniAdapt. Evaluation uses all 2,620 LibriSpeech test-clean utterances, each truncated to at most 20 s, with the same PESQ/STOI evaluation procedure. Audio is converted to mono and resampled to each codec's input rate without peak normalization. PESQ uses wideband evaluation at 16 kHz.

UniAdapt uses the three-domain acoustic controller and utterance-level allocation. EnCodec rate is measured using its official compression interface with `use_lm=False`; UniAdapt rate uses its dynamic container, including all side information and padding.

Uncertainty is estimated using 10,000 paired utterance-bootstrap resamples. Differences in Table 30 are candidate minus reference; intervals are pointwise.

Table 30: Paired differences over the same 2,620 utterances. Positive PESQ/STOI differences favor the candidate; negative rate differences indicate fewer bits per second.

Comparison / metric	Difference	95% bootstrap CI
Dynamic − Fixed, PESQ	$+0.129672$	$[+0.122993, +0.136169]$
Dynamic − Fixed, STOI	$+0.018120$	$[+0.017752, +0.018490]$
Dynamic − Fixed, kbps	-0.032750	$[-0.034445, -0.031083]$
Dynamic − EnCodec, PESQ	-0.704472	$[-0.712897, -0.696145]$
Dynamic − EnCodec, STOI	-0.049027	$[-0.049624, -0.048433]$
Dynamic − EnCodec, kbps	-0.069718	$[-0.072230, -0.067265]$

Dynamic allocation improves PESQ and STOI over the same frozen UniAdapt backbone while using 0.5390% less mean serialized bitrate; all three paired intervals exclude zero. EnCodec achieves higher absolute PESQ and STOI, while UniAdapt dynamic uses 1.1405% less mean bitrate than EnCodec.

The systems differ in architecture, sample rate, training objective, quantization, and container format. This comparison therefore contextualizes reconstruction quality rather than isolating the effect of allocation across architectures.

F.5 Overall Evaluation Scope

Within the frozen backbone, dynamic allocation improves objective speech reconstruction and produces a speech-specific listening benefit. Other listening domains and the primary music-tagging and retrieval summaries remain statistically inconclusive. The external comparison also shows that allocation gains do not remove limitations of the underlying codec. The evidence supports content-adaptive allocation under explicit serialized budgets, without establishing universal downstream gains or reconstruction state of the art.