BEYOND RECONSTRUCTION: UNDERSTANDING DISCRETE REPRESENTATIONS FOR AUTOREGRESSIVE WEARABLE SENSOR DATA GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete tokenization provides an interface between continuous wearable sensor data and autoregressive sequence models. Existing wearable sensor tokenizers are commonly optimized and evaluated for reconstruction, yet whether reconstruction quality is a reliable indicator of downstream autoregressive generation performance remains unclear. We systematically investigate this question using three vector-quantized tokenizers and T5-based autoregressive models. We find that reconstruction fidelity does not consistently reflect downstream generation quality. We further analyze token prediction errors and find that larger target–prediction codeword distances are consistently associated with larger signal-level distortions after decoding. Motivated by this finding, we propose HierCode, a coarse-to-fine generation framework that groups nearby codewords of a pretrained tokenizer into clusters. HierCode first autoregressively generates cluster tokens and then recovers fine-grained tokens in parallel conditioned on the generated cluster sequence. To improve robustness to imperfect cluster predictions, the recovery model is trained with perturbed cluster sequences and incorporates the coarse model's predictive distribution as a soft cluster prior at inference. Across five wearable sensing benchmarks and three tokenizer architectures, HierCode improves over direct fine-token autoregressive generation in the majority of evaluated settings.

1 INTRODUCTION

Wearable sensors have become an essential component in intelligent systems, enabling a wide range of applications, including health monitoring (Lara & Labrador, 2012; Xiao et al., 2025), smart homes (Lu et al., 2020; Miao et al., 2022), and industrial automation (Tang et al., 2022; Miao et al., 2024). However, collecting large-scale wearable sensor data can be costly and constrained by privacy considerations (Rousseau et al., 2026), motivating growing interest in generative models for wearable sensor data. Recent advances in sequence modeling provide an opportunity to model such signals using autoregressive architectures developed for discrete sequences (Xue & Salim, 2023; Kim et al., 2024; Merrill et al., 2026). A central challenge is therefore how to represent continuous, multivariate sensor signals as discrete sequences suitable for autoregressive modeling.

Discrete tokenization provides a natural interface between continuous sensor data and autoregressive sequence models. Vector-quantized models, such as VQ-VAE (van den Oord et al., 2017), encode continuous inputs into latent representations and map them to codewords from codebooks. The resulting codeword indices form discrete token sequences that can subsequently be modeled autoregressively. This paradigm has been increasingly explored for time-series representation and generation (Chung et al., 2023; Talukder et al., 2024; Chen et al., 2024), alongside more expressive tokenization schemes based on residual (Lee et al., 2022; Ray et al., 2025) and multi-granularity representations (Xu et al., 2026; Lee et al., 2026). Despite these advances, discrete tokenizers are commonly optimized and evaluated according to reconstruction fidelity. However, whether reconstruction fidelity reliably reflects downstream autoregressive generation quality remains unclear. Reconstruction evaluates how faithfully a tokenizer can represent and recover an observed signal through its discrete representation. In autoregressive generation, by contrast, an autoregressive model predicts token sequences, which are then mapped to their corresponding codeword embeddings and decoded into the wearable sensor data.

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.

To investigate this question, we conduct a systematic analysis using three vector-quantized tokenizers, VQ-VAE (van den Oord et al., 2017), RQ-VAE (Lee et al., 2022), and VAEVQ (Yang et al., 2026), whose token sequences are modeled using same autoregressive architecture. Our analysis finds that improvements in reconstruction fidelity do not consistently reflect downstream generation quality. This discrepancy motivates us to examine a behavior specific to autoregressive generation: token prediction errors. By evaluating prediction errors produced by the autoregressive model, we find that larger distances between target and predicted codewords are consistently associated with larger signal distortions after decoding.

Motivated by this relationship, we propose **HierCode**, a coarse-to-fine generation framework that organizes the fine-token vocabulary according to codeword proximity. Given a pretrained tokenizer, HierCode groups neighboring codewords into clusters and first autoregressively generates the corresponding cluster-token sequence. It then recovers the fine-grained tokens in parallel conditioned on the generated cluster sequence. To improve robustness to imperfect cluster generation, the fine-token recovery model is trained with perturbed cluster sequences and incorporates the coarse model's predictive distribution as a soft cluster prior at inference. The recovered fine-token sequence is then decoded using the pretrained tokenizer. Across multiple wearable datasets and tokenizer architectures, HierCode improves over direct fine-token autoregressive generation in the majority of evaluated settings without modifying the pretrained tokenizer. To the best of our knowledge, this is the first study of whether reconstruction fidelity of discrete wearable sensor data representations reflects their downstream autoregressive generation performance. Our main contributions are:

- We systematically examine the relationship between reconstruction fidelity and downstream autoregressive generation across three vector-quantized tokenizers and five wearable sensor data benchmarks, showing that reconstruction fidelity does not consistently reflect generation quality.
- We analyze token errors produced during autoregressive generation and identify a consistent association between target–prediction codeword distance and the resulting signal-level distortion.
- Motivated by this finding, we propose HierCode, a coarse-to-fine generation framework that organizes a pretrained fine-token vocabulary according to codeword proximity and improves over direct fine-token autoregressive generation in the majority of evaluated settings without modifying the underlying tokenizer.

2 PRELIMINARIES

Discrete autoregressive generation with vector-quantized representations consists of two components: discrete tokenization and autoregressive token modeling. A tokenizer maps a continuous sensor sequence to discrete token indices and reconstructs the signal through their corresponding codeword embeddings. An autoregressive model subsequently learns the distribution of these token sequences for generation.

2.1 STAGE I: DISCRETE TOKENIZATION

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_L]$ denote a multivariate wearable sensor sequence of length L, where $\mathbf{x}_t \in \mathbb{R}^d$ and d is the number of channels. An encoder $\mathcal{E}_{\text{enc}}$ maps $\mathbf{X}$ to a latent sequence $\mathbf{Z} = \mathcal{E}_{\text{enc}}(\mathbf{X}) = [\mathbf{z}_1, \ldots, \mathbf{z}_N], \mathbf{z}_i \in \mathbb{R}^{d_z}$, where N and d_z denote the latent sequence length and latent dimension, respectively.

Given a codebook $\mathcal{B} = \{\mathbf{e}_k\}_{k=1}^K$, where $\mathbf{e}_k \in \mathbb{R}^{d_z}$ and K is the codebook size, a latent vector can be quantized by nearest-codeword assignment: $c_i = \arg\min_{k \in \{1,\ldots,K\}} \|\mathbf{z}_i - \mathbf{e}_k\|_2^2$. This yields the discrete token sequence $\mathbf{C} = [c_1, \ldots, c_N]$ and the corresponding quantized latent sequence $\mathbf{Z}^q = [\mathbf{e}_{c_1}, \ldots, \mathbf{e}_{c_N}]$. The reconstructed signal is obtained as $\widetilde{\mathbf{X}} = \mathcal{D}(\mathbf{Z}^q)$, where $\mathcal{D}$ denotes the tokenizer decoder. The tokenizer is trained using a reconstruction objective together with architecture-specific quantization objectives.

2.2 Stage II: Autoregressive Token Modeling

Given a conditioning context $\mathbf{C}^{\mathrm{ctx}}$, an autoregressive model parameterized by θ factorizes the conditional distribution of the token sequence as

$$P_\theta(\mathbf{C} \mid \mathbf{C}^{\mathrm{ctx}}) = \prod_{i=1}^{N} P_\theta(c_i \mid c_{<i}, \mathbf{C}^{\mathrm{ctx}}), \tag{1}$$

where $c_{<i}$ denotes the tokens preceding position i.

At inference, the model sequentially predicts the target tokens conditioned on $\mathbf{C}^{\mathrm{ctx}}$ to obtain $\widehat{\mathbf{C}}$. The predicted token indices are mapped to their corresponding codeword embeddings and decoded using the tokenizer decoder: $\widehat{\mathbf{X}} = \mathcal{D}(\widehat{\mathbf{Z}}^q)$.

3 Analyzing Discrete Representations for Autoregressive Wearable Generation

We examine discrete representations from the perspective of downstream autoregressive generation. Specifically, we ask two questions: (1) Does reconstruction fidelity consistently reflect autoregressive generation quality, and (2) When token prediction errors occur, is their signal-level severity associated with the distance between target and predicted codewords?

3.1 Experimental Setup

Datasets. We evaluate on five public wearable and physiological time-series benchmarks: HHAR (Stisen et al., 2015), UCIHAR (Anguita et al., 2013), WISDM (Kwapisz et al., 2011), PAMAP2 (Reiss & Stricker, 2012), and PTB (Bousseljot et al., 1995). We use fixed training, validation, and test partitions with a 70%/15%/15% split across all experiments. Dataset statistics are provided in Appendix Table 5.

Models. We consider three vector-quantized tokenizers: VQ-VAE (van den Oord et al., 2017), RQ-VAE (Lee et al., 2022), and VAEVQ (Yang et al., 2026). After training, each tokenizer is frozen, and its token sequences are modeled using the same FLAN-T5-Base (Ni et al., 2022) architecture.

Evaluation Metrics. We use reconstruction mean squared error (rMSE) to measure the discrepancy between the input data and its tokenizer reconstruction. Generation mean squared error (gMSE) measures the discrepancy between the generated and ground-truth sequences.

3.2 Does Reconstruction Fidelity Reflect Autoregressive Generation Quality?

We first examine whether reconstruction fidelity provides a reliable indication of downstream autoregressive generation quality. For each tokenizer, we vary the codebook size as $K \in \{64, 128, 256, 512, 1024\}$ while keeping the remaining architecture, optimization settings, and training protocol fixed. Detailed hyperparameter settings are provided in Appendix B.1.

Appendix Figure. 4 characterizes the reconstruction performance under these configurations. Increasing K affects the three tokenizers differently, yielding configurations with varying reconstruction fidelity. We use these configurations to test whether their ordering according to reconstruction quality is preserved when the resulting tokens are modeled autoregressively.

For each configuration, we train the same FLAN-T5-Base model on the resulting token sequences using an identical autoregressive training and evaluation protocol. We measure reconstruction and generation performance using rMSE and gMSE, respectively, and compute Spearman's rank correlation across the five codebook sizes to quantify the rank association between reconstruction and generation performance. Figure. 1 shows that this relationship is not consistent across tokenizer–dataset settings. For example, on UCIHAR, lower rMSE generally corresponds to lower gMSE under VAEVQ and VQ-VAE, whereas on PAMAP2 the relationship is consistently reversed across the three tokenizers. The relationship can also differ across tokenizers on the same dataset. For example, on WISDM, rMSE and gMSE are positively associated for VAEVQ but negatively associated for VQ-VAE and RQ-VAE, while PTB exhibits the opposite pattern.

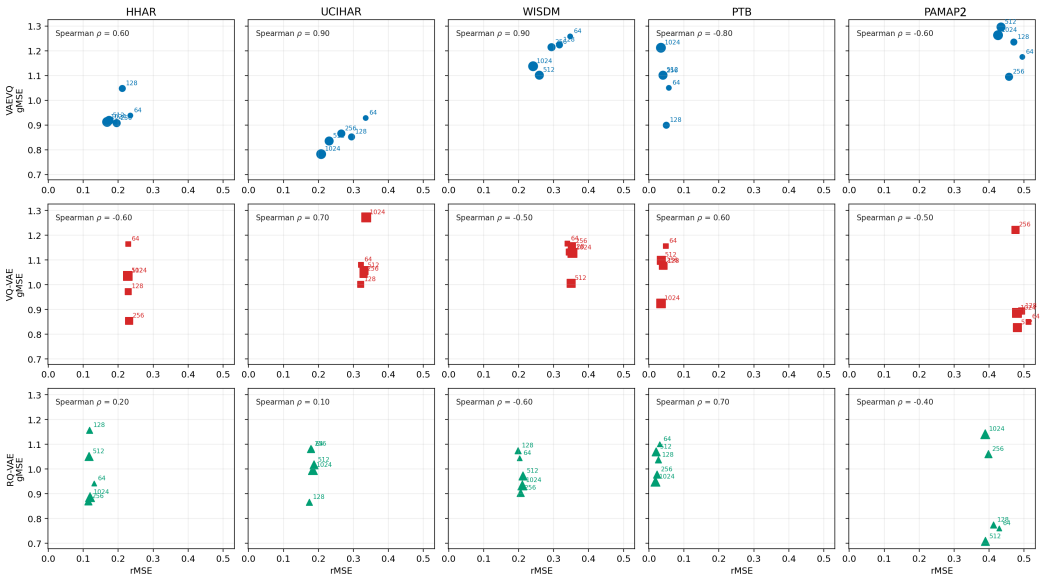

Figure 1: Relationship between rMSE and gMSE across codebook sizes $K \in \{64, 128, 256, 512, 1024\}$. Each point denotes one codebook configuration, and Spearman's ρ reports the rank association within each tokenizer–dataset setting. The varying associations across settings show that reconstruction-based configuration rankings do not consistently align with downstream generation performance.

Changing K, however, affects not only reconstruction fidelity but also other properties of the discrete representation, including the vocabulary size presented to the autoregressive model. We further inspect comparisons at matched codebook sizes. For example, at $K = 128$ on HHAR, RQ-VAE achieves the lowest rMSE but a higher gMSE than both VAEVQ and VQ-VAE. Moreover, although VQ-VAE has a higher rMSE than VAEVQ, it achieves a lower gMSE. Thus, the reconstruction ranking is not preserved under autoregressive generation even among configurations with the same codebook size. Overall, these results show that reconstruction fidelity does not provide a consistently reliable indicator of downstream autoregressive generation performance.

3.3 Is Token-Error Severity Associated with Codeword Distance?

The preceding analysis shows that reconstruction fidelity does not consistently reflect downstream autoregressive generation performance. We therefore turn to a behavior that arises specifically during autoregressive generation: token prediction errors. When a predicted token differs from its target, the corresponding codeword embedding presented to the decoder also changes. We examine whether the signal-level consequence of such an error is associated with the distance between the target and predicted codewords.

For each erroneous target–prediction $(c_i, \hat{c}_i)$ produced by the autoregressive model, where $\hat{c}_i \neq c_i$, we measure the corresponding distance i as $d_i = \|\mathbf{e}_{c_i} - \mathbf{e}_{\hat{c}_i}\|_2$. Because multiple prediction errors may occur within the same generated sequence and their individual signal-level consequences cannot be separated from the fully generated output. We therefore evaluate each observed error individually. Given the target token sequence $\mathbf{C} = [c_1, \ldots, c_N]$, for an error at position i we construct $\mathbf{C}^{(i)} = [c_1, \ldots, c_{i-1}, \hat{c}_i, c_{i+1}, \ldots, c_N]$, where only the target token c_i is replaced by its corresponding autoregressive prediction $\hat{c}_i$ and all other positions remain unchanged. We measure the resulting single-error distortion as $\Delta_i = \mathrm{MSE}\left(\mathcal{D}(\mathbf{C}), \mathcal{D}(\mathbf{C}^{(i)})\right)$, where $\mathcal{D}(\mathbf{C})$ denotes codebook lookup followed by decoding.

Figure. 2 shows the relationship between codeword distance d_i and single-error distortion Δ_i. Positive associations are observed across all tokenizer–dataset settings. Spearman's ρ ranges from 0.41

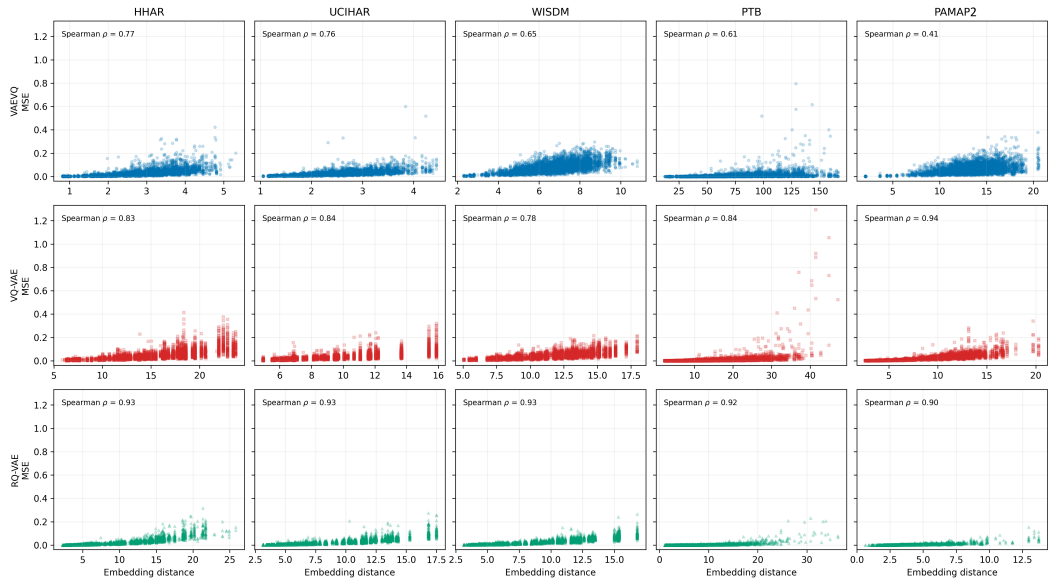

Figure 2: Relationship between target–prediction codeword distance and decoded distortion for autoregressive token errors. Each point represents an erroneous token prediction. The x-axis reports the Euclidean distance between the target and predicted codeword embeddings. The y-axis shows the distortion between the resulting decoded sequence and the target sequence.

to 0.77 for VAEVQ, from 0.78 to 0.94 for VQ-VAE, and from 0.90 to 0.93 for RQ-VAE. Thus, among the errors produced by the autoregressive model, predictions farther from their target codewords tend to induce larger decoded distortion. Although the magnitude of Δ_i can also depend on the token position and sequence context, the consistently positive association across tokenizer–dataset settings indicates that target–prediction codeword distance is informative of the signal-level severity of autoregressive token errors.

4 HierCode

The preceding analysis shows that, among autoregressive token errors, predictions farther from their target codewords tend to induce larger signal-level distortions. This suggests that target–prediction codeword proximity is relevant to the consequences of autoregressive prediction errors. Standard autoregressive modeling, however, predicts directly over fine-grained token indices without explicitly representing proximity among their corresponding codewords.

Motivated by this observation, we propose HierCode, a coarse-to-fine generation framework that organizes the fine-grained token vocabulary according to codeword proximity. As illustrated in Figure. 3, we partition the K codewords of a pretrained tokenizer into M clusters, and map each fine-grained token to the cluster containing its codeword. HierCode first autoregressively generates the resulting cluster-token sequence and then recovers the fine-grained token at each position conditioned on the generated cluster sequence.

4.1 Codeword Cluster Construction

Given a pretrained tokenizer with codebook $\mathcal{B} = \{\mathbf{e}_k\}_{k=1}^{K}$, we construct a coarse vocabulary by partitioning its K codewords into M clusters using K-means, where $M < K$. Specifically, K-means is applied to the learned codeword embeddings $\{\mathbf{e}_k\}_{k=1}^{K}$ using Euclidean distance, yielding $\mathcal{G} = \{\mathcal{G}_m\}_{m=1}^{M}$. This induces a fixed codeword-to-cluster mapping $g : \{1, \ldots, K\} \to \{1, \ldots, M\}$, $g(k) = m \iff \mathbf{e}_k \in \mathcal{G}_m$.

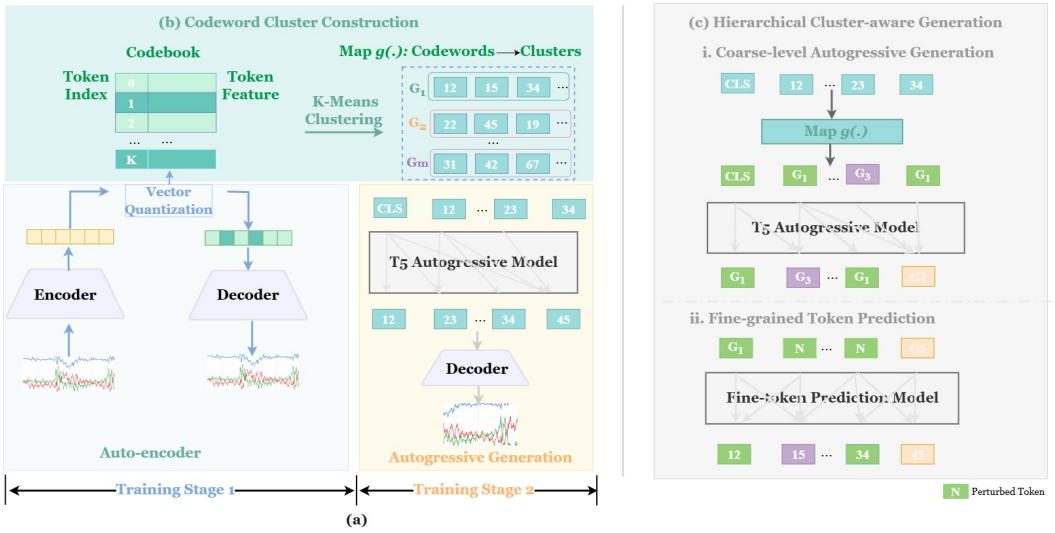

Figure 3: Overview of discrete autoregressive wearable sensor data generation and the proposed HierCode framework. (a) Standard discrete autoregressive generation, consisting of sensor tokenization followed by autoregressive token modeling. (b) Codeword cluster construction, which groups codewords according to their embedding distance to define a fixed mapping from fine-grained tokens to cluster tokens. (c) HierCode generation, which autoregressively generates cluster tokens and then recovers fine-grained tokens in parallel conditioned on the generated cluster sequence.

Given a fine-grained token sequence $\mathbf{C}^f = [c_1^f, c_2^f, \ldots, c_N^f], c_i^f \in \{1, \ldots, K\}$, we obtain its cluster representation by $c_i^c = g(c_i^f)$. The resulting cluster sequence is $\mathbf{C}^c = [c_1^c, c_2^c, \ldots, c_N^c], c_i^c \in \{1, \ldots, M\}$.

4.2 Hierarchical Cluster-aware Generation

Given the codeword-to-cluster mapping, we formulate HierCode through coarse-level autoregressive generation and fine-grained token recovery.

4.2.1 Coarse-level Autoregressive Generation

Given a conditioning sequence $\mathbf{C}^{\text{ctx}}$, the coarse autoregressive model parameterized by θ factorizes the conditional distribution of the cluster-token sequence as

$$P_\theta\left(\mathbf{C}^c \mid \mathbf{C}^{\text{ctx}}\right) = \prod_{i=1}^{N} P_\theta\left(c_i^c \mid c_{<i}^c, \mathbf{C}^{\text{ctx}}\right). \tag{2}$$

The model is trained using next-token negative log-likelihood,

$$\mathcal{L}_{\text{coarse}} = -\sum_{i=1}^{N} \log P_\theta\left(c_i^c \mid c_{<i}^c, \mathbf{C}^{\text{ctx}}\right). \tag{3}$$

4.2.2 Fine-grained Token Recovery

Given a cluster-token sequence $\mathbf{C}^c$, the recovery model parameterized by ϕ models the conditional distribution of the fine-token sequence as

$$P_\phi\left(\mathbf{C}^f \mid \mathbf{C}^c\right) = \prod_{i=1}^{N} P_\phi\left(c_i^f \mid \mathbf{C}^c\right). \tag{4}$$

Table 1: Comparison of generation performance across five benchmark datasets. Generation quality is evaluated by gMSE.

Model	Tokenizer	Method	HHAR	UCIHAR	WISDM	PTB	PAMAP2	Average.↓
T5-Base	VAEVQ	Vanilla AR	0.9702 ± 0.0032	0.8926 ± 0.0015	**1.0617 ± 0.0011**	0.9647 ± 0.0535	1.0058 ± 0.0065	0.9790 ± 0.0108
		HierCode	**0.9480 ± 0.0192**	**0.8644 ± 0.0119**	1.0810 ± 0.0036	**0.9535 ± 0.0601**	**0.9821 ± 0.0065**	**0.9658 ± 0.0140**
	VQ-VAE	Vanilla AR	0.9976 ± 0.0181	0.9452 ± 0.0156	1.0755 ± 0.0106	**0.9971 ± 0.0636**	1.0882 ± 0.0049	1.0207 ± 0.0145
		HierCode	**0.9735 ± 0.0137**	**0.9325 ± 0.0173**	**1.0409 ± 0.0099**	0.9992 ± 0.0356	**0.9896 ± 0.0056**	**0.9871 ± 0.0115**
	RQ-VAE	Vanilla AR	**1.0052 ± 0.0110**	0.9468 ± 0.0075	1.1018 ± 0.0056	0.9532 ± 0.0412	1.1894 ± 0.0092	1.0392 ± 0.0094
		HierCode	1.0056 ± 0.0087	**0.9376 ± 0.0053**	**0.9890 ± 0.0047**	**0.9497 ± 0.0274**	**0.9827 ± 0.0154**	**0.9729 ± 0.0077**
		TimeVAE	0.9987 ± 0.0000	1.0001 ± 0.0000	0.9997 ± 0.0000	0.6971 ± 0.0095	0.9081 ± 0.0240	0.9207 ± 0.0055
		RTSGAN	0.7838 ± 0.0158	1.0121 ± 0.0369	0.6715 ± 0.0152	1.7905 ± 0.5194	1.4190 ± 0.0733	1.1354 ± 0.1118
		SDformer	0.9465 ± 0.0758	0.9723 ± 0.0394	0.8814 ± 0.0747	0.6324 ± 0.0195	0.9084 ± 0.0458	0.8682 ± 0.0201
		Chronos-2	1.0851 ±0.0123	1.0797 ±0.0088	0.8999 ± 0.0114	1.5841 ± 0.0909	1.3163 ± 0.0112	1.1930 ± 0.0172
		Moirai2	1.1196 ± 0.0114	1.1998 ± 0.0094	1.0221 ± 0.0047	1.5426 ± 0.0408	1.3587 ± 0.0108	1.2485 ± 0.0077

Table 2: Generation performance across five benchmark datasets measured by MDD.

Model	Tokenizer	Method	HHAR	UCIHAR	WISDM	PTB	PAMAP2	Average.↓
T5-Base	VAEVQ	Vanilla AR	0.0651 ± 0.0005	0.1070 ± 0.0003	0.1273 ± 0.0003	0.0827 ± 0.0049	0.2016 ± 0.0041	0.1168 ± 0.0011
		HierCode	**0.0629 ± 0.0013**	**0.0942 ± 0.0024**	**0.1196 ± 0.0006**	**0.0550 ± 0.0034**	**0.1929 ± 0.0017**	**0.1049 ± 0.0006**
	VQ-VAE	Vanilla AR	0.0708 ± 0.0032	0.1006 ± 0.0073	0.1344 ± 0.0057	**0.0535 ± 0.0036**	0.1941 ± 0.0023	0.1107 ± 0.0021
		HierCode	**0.0697 ± 0.0022**	**0.1001 ± 0.0049**	**0.1235 ± 0.0037**	0.0564 ± 0.0034	**0.1926 ± 0.0026**	**0.1085 ± 0.0012**
	RQ-VAE	Vanilla AR	0.0643 ± 0.0004	0.0960 ± 0.0023	0.1136 ± 0.0015	**0.0498 ± 0.0033**	**0.1905 ± 0.0031**	0.1028 ± 0.0010
		HierCode	**0.0628 ± 0.0008**	**0.0856 ± 0.0015**	**0.1025 ± 0.0011**	0.0516 ± 0.0029	0.1912 ± 0.0055	**0.0987 ± 0.0012**
		TimeVAE	0.3061 ± 0.0000	0.2846 ± 0.0000	0.3090 ± 0.0000	0.0968 ± 0.0051	0.3012 ± 0.0315	0.2595 ± 0.0064
		RTSGAN	0.0751 ± 0.0034	0.0570 ± 0.0039	0.0750 ± 0.0037	0.0512 ± 0.0023	0.1125 ± 0.0035	0.0742 ± 0.0018
		SDformer	0.2648 ± 0.0117	0.2483 ± 0.0087	0.2434 ± 0.0111	0.0504 ± 0.0033	0.2610 ± 0.0117	0.2136 ± 0.0060
		Chronos-2	0.1284 ± 0.0014	0.1004 ± 0.0013	0.1054 ± 0.0021	0.0672 ± 0.0018	0.0893 ± 0.0018	0.0981 ± 0.0007
		Moirai2	0.1669 ± 0.0030	0.1400 ± 0.0021	0.1979 ± 0.0022	0.0767 ± 0.0028	0.1195 ± 0.0013	0.1401 ± 0.0011

Cluster-token perturbation. During training, the recovery model is conditioned on target cluster sequences, whereas at inference it receives cluster sequences generated by the coarse autoregressive model. To improve robustness to this discrepancy, we perturb the cluster sequence during training. Each cluster token is independently replaced by a randomly sampled cluster token with probability p_{noise}, yielding a perturbed sequence $\widetilde{\mathbf{C}}^c$. The recovery model is then optimized using

$$\mathcal{L}_{\text{fine}} = -\sum_{i=1}^{N} \log P_\phi \left(c_i^f \mid \widetilde{\mathbf{C}}^c \right). \tag{5}$$

Soft cluster-prior fusion. At inference, let $\widehat{\mathbf{C}}^c = [\hat{c}_1^c, \ldots, \hat{c}_N^c]$ denote the generated cluster sequence. The coarse autoregressive model also provides its predictive distribution over the M clusters, $q_i(m) = P_\theta \left(c_i^c = m \mid \widehat{\mathbf{C}}_{<i}^c \right), m \in \{1, \ldots, M\}$.

Conditioned on $\widehat{\mathbf{C}}^c$, the recovery model produces logits $\mathbf{s}_i = [s_{i,1}, \ldots, s_{i,K}]$ over the fine-token vocabulary. For each fine token k, we incorporate the predictive probability of its associated cluster $g(k)$ as $\widetilde{s}_{i,k} = s_{i,k} + \beta \log \left(q_i(g(k)) + \epsilon \right)$, where β controls the contribution of the cluster predictive distribution and ϵ ensures numerical stability. The final fine-token probability is

$$P_\phi^{\text{final}} \left(c_i^f = k \mid \widehat{\mathbf{C}}^c \right) = \frac{\exp(\widetilde{s}_{i,k})}{\sum_{j=1}^{K} \exp(\widetilde{s}_{i,j})}. \tag{6}$$

Unlike hard restriction to the generated cluster $\hat{c}_i^c$, this soft fusion retains all K fine-token candidates while weighting them according to the coarse model's predictive distribution. The recovered fine-token sequence is finally mapped to the original codebook embeddings and decoded by the frozen tokenizer decoder.

5 Experiments

We evaluate HierCode on wearable sensor data generation across five benchmark datasets. All experiments are repeated over 10 independent runs with different random seeds, and results are reported as the mean and standard deviation. Detailed experimental settings are provided in Appendix B.2.

Table 3: Comparison of coarse-to-fine token recovery strategies across three tokenizers and five benchmark datasets, evaluated using gMSE.

Tokenizer	Dataset	*Hard Top-1*	*Hard Top-3*	*Soft Posterior*	*HierCode*
VAEVQ	HHAR	1.0325 ± 0.0479	1.0132 ± 0.0353	1.0441 ± 0.0233	**0.9480 ± 0.0192**
	UCI-HAR	0.8965 ± 0.0143	0.9073 ± 0.0110	0.8983 ± 0.0087	**0.8644 ± 0.0119**
	WISDM	**1.0008 ± 0.0064**	1.0108 ± 0.0044	1.0147 ± 0.0056	1.0810 ± 0.0036
	PTB	0.9744 ± 0.0553	0.9698 ± 0.0572	0.9602 ± 0.0580	**0.9535 ± 0.0601**
	PAMAP2	1.0454 ± 0.0064	1.0412 ± 0.0068	0.9981 ± 0.0065	**0.9821 ± 0.0065**
	Average	0.9899 ± 0.0139	0.9885 ± 0.0128	0.9830 ± 0.0131	**0.9658 ± 0.0140**
VQ-VAE	HHAR	0.9024 ± 0.0741	1.0935 ± 0.0237	**0.8966 ± 0.0293**	0.9735 ± 0.0137
	UCI-HAR	**0.8572 ± 0.0286**	0.8757 ± 0.0175	0.8622 ± 0.0150	0.9325 ± 0.0173
	WISDM	**1.0409 ± 0.0190**	1.0423 ± 0.0114	1.0412 ± 0.0104	**1.0409 ± 0.0099**
	PTB	1.0374 ± 0.0387	1.0047 ± 0.0367	1.0152 ± 0.0361	**0.9992 ± 0.0356**
	PAMAP2	0.9708 ± 0.0042	**0.9690 ± 0.0060**	0.9712 ± 0.0050	0.9896 ± 0.0056
	Average	0.9617 ± 0.0161	0.9970 ± 0.0127	**0.9573 ± 0.0132**	0.9871 ± 0.0115
RQ-VAE	HHAR	1.0558 ± 0.0108	1.0089 ± 0.0175	**0.9944 ± 0.0441**	1.0056 ± 0.0087
	UCI-HAR	1.0062 ± 0.0071	0.9937 ± 0.0062	1.0085 ± 0.0050	**0.9376 ± 0.0053**
	WISDM	1.1572 ± 0.0045	1.1554 ± 0.0045	1.1572 ± 0.0042	**0.9890 ± 0.0047**
	PTB	0.8375 ± 0.0269	0.8424 ± 0.0262	**0.8367 ± 0.0257**	0.9497 ± 0.0274
	PAMAP2	0.9499 ± 0.0157	**0.9418 ± 0.0147**	0.9767 ± 0.0134	0.9827 ± 0.0154
	Average	1.0013 ± 0.0061	0.9884 ± 0.0072	0.9947 ± 0.0099	**0.9729 ± 0.0077**

Table 4: Effect of the number of clusters M on generation performance across five benchmark datasets, evaluated using gMSE.

Tokenizer	Dataset	$\mathbf{M = 4}$	$\mathbf{M = 8}$	$\mathbf{M = 16}$	$\mathbf{M = 32}$	$\mathbf{M = 64}$
VAEVQ	HHAR	0.8368 ± 0.0078	1.0156± 0.0213	0.9480 ± 0.0192	**0.8352±0.0208**	0.9120±0.0153
	UCIHAR	0.8937± 0.0039	**0.8490 ± 0.0127**	0.8644 ± 0.0119	0.9089± 0.0090	0.8802±0.0113
	WISDM	0.9966± 0.0029	1.0413± 0.0025	1.0810 ± 0.0036	1.0028 ± 0.0080	**0.9875 ± 0.0029**
	PTB	0.9345± 0.0436	0.9684± 0.0314	0.9535 ± 0.0601	0.9043± 0.0347	**0.8349± 0.0464**
	PAMAP2	1.0071 ±0.0066	**0.9241** ± 0.0063	0.9821 ± 0.0065	1.0320± 0.0048	1.0193± 0.0028
	Average	0.9337 ±0.0095	0.9597± 0.0072	0.9658 ± 0.0140	0.9366 ± 0.0065	**0.9268± 0.0092**
VQ-VAE	HHAR	1.0282 ±0.0190	**0.9312± 0.0109**	0.9735± 0.0137	1.0184 ± 0.0165	1.0280±0.0145
	UCIHAR	1.0044 ±0.0090	0.8795± 0.0114	0.9325± 0.0173	**0.8773± 0.0090**	0.9045± 0.0127
	WISDM	1.0456±0.0062	**1.0075± 0.0057**	1.0409± 0.0099	1.0268 ± 0.0106	1.0418 ± 0.0074
	PTB	**0.9746± 0.0203**	1.0420± 0.0561	0.9992± 0.0356	1.0035 ±0.0544	1.0787± 0.0255
	PAMAP2	0.9867±0.0054	0.9846± 0.0068	0.9896 ± 0.0056	1.0039± 0.0063	**0.9776± 0.0040**
	Average	1.0079±0.0056	**0.9690 ± 0.0115**	0.9871 ± 0.0115	0.9860± 0.0118	1.0061± 0.0058
RQ-VAE	HHAR	1.0658 ±0.0158	1.1557± 0.0113	1.0056 ± 0.0087	**0.9653± 0.0099**	1.0605 ± 0.0118
	UCIHAR	0.9127 ±0.0078	**0.8481±0.0061**	0.9376 ±0.0053	0.8946± 0.0066	0.9258± 0.0042
	WISDM	1.1099±0.0027	1.1428±0.0103	**0.9890 ± 0.0047**	1.1100 ± 0.0133	1.1288 ± 0.0050
	PTB	0.9376± 0.0334	**0.8766±0.0285**	0.9497 ± 0.0274	0.8974 ± 0.0204	0.8985± 0.0331
	PAMAP2	0.9834 ±0.0065	**0.9117± 0.0114**	0.9827 ± 0.0154	1.0677± 0.0180	1.0062±0.0101
	Average	1.0019 ±0.0062	0.9870± 0.0055	**0.9729 ± 0.0077**	0.9870± 0.0070	1.0040± 0.0077

5.1 Experimental Results

Tables 1 and 2 report generation performance in terms of gMSE and MDD, respectively. We primarily compare HierCode with direct fine-token autoregressive generation under the same pretrained tokenizer and FLAN-T5 backbone. Compared with direct fine-token autoregressive generation, HierCode reduces the average gMSE by 1.32%, 3.36%, and 6.63% for VAEVQ, VQ-VAE, and RQ-VAE, respectively, and achieves lower gMSE in 12 of the 15 tokenizer–dataset settings. A similar pattern is observed for MDD, with average reductions of 1.19%, 0.22%, and 0.41% for VAEVQ, VQ-VAE, and RQ-VAE, respectively. Overall, these results show that HierCode can improve downstream generation over direct fine-token autoregressive modeling across different vector-quantized representations without modifying the pretrained tokenizer.

5.2 Ablation Studies

We conduct ablation studies to examine two design choices in HierCode: The strategy for incorporating coarse predictions into fine-grained token recovery and the granularity of the cluster token space.

Effect of coarse-to-fine prediction strategies. Table 3 and Appendix Table 9 report gMSE and MDD, respectively, for different coarse-to-fine prediction strategies under the same FLAN-T5-Base backbone. *Hard Top-k* retains only the k most probable clusters and restricts fine-token prediction to candidates belonging to the selected clusters. *Soft Cluster Prior* retains the full predictive distribution over clusters and uses the probability assigned to each fine token's cluster to adjust its prediction score. *HierCode* further introduces cluster-token perturbation during fine-recovery training.

Replacing Hard Top-1 with Soft Cluster Prior reduces the average gMSE by 0.68%, 0.44%, and 0.66% for VAEVQ, VQ-VAE, and RQ-VAE, respectively. Although the improvements are modest, their consistent direction across the three tokenizers indicates that retaining coarse predictive uncertainty is preferable to restricting fine-token recovery to a single predicted cluster. Building on the soft prior, cluster-token perturbation further reduces the average gMSE by 2.16% and 2.64% for VAEVQ and VQ-VAE, respectively, with corresponding improvements in MDD. These results show that the soft cluster prior consistently benefits fine-token recovery, while training-time perturbation provides additional robustness for discrete representations.

Effect of cluster granularity. The number of clusters M controls the granularity of the coarse token space and introduces a trade-off between coarse-level prediction and fine-token recovery. A smaller M yields a more compact coarse vocabulary but groups a larger number of fine-grained codewords within each cluster, providing less specific guidance for fine-token recovery. Conversely, a larger M provides finer-grained cluster information but may increase the complexity of coarse-level autoregressive prediction. Table 4 and Appendix Table 9 report generation performance for different numbers of clusters $M \in \{4, 8, 16, 32, 64\}$ using FLAN-T5-Base. The preferred cluster granularity varies across tokenizers. VAEVQ achieves its lowest average gMSE at $M = 64$, whereas VQ-VAE and RQ-VAE achieve their lowest average gMSE at $M = 8$ and $M = 16$, respectively. Importantly, across all three tokenizers, even the least favorable value of M yields a lower average gMSE than direct fine-token autoregressive generation. This result shows that the improvement of HierCode is not confined to a single carefully tuned cluster granularity, although the optimal value of M remains tokenizer-dependent.

6 Conclusion

This work examined whether reconstruction-oriented evaluation is sufficient for selecting discrete representations for autoregressive wearable sensor generation. Our experiments show that reconstruction quality does not consistently reflect downstream generation quality. By further examining token errors produced during autoregressive generation, we find that predictions farther from their target codewords tend to produce larger data distortions after decoding. Based on this finding, we developed HierCode, which organizes nearby codewords into clusters and decomposes generation into autoregressive cluster generation followed by parallel fine-token recovery. Across five public datasets, HierCode improves over direct fine-token autoregressive generation in the majority of evaluated settings without modifying the pretrained tokenizer.

However, this study has several limitations. The preferred number of clusters varies across tokenizers, suggesting that the appropriate hierarchical organization may depend on the underlying discrete representation. In addition, our analysis considers a specific set of vector-quantized tokenizers and measures token deviation using codeword distance. Future work could explore adaptive hierarchies and investigate whether other properties of discrete representations are informative for autoregressive generation.

References

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pp. 3–4,

2013.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Ralf Bousseljot, Dieter Kreiseler, and Allard Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. 1995.

Zhicheng Chen, Shibo Feng, Zhong Zhang, Xi Xiao, Xingyu Gao, and Peilin Zhao. Sdformer: Similarity-driven discrete transformer for time series generation. *Advances in Neural Information Processing Systems*, 37:132179–132207, 2024.

Hyunseung Chung, Jiho Kim, Joon-Myoung Kwon, Ki-Hyun Jeon, Min Sung Lee, and Edward Choi. Text-to-ecg: 12-lead electrocardiogram synthesis conditioned on clinical text reports. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

A. Desai, C. Freeman, Z. Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*, 2021.

M. Goswami, K. Szafer, A. Choudhry, et al. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Wei-Bang Jiang, Yansen Wang, Bao-Liang Lu, and Dongsheng Li. Neurolm: A universal multi-task foundation model for bridging the gap between language and eeg signals. In *International Conference on Learning Representations*, volume 2025, pp. 55436–55457, 2025.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *International conference on learning representations*, volume 2024, pp. 23857–23880, 2024.

Justin Khasentino, Anastasiya Belyaeva, Xin Liu, Zhun Yang, Nicholas A Furlotte, Chace Lee, Erik Schenck, Yojan Patel, Jian Cui, Logan Douglas Schneider, et al. A personal health large language model for sleep and fitness coaching. *Nature Medicine*, 31(10):3394–3403, 2025.

Yubin Kim, Xuhai Xu, Daniel McDuff, Cynthia Breazeal, and Hae Won Park. Health-llm: Large language models for health prediction via wearable sensor data. In *Conference on Health, Inference, and Learning*, pp. 522–539. PMLR, 2024.

Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

Patrick Langer, Thomas Kaar, Max Rosenblattl, Maxwell A. Xu, Winnie Chow, Martin Maritsch, Robert Jakob, Ning Wang, Juncheng Liu, Aradhana Verma, et al. Opentslm: Time-series language models for reasoning over multivariate medical text- and time-series data. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, 2026.

Oscar D. Lara and Miguel A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2012.

D. Lee, S. Malacarne, and E. Aune. Vector quantized time series generation with a bidirectional prior model. In *International Conference on Artificial Intelligence and Statistics*, pp. 7665–7693. PMLR, 2023.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11523–11532, 2022.

Seokhyun Lee, Jaeho Kim, Changjun Oh, Mihaela van der Schaar, and Changhee Lee. Timetok: Granularity-controllable time-series generation via hierarchical tokenization. *arXiv preprint arXiv:2605.01418*, 2026.

Z. Li, S. Deldari, L. Chen, et al. Sensorllm: Aligning large language models with motion sensors for human activity recognition. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 354–379, 2025.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025.

J. Lu, X. Zheng, M. Sheng, et al. Efficient human activity recognition using a single wearable sensor. *IEEE Internet of Things Journal*, 7(11):11137–11146, 2020.

Mike A Merrill, Akshay Paruchuri, Naghmeh Rezaei, Geza Kovacs, Javier Perez, Yun Liu, Erik Schenck, Nova Hammerquist, Jake Sunshine, Shyam Tailor, et al. Transforming wearable data into personal health insights using large language model agents. *Nature Communications*, 2026.

S. Miao, L. Chen, R. Hu, et al. Towards a dynamic inter-sensor correlations learning framework for multi-sensor-based wearable human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–25, 2022.

S. Miao, L. Chen, and R. Hu. Spatial-temporal masked autoencoder for multi-device wearable human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(4):1–25, 2024.

Ilan Naiman, N Benjamin Erichson, Pu Ren, Michael W Mahoney, and Omri Azencot. Generative modeling of regular and irregular time series data via koopman vaes. In *International conference on learning representations*, volume 2024, pp. 50547–50573, 2024.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pp. 1864–1874, 2022.

Jeha Paul, Bohlke-Schneider Michael, Mercado Pedro, Kapoor Shubham, Singh Nirwan Rajbir, Flunkert Valentin, Gasthaus Jan, and Januschowski Tim. Psa-gan: Progressive self attention gans for synthetic time series. *arXiv preprint arXiv:2108.00981*, 2021.

Hengzhi Pei, Kan Ren, Yuqing Yang, Chang Liu, Tao Qin, and Dongsheng Li. Towards generating real-world time series data. In *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 469–478. IEEE, 2021.

Lala Shakti Swarup Ray, Lars Krupp, Vitor Fortes Rey, Bo Zhou, Sungho Suh, and Paul Lukowicz. Txp: Reciprocal generation of ground pressure dynamics and activity descriptions for improving human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(2):1–32, 2025.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*, pp. 108–109. IEEE, 2012.

Cécile Rousseau, Tobia Boschi, Giandomenico Cornacchia, Dhaval Salwala, Alessandra Pascale, and Juan Moreno. Forging time series with language: A large language model approach to synthetic data generation. *Advances in neural information processing systems*, 38:78402–78432, 2026.

Feng Shibo, Peilin Zhao, Liu Liu, Pengcheng Wu, and Zhiqi Shen. Hdt: Hierarchical discrete transformer for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 746–754, 2025.

Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pp. 127–140, 2015.

Sabera Talukder, Yisong Yue, and Georgia Gkioxari. Totem: Tokenized time series embeddings for general time series analysis. *arXiv preprint arXiv:2402.16412*, 2024.

Y. Tang, L. Zhang, F. Min, et al. Multiscale deep feature learning for human activity recognition using wearable sensors. *IEEE Transactions on Industrial Electronics*, 70(2):2106–2116, 2022.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yunshi Wen, Tengfei Ma, Lily Weng, Lam Nguyen, and Anak Agung Julius. Abstracted shapes as tokens-a generalizable and interpretable model for time-series classification. *Advances in Neural Information Processing Systems*, 37:92246–92272, 2024.

Z. Xiao, H. Xing, R. Qu, et al. Heterogeneous mutual knowledge distillation for wearable human activity recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

Xiangyu Xu, Qingsong Zhong, and Jilin Hu. Timemar: Multi-scale autoregressive modeling for unconditional time series generation. In *Proceedings of the ACM Web Conference 2026*, pp. 5132–5143, 2026.

Hao Xue and Flora D Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6851–6864, 2023.

S. Yang, X. Hu, Q. Wu, et al. Vaevq: Enhancing discrete visual tokenization through variational modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11703–11711, 2026.

Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

Xiyuan Zhang, Diyan Teng, Ranak R Chowdhury, Shuheng Li, Dezhi Hong, Rajesh K Gupta, and Jingbo Shang. Unimts: Unified pre-training for motion time series. *Advances in Neural Information Processing Systems*, 37:107469–107493, 2024.

Y. Zhang, K. Ayush, S. Qiao, et al. Sensorlm: Learning the language of wearable sensors. *Advances in Neural Information Processing Systems*, 38:46813–46849, 2026.

A Related Work

This section reviews from wearable and time-series generation, foundation models for time series, and discrete tokenization for time-series modeling.

A.1 Time-series Generation

Early deep generative approaches for time-series synthesis primarily relied on generative adversarial networks (GANs) (Paul et al., 2021; Pei et al., 2021). For example, TimeGAN (Yoon et al., 2019) combines adversarial learning with a supervised temporal objective to preserve the dynamics of real time series. RTSGAN (Pei et al., 2021) further integrates an encoder–decoder architecture with wasserstein adversarial learning for multivariate time-series generation.

Variational autoencoders (VAEs) provide an alternative by explicitly modeling time series in a probabilistic latent space. TimeVAE (Desai et al., 2021) adapts the VAE framework to multivariate time-series generation and introduces interpretable components that can capture temporal structures such as trends and seasonal patterns. KoVAE (Naiman et al., 2024) further incorporates Koopman-inspired dynamics into the latent prior, representing the latent temporal evolution through a linear operator and supporting both regularly and irregularly sampled time series.

A.2 Foundation Models for Time-Series

Large-scale pretraining has recently shifted time-series modeling from task-specific architectures toward foundation models that can transfer across downstream tasks. One line of research develops

general-purpose time-series foundation models. MOMENT (Goswami et al., 2024) learns transferable representations through large-scale pretraining over heterogeneous time-series data. Time-LLM (Jin et al., 2024) instead reprograms numerical time-series inputs into representations compatible with pretrained language models for forecasting. More recent models, including Moirai2 (Liu et al., 2025) and Chronos-2 (Ansari et al., 2025), further explore scalable pretrained sequence models for forecasting across diverse time-series domains and task settings.

A complementary direction focuses on connecting wearable sensor data with text representations. UniMTS (Zhang et al., 2024) aligns multivariate sensor data with textual semantics through contrastive learning. SensorLLM (Li et al., 2025) extends pretrained language models to align wearable sensor data with language descriptions for activity understanding and reasoning. More recently, SensorLM (Zhang et al., 2026) develops a unified sensor–language foundation model supporting multiple tasks, including recognition, retrieval, and generation.

Together, these studies demonstrate the growing role of large-scale sequence models in time-series and wearable sensing and illustrate diverse strategies for interfacing continuous temporal signals with sequence models. This has motivated increasing interest in discrete tokenization as an explicit interface between continuous sensor data and sequence models.

A.3 Discrete Tokenization for Time-Series Modeling

Vector quantization provides a widely adopted approach for representing time series as discrete sequences. VQ-VAE (van den Oord et al., 2017) established the standard single-codebook formulation, while subsequent variants have explored alternative ways of organizing and learning the discrete latent space. For example, VAEVQ (Yang et al., 2026) introduces variational modeling into vector quantization to obtain a smoother latent space and improve codebook utilization. Subsequent time-series studies have extended this paradigm by introducing different representation designs. TimeVQ-VAE (Lee et al., 2023) separately quantizes low- and high-frequency components to capture complementary temporal patterns, while TOTEM (Talukder et al., 2024) learns reusable vector-quantized representations across multiple time-series tasks. VQShape (Wen et al., 2024) constructs discrete representations around recurring temporal shapes to improve generalization. Discrete tokens have also been adopted as the modeling units of downstream generative and foundation models. Auto-TTE (Chung et al., 2023) employs discrete representations for text-conditioned physiological signal generation. SDformer (Chen et al., 2024) performs generation over discretized time-series representations, and NeuroLM (Jiang et al., 2025) integrates discrete neural-signal tokens into a multi-task foundation model.

Discrete representations have also increasingly been structured at multiple levels of granularity. Residual quantization, as introduced by RQ-VAE (Lee et al., 2022), represents each latent through a sequence of residual codewords and has been extended in models such as TXP (Ray et al., 2025). Other approaches construct hierarchy along the temporal dimension. HDT (Shibo et al., 2025) performs hierarchical generation by first modeling a downsampled target representation and then using it to guide generation at the original temporal resolution. TimeMAR (Xu et al., 2026) encodes time series into discrete representations at multiple temporal resolutions and performs autoregressive generation, while TimeTok (Lee et al., 2026) similarly organize discrete representations across multiple temporal resolutions to support generation at controllable temporal granularities.

B Experimental Setup

B.1 Analysis Experiments

We evaluate on five public wearable and physiological time-series datasets. Dataset statistics are summarized in Table 5, and implementation details are provided in Tables 6 and 7.

Tokenizer architecture and training objective. All tokenizers use the same convolutional encoder–decoder backbone to ensure a controlled comparison. The encoder contains two temporal downsampling stages with stride 2, resulting in an overall temporal downsampling factor of 4. The decoder symmetrically restores the original temporal resolution.

Table 5: Statistics of the public benchmark datasets.

Dataset	Domain	Sensors	Classes	Subjects	Sampling Rate	Window Length
HHAR	HAR	Acc + Gyro	6	9	100 Hz	128
UCI-HAR	HAR	Acc + Gyro	6	30	50 Hz	128
WISDM	HAR	Acc + Gyro	18	51	20 Hz	128
PAMAP2	HAR	IMU + HR	18	9	100 Hz	128
PTB	ECG	12-lead ECG	2	4	500 Hz	300

HAR = human activity recognition; Acc = accelerometer; Gyro = gyroscope; IMU = inertial measurement unit; HR = heart rate; ECG = electrocardiogram.

Table 6: Architecture and training hyperparameters of the wearable tokenizers.

Parameter	Setting
Architecture	
Base channel width	64
Channel multipliers	$(1, 2, 2)$
Residual blocks	2 per resolution level
Latent channels	128
Embedding dimension	64
Quantization	
Codebook size K	$\{64, 128, 256, 512, 1024\}$
Default codebook size	512
Commitment coefficient β	0.25
RQ-VAE levels	2
Training	
Normalization	Per-channel z-score
Reconstruction loss	ℓ_1
Optimizer	Adam
Adam coefficients	$\beta_1 = 0.5,\ \beta_2 = 0.9$
Learning rate	1×10^{-4}
Batch size	64
Maximum epochs	500

The three tokenizers differ only in their quantization mechanisms. VAEVQ uses a learned codebook with a learnable projection for nearest-neighbor quantization. VQ-VAE employs a standard single-level vector quantizer with a codebook of size K, whereas RQ-VAE uses two sequential residual quantization levels, each with a codebook of size K.

All tokenizers are trained using an L1 reconstruction objective together with the corresponding vector-quantization objective. The commitment weight is set to $\beta = 0.25$. For RQ-VAE, the quantization objective is averaged over the two residual levels.

B.2 HierCode Experiments

Evaluation Metrics. We evaluate generation quality using generation mean squared error (gMSE) and Marginal Distribution Difference (MDD), both computed on the decoded generated suffix. gMSE measures the pointwise discrepancy between the generated and ground-truth signals,

$$\mathrm{gMSE} = \frac{1}{|\mathcal{T}||\mathcal{C}|} \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}} \left(X_{t,c} - \widehat{X}_{t,c} \right)^2, \tag{7}$$

where $\mathcal{T}$ and $\mathcal{C}$ denote the evaluated suffix positions and sensor channels, respectively.

MDD measures the discrepancy between the marginal distributions of real and generated sensor values independently at each time position and channel. For each (t, c), we construct normalized

Table 7: Training configuration for vanilla autoregressive sensor-token generation. The sensor tokenizer is frozen during autoregressive model training.

Parameter	**Setting**
Backbone	FLAN-T5-base
Codebook size	$K = 512$
Generation protocol	50% observed prefix, 50% generated suffix
Maximum training samples	5,000
Optimizer	AdamW
Learning rate	2×10^{-4}
Learning-rate schedule	Linear decay with 5% warm-up
Batch size	16
Maximum epochs	1,000
Gradient clipping	Maximum norm 1.0
Decoding strategy	Sampling ($T = 1.0$)
Independent runs	10

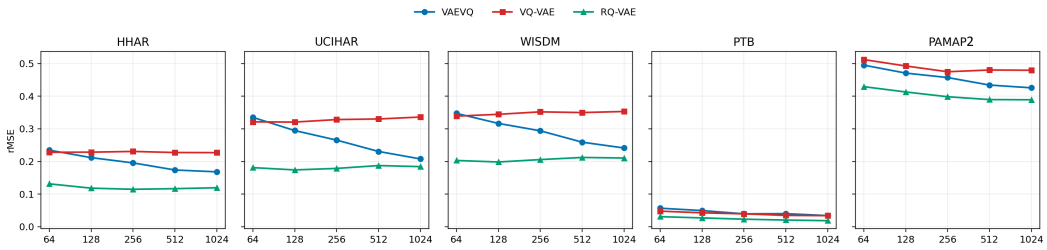

Figure 4: Effect of codebook size $K \in \{64, 128, 256, 512, 1024\}$ on reconstruction performance (rMSE) across tokenizers and datasets.

histograms over the evaluation samples using $B = 50$ bins and compute

$$\mathrm{MDD}_{t,c} = \frac{1}{B} \sum_{b=1}^{B} \left| p_{\mathrm{real}}(b \mid t, c) - p_{\mathrm{gen}}(b \mid t, c) \right|, \tag{8}$$

and average over the evaluated positions and channels,

$$\mathrm{MDD} = \frac{1}{|\mathcal{T}||\mathcal{C}|} \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}} \mathrm{MDD}_{t,c}. \tag{9}$$

Lower gMSE and MDD indicate better generation performance.

Baselines. *Direct autoregressive baselines.* Our primary comparison is direct autoregressive generation using VQ-VAE, RQ-VAE, and VAEVQ as discrete tokenizers. For each tokenizer, the corresponding baseline and HierCode use the same pretrained tokenizer and FLAN-T5 backbone. Detailed hyperparameter settings are provided in Table 8.

External generation baselines. We additionally compare with representative time-series generation methods, including SDformer (Chen et al., 2024), TimeVAE (Desai et al., 2021), and RTSGAN (Pei et al., 2021), as well as recent pretrained forecasting models Moirai2 (Liu et al., 2025) and Chronos-2 (Ansari et al., 2025). Because these methods employ different representation and generation mechanisms, they serve as broader reference points for overall generation quality rather than controlled ablations of HierCode.

C Experimental Results

Table 8: Training configuration for HierCode.

Parameter	Setting
Fine codebook size	$K = 512$
Number of clusters	$M = 16$
Codebook clustering	k-means++
Coarse backbone	FLAN-T5-base
Fine backbone	T5-based refinement model
Coarse training epochs	Maximum 1,000
Fine training epochs	Maximum 2,000
Coarse-token pertub probability	$p = 0.3$
Fine conditioning	Soft coarse distribution
Soft-conditioning top-k	16
Soft-conditioning strength	$\beta = 1.0$
Numerical stability	$\epsilon = 10^{-4}$
Optimizer	AdamW
Learning rate	2×10^{-4}
Learning-rate schedule	Linear decay with 5% warm-up
Training batch size	16
Gradient clipping	Maximum norm 1.0
Decoding strategy	Sampling ($T = 1.0$)
Independent runs	10

Table 9: Comparison of different fine-grained token recovery strategies across three tokenizers and five benchmark datasets using MDD.

Tokenizer	Dataset	Hard Top-1	Hard Top-3	Soft Posterior	HierCode
VAEVQ	HHAR	0.1224 ± 0.0251	0.0800 ± 0.0082	0.0657 ± 0.0013	$\mathbf{0.0629 \pm 0.0013}$
	UCI-HAR	0.1272 ± 0.0206	0.1019 ± 0.0073	0.0975 ± 0.0034	$\mathbf{0.0941 \pm 0.0024}$
	WISDM	0.1368 ± 0.0059	0.1270 ± 0.0019	0.1216 ± 0.0017	$\mathbf{0.1196 \pm 0.0006}$
	PTB	0.0538 ± 0.0041	0.0539 ± 0.0043	$\mathbf{0.0530 \pm 0.0041}$	0.0550 ± 0.0034
	PAMAP2	0.1979 ± 0.0027	0.1961 ± 0.0026	0.1962 ± 0.0025	$\mathbf{0.1929 \pm 0.0017}$
	Average	0.1276 ± 0.0067	0.1118 ± 0.0022	0.1068 ± 0.0013	$\mathbf{0.1049 \pm 0.0006}$
VQ-VAE	HHAR	0.1081 ± 0.0304	0.0742 ± 0.0079	0.0886 ± 0.0559	$\mathbf{0.0697 \pm 0.0022}$
	UCI-HAR	0.1281 ± 0.0296	0.1049 ± 0.0081	0.1032 ± 0.0057	$\mathbf{0.1001 \pm 0.0049}$
	WISDM	0.1459 ± 0.0162	0.1280 ± 0.0060	0.1255 ± 0.0045	$\mathbf{0.1235 \pm 0.0037}$
	PTB	0.0565 ± 0.0037	0.0558 ± 0.0033	$\mathbf{0.0548 \pm 0.0032}$	0.0564 ± 0.0034
	PAMAP2	0.1963 ± 0.0021	0.1955 ± 0.0032	0.1931 ± 0.0026	$\mathbf{0.1926 \pm 0.0026}$
	Average	0.1270 ± 0.0077	0.1117 ± 0.0030	0.1131 ± 0.0128	$\mathbf{0.1085 \pm 0.0012}$
RQ-VAE	HHAR	0.0670 ± 0.0028	0.0643 ± 0.0023	0.0659 ± 0.0104	$\mathbf{0.0628 \pm 0.0008}$
	UCI-HAR	0.0979 ± 0.0020	0.0930 ± 0.0017	0.0927 ± 0.0016	$\mathbf{0.0856 \pm 0.0015}$
	WISDM	0.1173 ± 0.0015	0.1148 ± 0.0014	0.1143 ± 0.0012	$\mathbf{0.1025 \pm 0.0011}$
	PTB	0.0520 ± 0.0038	0.0511 ± 0.0026	$\mathbf{0.0516 \pm 0.0029}$	0.0516 ± 0.0029
	PAMAP2	0.1951 ± 0.0060	0.1950 ± 0.0060	0.1924 ± 0.0048	$\mathbf{0.1912 \pm 0.0055}$
	Average	0.1059 ± 0.0014	0.1036 ± 0.0014	0.1031 ± 0.0025	$\mathbf{0.1021 \pm 0.0012}$

Table 10: Effect of the number of clusters M on generation performance (MDD) across different tokenizers and datasets.

Tokenizer	Dataset	$\mathbf{M = 4}$	$\mathbf{M = 8}$	$\mathbf{M = 16}$	$\mathbf{M = 32}$	$\mathbf{M = 64}$
VAEVQ	HHAR	0.0640 ± 0.0008	0.0633 ± 0.0010	**0.0629 ± 0.0013**	0.0636 ± 0.0009	0.0640 ± 0.0006
	UCI-HAR	0.0950 ± 0.0007	**0.0933 ± 0.0024**	0.0941 ± 0.0024	0.0958 ± 0.0021	0.0955 ± 0.0023
	WISDM	0.1187 ± 0.0009	**0.1184 ± 0.0011**	0.1196 ± 0.0006	0.1199 ± 0.0013	0.1206 ± 0.0008
	PTB	0.0592 ± 0.0024	0.0572 ± 0.0035	**0.0550 ± 0.0034**	0.0564 ± 0.0033	0.0555 ± 0.0033
	PAMAP2	**0.1873 ± 0.0040**	0.1922 ± 0.0035	0.1929 ± 0.0017	0.1949 ± 0.0022	0.1977 ± 0.0018
	Average	**0.1048 ± 0.0009**	0.1049 ± 0.0011	0.1049 ± 0.0006	0.1061 ± 0.0010	0.1067 ± 0.0009
VQ-VAE	HHAR	0.0702 ± 0.0023	**0.0688 ± 0.0020**	0.0697 ± 0.0022	0.0697 ± 0.0022	0.0707 ± 0.0020
	UCI-HAR	0.1027 ± 0.0047	0.1016 ± 0.0039	0.1001 ± 0.0049	**0.0993 ± 0.0032**	0.1008 ± 0.0044
	WISDM	**0.1225 ± 0.0011**	0.1233 ± 0.0016	0.1235 ± 0.0037	0.1236 ± 0.0045	0.1280 ± 0.0028
	PTB	0.0604 ± 0.0033	0.0568 ± 0.0062	0.0564 ± 0.0034	0.0566 ± 0.0028	**0.0523 ± 0.0024**
	PAMAP2	0.1929 ± 0.0031	**0.1915 ± 0.0033**	0.1926 ± 0.0026	0.1925 ± 0.0023	0.1928 ± 0.0020
	Average	0.1097 ± 0.0016	0.1084 ± 0.0014	0.1085 ± 0.0012	**0.1083 ± 0.0011**	0.1089 ± 0.0015
RQ-VAE	HHAR	0.0640 ± 0.0005	0.0628 ± 0.0007	0.0628 ± 0.0008	**0.0627 ± 0.0006**	0.0631 ± 0.0012
	UCI-HAR	0.0924 ± 0.0025	0.0928 ± 0.0014	**0.0856 ± 0.0015**	0.0916 ± 0.0019	0.0917 ± 0.0013
	WISDM	0.1151 ± 0.0007	0.1156 ± 0.0040	**0.1025 ± 0.0011**	0.1166 ± 0.0086	0.1134 ± 0.0013
	PTB	0.0592 ± 0.0034	0.0544 ± 0.0032	0.0516 ± 0.0029	0.0520 ± 0.0028	**0.0501 ± 0.0029**
	PAMAP2	0.1948 ± 0.0026	0.1917 ± 0.0038	0.1912 ± 0.0055	0.1902 ± 0.0058	**0.1896 ± 0.0039**
	Average	0.1051 ± 0.0011	0.1035 ± 0.0016	**0.0987 ± 0.0012**	0.1026 ± 0.0021	0.1016 ± 0.0012