CONTINUOUS FIRST, DISCRETE LATER: VQ-VAES WITHOUT DIMENSIONAL COLLAPSE

Anonymous authors
Paper under double-blind review

ABSTRACT

While many approaches to improve VQ-VAE performance focus on codebook size and utilization, the effect of dimensional collapse, where trained VQ-VAE representations live in an extremely low-dimensional subspace ($1-2\%$ of full rank), remains poorly understood. We show theoretically and empirically that dimension collapse causes a hard loss lower bound that various codebook improvement techniques fail to surpass. Our analytic framework extends the sequential learning effect of Saxe et al. (2014) by introducing ideas from rate-distortion theory and explains how the latent collapse is caused by the VQ suppressing lower-variance directions. Our theory justifies a simple solution: a "warm-up phase" that trains the model as an (unquantized) autoencoder before introducing VQ. On both synthetic experiments and large-scale image (VQGAN) and audio (WavTokenizer) VQ-VAEs, we show that AE Warm-Up successfully restores representation dimension, leading to lower reconstruction and perceptual loss at the same training budget. Across codebook sizes $K \in \{2^{10}, 2^{14}, 2^{16}\}$, AE warm-up raises VQGAN codebook effective dimension from 3–5 to 17–19 and reduces rFID by 17–35%; on WavTokenizer at $K \in \{2^{13}, 2^{14}\}$, it raises codebook dimension from 4 to 17–19 and improves PESQ by 11–14%. We empirically characterize how warm-up duration governs the achievable final loss. In agreement with experiment, our theoretical analysis predicts the latent dimension as a function of warm-up length, enabling an adaptive criterion for switching from AE Warm-up to VQ-VAE training. [1]

1 INTRODUCTION

Vector-quantized variational autoencoders (VQ-VAEs) are the backbone of modern discrete tokenization for audio and image representation learning and generative models (van den Oord et al., 2017; Razavi et al., 2019; Esser et al., 2021; Zeghidour et al., 2021; Ji et al., 2025). Scaling the codebook is the standard lever for improving VQ-VAE reconstruction, and a sustained line of work has improved codebook utilization through techniques such as EMA updates, dead-code respawn, and codebook reparameterization (Lancucki et al., 2020; Zheng & Vedaldi, 2023; Zhu et al., 2024; 2025). It is thus quite surprising that large-scale VQ-VAEs use only a small fraction of allocated latent dimensions: we find that WavTokenizer on LibriTTS, with a 512-dimensional pre-quantization latent, uses only 4 effective dimensions at codebook size $K = 2^{12}$ despite 100% codebook utilization, or as Zhang et al. (2025) shows, VQGAN on ImageNet collapses to 2–5 effective dimensions across codebook sizes from $K = 2^{10}$ to $K = 2^{14}$. This dimensional collapse creates an irreducible loss floor that does not improve with codebook scaling. The bottleneck is not the codebook size, but the effective dimension of the latent subspace.

To theoretically analyze the mechanism behind collapse, we build on the seminal work of Saxe et al. (2014); Refinetti & Goldt (2022), which demonstrated how plain autoencoders learn the principal modes of their data sequentially, in decreasing order of variance. Unfortunately, extending their idea to VQ-VAE is difficult, since it requires analyzing coupled dynamics of the K codebook vectors and encoder-decoder matrices, complicated by the non-differentiable nearest-neighbor quantizer. Our contribution is the introduction of a tractable model, *RD-AE flow*, which replaces VQ with a stochastic noising operation, chosen to minimize mean-squared error subject to mutual information upper bound of $\log_2 K$. We show that under the optimal coupling, latent directions corresponding

[1] Our code is released at https://anonymous.4open.science/r/vqvae_latent_span_collapse-51BB.

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

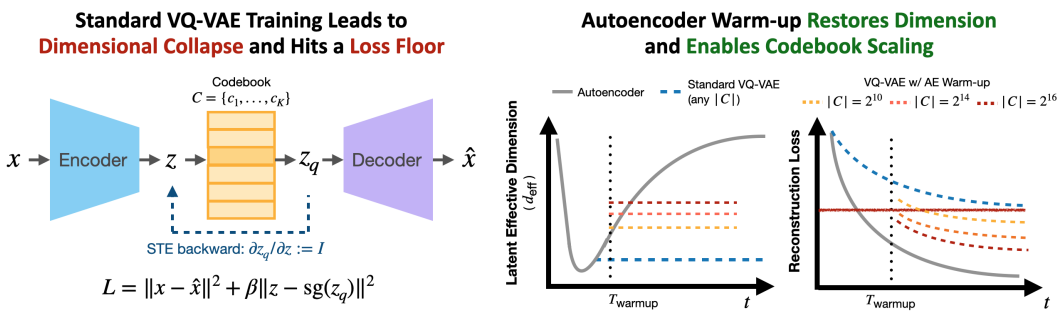


Figure 1: **Dimensional collapse in VQ-VAEs and the warm-up fix.** During plain autoencoder training (gray), the latent effective dimension d_{eff} grows as directions of data variance ("modes") are learned sequentially. Turning on the quantizer too early collapses d_{eff}; reconstruction loss hits a floor that is independent of codebook size $|C|$ (blue). Training the encoder–decoder as a plain autoencoder before introducing the quantizer (AE Warm-up) allows the encoder to learn lower-variance directions, preserving higher d_{eff} after introducing VQ and restoring the expected scaling of reconstruction quality with $|C|$ (yellow-red). We give an information-theoretic account of this mechanism (Section 3) and validate it in both image (VQGAN trained with ImageNet-100, Section 4.1) and audio (WavTokenizer trained with LibriTTS, Section 4.2) VQ-VAEs.

to low-variance modes may fall below a so-called *water level* and stop evolving due to interaction between the commitment loss term and the STE, see Fig. 2, thereby collapsing the latent dimension.

This model shows that the cause of dimensional collapse is the timing of quantization: latent directions that were not yet properly learned when the quantizer turns on are deactivated by the rate constraint and never recover. Our proposed fix trains the encoder and decoder as a plain autoencoder until enough directions are represented by the encoder before introducing VQ; we call this *AE warm-up*. Our theory predicts dependence of the latent effective dimension on the length of the warm-up, and is corroborated by experiments on VQGAN, see Figure 5. AE Warm-up raises the VQGAN codebook dimension from 5 to 18 at $K = 2^{14}$ and the WavTokenizer codebook dimension from 4 to 19 at $K = 2^{14}$, breaking the loss floor that cold-start training hits regardless of codebook size and restoring expected scaling with K. Increasing warm-up steps up to the predicted stopping point both increases final codebook dimension and reduces final loss, with gains scaling with target codebook size.

Contributions.

- (practice) We demonstrate dimensionality collapse in popular VQ-VAEs for audio and image tokenization (WavTokenizer, VQGAN) and show that the resulting performance degradation cannot be fixed by increasing codebook size K ("loss floor").
- (practice) We propose a simple fix, *AE Warm-up*, in which the bottleneck between encoder and decoder remains unquantized during the initial training phase.
- (theory) We introduce *RD-AE*, a toy model for studying VQ-VAE training dynamics that extends the well-known sequential learning analysis of Saxe et al. (2014) to the VQ-VAE setting.
- (theory+practice) We derive a bound on the latent dimensionality and reconstruction loss as a function of warm-up duration. This allows us to adaptively detect the optimal warm-up duration.
- (practice) Our approach improves VQGAN (ImageNet-1K) and WavTokenizer (LibriTTS), where AE Warm-up raises codebook effective dimension by 4-5$\times$ and restores loss scaling with K.

Prior and concurrent work. The warm-up approach has been previously proposed in various ways, although without similar justification or connection to latent dimension. Lancucki et al. (2020) delay quantization as a stability measure within their codebook-utilization method, Zhao et al. (2024) report empirical gains from AE pretraining followed by VQ fine-tuning, Wang et al. (2025) take this to its limit by applying post-hoc quantization to a frozen pretrained continuous VAE without any VQ training, and (concurrently with this work) Zhao et al. (2026) propose the warm-up under the name *deferred quantization*, though not connecting it to the latent dimension collapse. Independently and concurrently with our work, Zhang et al. (2026) observe empirically that quantization suppresses information-carrying low-variance directions, and propose a codebook regularlization loss term to fix

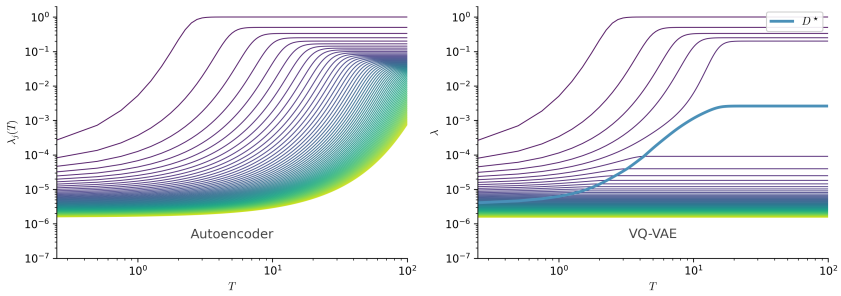


Figure 2: **Sequential learning in AE and VQ-VAE.** A 2-layer linear AE learns latent modes sequentially Saxe et al. (2014) (left). In VQ-VAEs, however, quantizing the bottleneck freezes lower modes (those under the water level D^*), thus severely constraining effective dimension (right). See Section 3 for details.

dimension collapse and improve reconstruction. Their results corroborate the premise of this paper. Our contribution is complementary: we explain *why* these directions are lost, through the water-level dynamics of Section 3, and prevent it by changing only the training schedule, with no additional loss term that requires tuning.

2 Background

Autoencoders and vector quantization. An autoencoder pairs an encoder $f_\theta : \mathcal{X} \to \mathbb{R}^k$ with a decoder $g_\phi : \mathbb{R}^k \to \mathcal{X}$ and minimizes the reconstruction loss $\|x - g_\phi(f_\theta(x))\|^2$. A vector-quantized autoencoder (van den Oord et al., 2017) inserts a discretization step between the two: a finite codebook $\mathcal{C} = \{c_1, \ldots, c_K\} \subset \mathbb{R}^k$ and a nearest-neighbor quantizer $Q(z) = \arg\min_{c \in \mathcal{C}} \|z - c\|$. The encoder produces a continuous pre-quantization latent $z = f_\theta(x)$; the quantizer replaces it with the nearest codeword $z_q = Q(z)$; the decoder reconstructs $\hat{x} = g_\phi(z_q)$. Training minimizes

$$\mathcal{L}(\phi, \theta) \;=\; \mathcal{L}_{\text{rec}} \;+\; \mathcal{L}_{\text{com}} \;=\; \mathbb{E}[\|x - \hat{x}\|^2 \;+\; \beta\, \|z - \text{sg}(z_q)\|^2], \tag{1}$$

where β is the commitment weight and sg is the stop-gradient and expectation is over the batch (practice) or population (theory). The encoder receives its gradient through the straight-through estimator (van den Oord et al., 2017), $\partial z_q / \partial z := I$. Codebook entries are typically updated by exponential moving averages of assigned latents.

Evaluation metrics. VQ-VAEs are evaluated on reconstruction quality and on the health of the learned codebook. Reconstruction is measured directly via pixel- or sample-level losses (L1, L2, mel-spectrogram), and perceptually via task-specific metrics — LPIPS (Zhang et al., 2018) and rFID for images, PESQ (Rix et al., 2001) and STOI (Taal et al., 2010) for audio. Modern VQ-VAEs additionally include an adversarial discriminator loss (Esser et al., 2021; Ji et al., 2025). On the codebook side, *utilization* measures how many allocated codewords are actually in use.

Even at full utilization, standard VQ-VAE representations live in a low-dimensional subspace (typically 4–10 dimensions), as measured by d_{eff}, the number of principal components needed to explain 99% of the pre-quantization latent variance:

$$d_{\text{eff}} = \min\left\{m : \textstyle\sum_{i \le m} \lambda_i / \sum_i \lambda_i \ge 0.99\right\}, \tag{2}$$

with $\{\lambda_i\}$ as the PCA eigenvalues of z in decreasing order. The same definition applies to the codebook entries themselves; we report both $d_{\text{eff}}(z)$ and $d_{\text{eff}}(\mathcal{C})$ in Section 4.

Autoencoder gradient flow. Saxe et al. (2014) analyzed the gradient flow of the plain-AE objective $L_{\text{AE}}(W_1, W_2) = \mathbb{E}\|x - W_2 W_1 x\|^2$ for the two-layer linear network $\hat{X} = W_2 W_1 X$. On the diagonal manifold $W_1 = \text{diag}(u_j)$, $W_2 = \text{diag}(v_j)$, the loss decouples as $L_{\text{AE}}(u, v) = \sum_j \sigma_j^2 (1 - u_j v_j)^2$, and the flow acts independently on each mode:

$$\dot{u}_j = 2\sigma_j^2\, v_j\, (1 - u_j v_j), \qquad \dot{v}_j = 2\sigma_j^2\, u_j\, (1 - u_j v_j). \tag{3}$$

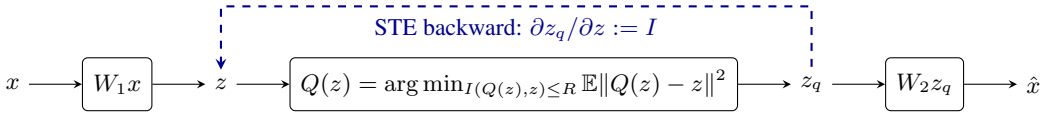


Figure 3: **The linear RD-AE.** The linear encoder W_1 maps data to the pre-quantization latent z; the (potentially stochastic) quantizer Q assigns z to code z_q; the linear decoder W_2 reconstructs $\hat{x}$. The backward pass through the quantizer uses the straight-through estimator.

With balanced initialization $u_j(0) = v_j(0) = \sqrt{s}$ at small scale $s \in (0, 1)$, the activation for each mode $r_j(t) := u_j(t)v_j(t)$ has the closed form

$$r_j(t) = \left(1 + \tfrac{1-s}{s}\, e^{-4\sigma_j^2 t}\right)^{-1}, \tag{4}$$

so modes activate sequentially in decreasing order of variance (Figure 2, left). Consequently d_{eff} dips as top modes saturate first, then recovers as the rest catch up (Figure 4, black) — a pattern also visible in large-scale VQGAN and WavTokenizer AEs trained with Adam (Figure 5 and Figure 7).

3 Theoretical Model of Collapse

We propose to model VQ-VAE training dynamics by replacing the nearest-neighbor quantizer with the rate-distortion-optimal continuous channel at rate $R = \log_2 K$, resulting in a model that we call *RD-AE*. This relaxation makes the joint dynamics of the encoder, decoder, and channel analytically tractable. We study the classical linear-Gaussian setting where AE training dynamics admit closed-form analysis (Saxe et al., 2014). Despite the simplifications, the resulting toy model exhibits dimensional collapse (Figure 4). Rigorous analysis predicts that the collapse results from the interplay of two mechanisms: sequential learning in autoencoders (Saxe et al., 2014; Refinetti & Goldt, 2022) and a fundamental property of MSE-trained vector-quantizers to collapse low-variance dimensions (which we model here information-theoretically via rate-distortion bottleneck).

This section is organized as follows. RD-AE and the simulation of the dense case is introduced in Section 3.1. The dense case is still difficult to analyze, and thus in Section 3.2 we further simplify to the case of diagonal covariance matrices and weight initialization, and present the collapse mechanism in detail. Section 3.3 explains how to predict resulting VQ-VAE effective dimension from the state of the AE Warm-up checkpoint used. Section 4 validates that the behavior of non-linear VQ-VAEs trained on real data is in qualitative agreement with the behavior of our toy model.

3.1 Setup

The data is $x \sim \mathcal{N}(0, \Sigma)$ on $\mathbb{R}^d$ with $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$ and $\sigma_1^2 \geq \cdots \geq \sigma_d^2 > 0$. A linear encoder $W_1 \in \mathbb{R}^{d\times d}$ produces the pre-quantization latent $z = W_1 x$; a (potentially stochastic) quantizer Q produces $z_q := Q(z)$; a linear decoder $W_2 \in \mathbb{R}^{d\times d}$ returns $\hat{x} = W_2 z_q$ (Figure 3). Training objective is the standard VQ-VAE loss (1). The learning dynamics follow standard VQ-VAE conventions: the encoder reconstruction "gradient" uses the straight-through estimator $\partial z_q/\partial z := I$, and the commitment term treats z_q as constant via stop-gradient. As a result, the update equations for W_1 and W_2 are not the gradient of (1).

Information-theoretic relaxation. Analyzing VQ's effect on the training dynamics requires jointly tracking K high-dimensional codewords alongside the encoder and decoder. The codebook also breaks the diagonal structure exploited by Saxe et al. (2014), preventing a per-mode decomposition. Our key insight is that we can replace the dynamically evolving VQ with its information-theoretic optimal counterpart. Specifically, we assume that the map $z \mapsto z_q$ at every instant of training is the one that minimizes MSE subject to the information bottleneck constraint:

$$\min_{P_{z_q|z}} \left\{\mathbb{E}[\|z - z_q\|^2] : I(z; z_q) \leq \log_2 K\right\} \triangleq D(R), \tag{5}$$

where we denote the minimum MSE as the optimal rate-R distortion in *latent space*, $D(R)$. This can be understood as relaxing the hard constraint $|\text{supp} z_q| \leq K$ to a softer mutual information constraint. Although we optimistically assume that quantizer is trained fast enough to remain optimal for the

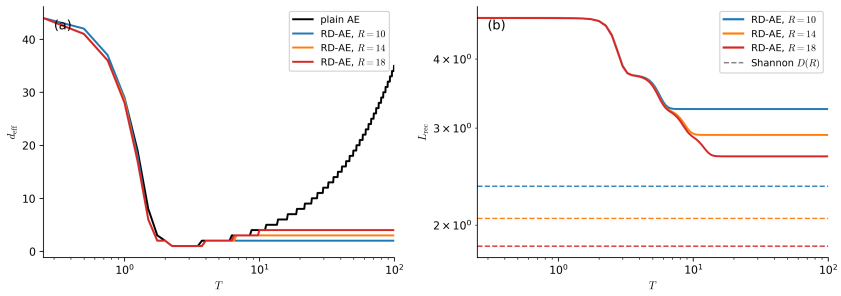


Figure 4: **Plain AE vs. RD-AE flow.** $d = 64$, $\Sigma = \mathrm{diag}(j^{-1})$, $\beta = 1$. W_1, W_2 are $d \times d$ matrices with i.i.d. entries $(W_i)_{jk} \sim \mathcal{N}(0, s^2/(4d))$, $s = 0.01$, evolved under the RD-AE flow (6)–(7). Each curve is the median over 128 seeds; cross-seed variance is below 10^{-3}. **(Left)** Effective dimension d_{eff}. Plain AE (black) dips and recovers; RD-AE (colored by rate R) dips and plateaus. **(Right)** Reconstruction loss. Dashed lines are Shannon floors $D(R)$, achieved by optimal enc-dec pair.

evolving statistics of z, RD-AE nevertheless exhibits dimensional collapse. Due to the Gaussianity of z, the solution of (5) is given by a classical waterfilling formula (Polyanskiy & Wu, 2025, Ex. I.9) and conveniently diagonalizes when the initialization and Σ does. Thus we obtain a similarly simple and insightful dynamics as in Saxe et al. (2014) except now including the VQ part.

RD-AE flow (dense case). Training is governed by the system of ODEs on W_1, W_2 (derived in Appendix A.1):

$$\dot{W}_1 = -2\big(W_2^\top W_2\, M\, W_1 - W_2^\top\big)\Sigma - 2\beta\big(W_1 - M W_1\big)\Sigma, \tag{6}$$

$$\dot{W}_2 = 2\Sigma W_1^\top M - 2 W_2\, U\, \mathrm{diag}(c_j \lambda_j) U^\top\,, \tag{7}$$

where M is the gain of the $z \to z_q$ channel ($\mathbb{E}[z_q \mid z] = Mz$), and c_j and λ_j are the spectra of M and $\mathrm{Cov}(z)$, respectively (see Appendix A.1). Importantly, when $K \to \infty$, the channel matrix $M \to I$, and the commitment term vanishes. In this limit the RD-AE flow (6)–(7) reduces to the *gradient flow* of the AE objective in Section 2, though the K required for this to happen is impractically large.

Numerical simulation (dense case). Before deriving the dynamics formally, we preview the qualitative behavior. Figure 4 simulates the *RD-AE flow* with $d = 64$, $\sigma_j^2 = j^{-1}$, $\beta = 1$, with i.i.d. Gaussian inititalization, compared with the plain-AE limit. The plain AE drives $\mathcal{L}_{\text{rec}} \to 0$ while d_{eff} dips and recovers. The RD-AE, by contrast, exhibits dimensional collapse at every rate $R \in \{10, 14, 18\}$ bits, to $d_{\text{eff}} = 3, 4, 5$ respectively, mirroring the behavior we see in practice. The reconstruction loss plateaus strictly above the Shannon distortion floor $D(R)$.

3.2 Analysis of the RD-AE dynamics in the diagonal case

To get more insight into the dynamics, we restrict attention to diagonal W_1, W_2:

$$W_1(t) =: \mathrm{diag}(u_j(t)), \quad W_2(t) =: \mathrm{diag}(v_j(t)), \quad \Sigma =: \mathrm{diag}(\sigma_j^2)\,.$$

It is clear that in this case equations (6)-(7) decouple and diagonality at initialization is preserved throughout training. Next, we derive equations for u_j, v_j and characterize the dimensional collapse mechanism via: (1) a per-mode flow that reduces to a Saxe et al. (2014) logistic with a channel-dependent timescale, (2) a dynamical water level whose rise pushes active modes below threshold and freezes them, and (3) a closed-form expression for the resulting loss plateau. The restriction to the diagonal case is only done for analytic convenience; this simplification still expresses all the major phenomena in the dense dynamics of W_1, W_2 (cf. (6)-(7)) that is simulated in Figure 2 and Figure 4.

The rate-distortion channel. Since W_1 is diagonal, $z_j = u_j x_j$ has independent coordinates $z_j \sim \mathcal{N}(0, \lambda_j)$ with $\lambda_j = u_j^2 \sigma_j^2$. The rate-$R$ channel $P_{z_q|z}$ minimizing $\mathbb{E}\|z - z_q\|^2$ under the information constraint factorizes across coordinates (Polyanskiy & Wu, 2025, Sec. 20.4, Ch. 26) and is given by reverse water-filling: a unique *water level* $D^\star = D^\star(\lambda_1, \ldots, \lambda_d; R)$, representing the variance threshold below which modes are dropped from the channel, solves

$$\textstyle\sum_j \frac{1}{2} \log_2^+(\lambda_j / D^\star) \;=\; R, \tag{8}$$

and the channel acts coordinatewise as $z_{q,j} \mid z_j \sim \mathcal{N}(c_j z_j, \tau_j^2)$ with per-mode distortion D_j, channel gain c_j, and noise variance τ_j^2 given by

$$D_j \;=\; \min(\lambda_j, D^\star), \qquad c_j \;=\; 1 - D_j/\lambda_j, \qquad \tau_j^2 \;=\; c_j D^\star. \tag{9}$$

Coordinates with $\lambda_j \le D^\star$ satisfy $c_j = \tau_j^2 = 0$ and are passed through as zero; we call these *inactive* and the rest *active*, and write the active set $\mathcal{A}(t) = \{j : \lambda_j(t) > D^\star(t)\}$ with $k(t) = |\mathcal{A}(t)|$. The channel decomposes as $z_j = z_{q,j} + n_j$ jointly, with $n_j \sim \mathcal{N}(0, D_j)$ independent of $z_{q,j}$, so that the total latent distortion $\mathbb{E}\|z - z_q\|^2 = \sum_j D_j$ equals $D(R)$ from (5).

Diagonal RD-AE flow. Differentiating the full training objective (1) requires care. We begin with the reconstruction loss $\mathcal{L}_{\text{rec}} = \mathbb{E}\|x - W_2 z_q\|^2$, which decomposes per mode using the joint decomposition $z = z_q + n$, coupled across modes through $D^\star$:

$$\mathcal{L}_{\text{rec}}(u, v) \;=\; \sum_{j=1}^{d} \Big[\sigma_j^2 (1 - c_j u_j v_j)^2 \;+\; v_j^2 \tau_j^2 \Big], \tag{10}$$

Under these conditions, the decoder gradient is directly computed from (10). For the encoder, we apply the chain rule with the STE and SG conventions ($\partial z_q / \partial z =: I$ and $\partial\, \mathrm{sg}(x)/\partial x := 0$) per-sample and take expectation. We get (full derivation in Section A.2):

$$\frac{\partial \mathcal{L}_{\text{rec}}}{\partial v_j} \;=\; -2 c_j u_j \sigma_j^2 (1 - u_j v_j), \qquad \left.\frac{\partial \mathcal{L}_{\text{rec}}}{\partial u_j}\right|_{\text{STE}} \;=\; -2\sigma_j^2 v_j (1 - c_j u_j v_j) \tag{11}$$

The encoder commitment gradient gives $\partial \mathcal{L}_{\text{com}} / \partial u_j = 2 D_j / u_j$. Assembling, we get the diagonal RD-AE flow

$$\dot{u}_j \;=\; 2\sigma_j^2 v_j (1 - c_j u_j v_j) \;-\; \frac{2\beta D_j}{u_j}, \qquad \dot{v}_j \;=\; 2\sigma_j^2 c_j u_j (1 - u_j v_j). \tag{12}$$

We note that due to the straight-through estimator, the flow (12) is *not a gradient* flow, since the symmetry condition $\partial \dot{u}_j / \partial v_j = \partial \dot{v}_j / \partial u_j$ fails whenever $c_j \neq 1$.

Mode dynamics at balanced initialization. We specialize to $\beta = 1$ and balanced initialization $u_j(0) = v_j(0) = \sqrt{s}$. Balance is preserved throughout training: for $j \in \mathcal{A}$, $D_j = D^\star = \lambda_j(1 - c_j) = u_j^2 \sigma_j^2 (1 - c_j)$, and at $u_j = v_j =: a_j$, we have $\dot{u}_j - \dot{v}_j = 0$.

Hence $u_j(t) = v_j(t)$ for all t where mode j is active. Then, $r_j := u_j v_j = u_j^2$ obeys

$$\dot{r}_j \;=\; 4\sigma_j^2\, c_j(t)\, r_j (1 - r_j) \quad \text{for } j \in \mathcal{A}, \qquad \dot{r}_j \;=\; 0 \quad \text{for } j \notin \mathcal{A}, \tag{13}$$

derived in Appendix A.4. This is structurally similar to logistic (4) with $c_j(t)$ acting as a time-dependent scale factor: at $c_j(t) \equiv 1$ (the $K \to \infty$ limit), (13) reduces to (3). For $j \notin \mathcal{A}$, $c_j = 0$ and $D_j = \lambda_j$ give $\dot{u}_j = \dot{v}_j = 0$ on the balanced manifold. The choice $\beta = 1$ preserves the encoder-decoder balance; our experiments reproduce the resulting predictions at β values used in practice, and we analyze other β in Appendix A.7.

Collapse mechanism. Implicit differentiation of (8) yields the water level dynamics:

$$\frac{\dot{D}^\star}{D^\star} \;=\; \frac{2}{k} \sum_{i \in \mathcal{A}} \frac{\dot{u}_i}{u_i}, \tag{14}$$

derived in Appendix A.3. As active modes grow, $D^\star$ rises through (14), and modes whose λ_j falls below the water line drop out of $\mathcal{A}$ and freeze permanently. The active set is therefore non-increasing, unlike in the plain AE, where every mode eventually activates. *This is the dimensional collapse phenomenon*: weak modes that have not yet activated when $D^\star$ overtakes them will remain unrepresented in the latent space forever. For general β, the mechanism is similar, because inactive modes receive no decoder gradient, see Appendix A.7.

Loss plateau. At convergence, with $k_\infty = |\mathcal{A}_\infty|$ active modes surviving, the reconstruction loss is

$$\mathcal{L}_{\text{rec}}^{\infty} \;=\; k_\infty D_\infty^\star \;+\; \sum_{j \notin \mathcal{A}_\infty} \sigma_j^2, \tag{15}$$

where the rate budget R is spent only on $\mathcal{A}_\infty$, leaving frozen modes unrepresented (paying their full variance σ_j^2). The plateau therefore exceeds the Shannon optimum $D(R)$ whenever $\mathcal{A}_\infty$ is a strict subset of the optimal active set (Figure 4).

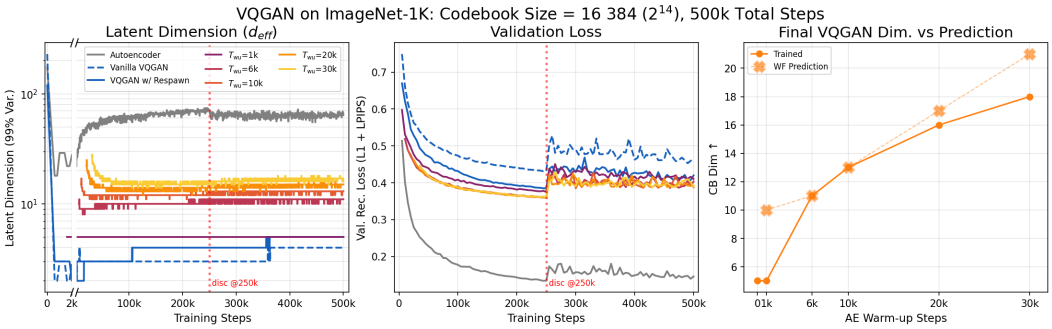


Figure 5: **VQGAN on ImageNet-1K,** $|\mathcal{C}| = 2^{14}$. Codebook d_{eff} (left) and validation reconstruction loss (center) during training for AE warm-up durations $T_{\text{wu}} \in \{0, 1\text{k}, \ldots, 30\text{k}\}$; longer warm-up raises codebook dimension and lowers the loss floor, with returns diminishing past $T_{\text{wu}} \approx 10\text{k}$. *Right:* reverse water-filling on the AE checkpoint's PCA spectrum at rate $R = 14$ is an empirical upper bound on the final trained codebook d_{eff}.

Special case of $\beta = 0$. Removing the commitment term by setting $\beta = 0$ leads to very different dynamics: (12) gives $\dot{u}_j = 2\sigma_j^2 v_j > 0$ on inactive modes, so RD-AE always eventually recovers from dimensional collapse at $\beta = 0$. This does not, however, justify dropping the commitment loss in practice: in VQGAN, $\beta = 0$ with dead-code reset recovers some dimension only at small K and remains clearly behind AE warm-up (Appendix C.2). We hypothesize that this discrepancy can be attributed to the finite codebook adaptation speed (causing staleness between codewords and z).

3.3 Effect of AE Warm-up

Section 3.2 showed that dimensional collapse happens because the rate constraint imposes a rising water level that freezes modes falling below it. A natural fix is *AE Warm-up*: delay quantization until enough modes have activated. This runs the plain-AE flow (3) from balanced diagonal initialization $W_1(0) = W_2(0) = \varepsilon I$ for warm-up duration T_{wu}, then switch to the RD-AE flow (12) with rate R and $\beta = 1$. The following theorem bounds how many modes survive the switch as a function of T_{wu}.

Theorem 1. *Let $x \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_d^2)$ and $\sigma_1^2 > \cdots > \sigma_d^2 > 0$. Let $g_j(T) = \left(1 + \frac{1-\varepsilon^2}{\varepsilon^2} e^{-4\sigma_j^2 T}\right)^{-1}$ denote the activation of mode j at time T in plain-AE training (4). Define*

$$m^{\text{wu}}(T_{\text{wu}}, R) \;:=\; \max\Big\{ m \in \{1, \ldots, d\} \;:\; R > \tfrac{1}{2} \sum_{i=1}^{m-1} \log_2 \tfrac{\sigma_i^2\, g_i(T_{\text{wu}})}{\sigma_m^2\, g_m(T_{\text{wu}})} \Big\}. \tag{16}$$

After switching to RD-AE flow at T_{wu}, at most m^{wu} modes remain active at convergence, and

$$\mathcal{L}_{\text{rec}}^{\infty} \;\geq\; m^{\text{wu}}\, \delta_{m^{\text{wu}}}(R) \;+\; \sum_{j > m^{\text{wu}}} \sigma_j^2, \qquad \delta_m(R) := \Big(\prod_{i=1}^{m} \sigma_i^2\Big)^{1/m} 2^{-2R/m}. \tag{17}$$

Proof sketch. At the switch time, $\lambda_j(T_{\text{wu}}) = g_j(T_{\text{wu}})\sigma_j^2$, and the condition in (16) is the water-filling requirement for the top m^{wu} modes to lie above $D^\star(T_{\text{wu}})$. Modes inactive at T_{wu} stay at $r_j = g_j(T_{\text{wu}})$, contributing variance σ_j^2 to the reconstruction loss. The bound on $\mathcal{L}_{\text{rec}}^{\infty}$ follows from optimal water-filling of rate R over the top m^{wu} modes; full proof in Appendix A.6.

Empirical validation of the warm-up theory. Although Theorem 1 is stated for the diagonal RD-AE model (where it is nearly tight; see Appendix A.6.1), its core construction transfers directly to non-linear VQ-VAEs. Indeed, one can snapshot a state of AE during the warm-up stage, compute PCA of the covariance matrix of the latents, evaluate water-filling rate allocation, and compute the number of surviving dimensions. The resulting number turns out to give a rather accurate upper bound on the final latent dimension of the trained VQ-VAE if the warm-up is switched off at the time of the snapshot (see Figure 5 (right) and Appendix C). This allows us to make on-the-fly automatic decision about when to switch from the AE (warm-up) phase and start training the actual VQ-VAE.

Table 1: **VQGAN on ImageNet-100 (20k) warm-up ablation.** L1, LPIPS, and rFID: lower is better. D: codebook d_{eff} (higher is better). Bold indicates the best value within each codebook size. Vanilla VQGAN uses uniform codebook init and no respawn; VQGAN w/ Respawn adds k-means init and respawn; $T_{\text{wu}} = \{1\text{k}, \ldots, 40\text{k}\}$ additionally pre-trains the encoder–decoder as a plain autoencoder for the indicated number of steps before introducing the codebook. All runs share the same 100k-step budget, so warm-up trades VQ steps for AE steps.

	$\lvert\mathcal{C}\rvert = 2^{10}$					$\lvert\mathcal{C}\rvert = 2^{14}$					$\lvert\mathcal{C}\rvert = 2^{16}$				
Variant	Util%	$D\uparrow$	L1↓	LPIPS↓	rFID↓	Util%	$D\uparrow$	L1↓	LPIPS↓	rFID↓	Util%	$D\uparrow$	L1↓	LPIPS↓	rFID↓
Vanilla VQGAN	82.7	2	0.153	0.405	87.8	8.0	2	0.153	0.408	81.0	2.2	2	0.153	0.406	83.6
w/ Respawn	100.0	3	0.148	0.389	74.2	99.9	3	0.140	0.372	69.7	47.0	5	0.140	0.363	67.9
$T_{\text{wu}} = 1\text{k}$	100.0	5	0.142	0.355	69.1	100.0	4	0.134	0.338	61.9	98.4	5	0.128	0.313	52.6
$T_{\text{wu}} = 5\text{k}$	100.0	7	0.141	0.343	61.7	100.0	7	0.129	0.308	51.4	98.5	7	**0.124**	0.294	47.4
$T_{\text{wu}} = 10\text{k}$	100.0	7	**0.141**	0.339	63.4	99.9	10	**0.128**	0.302	52.0	98.0	12	**0.124**	0.284	47.8
$T_{\text{wu}} = 15\text{k}$	100.0	10	0.142	0.337	67.0	99.9	12	0.129	0.299	51.4	97.7	13	**0.124**	0.282	47.3
$T_{\text{wu}} = 20\text{k}$	100.0	11	0.142	0.334	**60.9**	99.8	13	0.129	0.297	52.6	97.6	14	0.125	0.281	47.4
$T_{\text{wu}} = 30\text{k}$	100.0	14	0.142	**0.332**	61.8	99.8	15	0.130	0.295	**49.8**	97.2	16	0.126	0.279	46.7
$T_{\text{wu}} = 40\text{k}$	100.0	**19**	0.143	0.334	61.3	99.8	**17**	0.131	**0.294**	50.6	97.0	**18**	0.126	**0.279**	**43.9**

4 Escaping Dimensional Collapse with Autoencoder Warm-up

Section 3 argues that VQ-VAE reconstruction is bottlenecked by the latent effective dimension, not the codebook size: modes inactive when the quantizer turns on are frozen by the rate constraint and never recover. The suggested fix, *AE Warm-up*, trains the encoder–decoder as a plain autoencoder for T_{wu} steps so that lower-variance directions activate, and then introduces VQ. A single warm-up checkpoint can be reused across codebook sizes, amortizing its cost over a sweep.

We now test whether AE Warm-up prevents dimensional collapse in large-scale image (Section 4.1) and audio (Section 4.2) VQ-VAEs, and show that warm-up restores performance scaling with codebook scaling in VQGAN and WavTokenizer. In practice, T_{wu} can be selected by monitoring d_{eff} during AE training and switching once it plateaus; reverse water-filling on the latent PCA spectrum provides a rate-dependent prediction of the surviving VQ dimension (Section 3.3).

4.1 VQGAN on ImageNet

Setup. We evaluate AE Warm-up with f16 VQGAN in two settings: (i) controlled ablations on 20k ImageNet-100 images across $K \in \{2^{10}, 2^{14}, 2^{16}\}$ and $T_{\text{wu}} \in \{1\text{k}, \ldots, 40\text{k}\}$, and (ii) 500k-step ImageNet-1K training at $K = 2^{14}$ with adversarial loss enabled at 250k steps. We compare Vanilla VQGAN, VQGAN with k-means initialization, EMA codebook updates and dead-code respawn ("Respawn"), and AE Warm-up followed by the Respawn protocol. Full details are in Appendix B.1.

Key findings. Table 1 shows the final metrics. Without warm-up, neither Vanilla VQGAN nor the Respawn-only baseline improves substantially as K grows. AE Warm-up restores scaling with codebook size and improves all metrics over both baselines: 8.1–18.6% over Vanilla VQGAN and 5.3–11.0% over Respawn on L1, 18.0–31.4% and 14.5–23.1% on LPIPS, and 30.6–47.5% and 17.9–35.4% on rFID. Returns plateau beyond $T_{\text{wu}} = 20\text{k}$.

Cold-start training exhibits a loss floor. Vanilla VQGAN remains at $d_{\text{eff}} = 2$ as K grows from 2^{10} to 2^{2^16}, with L1 fixed at 0.153 and utilization falling from 82.7% to 2.2%. Respawn improves utilization and only marginally improves reconstruction, but reaches only d_{eff} =2-5, consistent with redistributing codes within the active subspace rather than recovering collapsed modes.

Warm-up lowers the loss floor. Increasing T_{wu} from 1k to 40k steps monotonically raises d_{eff} from 4–5 to 17–19 and improves LPIPS from 0.31–0.36 to 0.28–0.33, with noisier but consistent rFID gains. As per Theorem 1, additional warm-up allows more modes to be learned before quantization.

AE Warm-up benefits scale to training on ImageNet-1K with adversarial losses. Figure 5 and Table 3 show the results for $K = 2^{14}$ across several warm-up durations. Cold-start VQGAN collapses to d_{eff} =4–5, while AE Warm-up raises d_{eff} to 18 at $T_{\text{wu}} = 30\text{k}$, bounded by the reverse water-filling prediction. The best warm-up duration based on validation loss is $T_{\text{wu}} = 10\text{k}$, and the best rFID (5.35) is achieved by $T_{\text{wu}} = 20\text{k}$, beyond which increasing warm-up and dimension plateaus.

Table 2: **WavTokenizer audio reconstruction performance on LibriTTS test-clean/test-other.** Each row pair compares cold-start WavTokenizer training against AE Warm-up at the same codebook size. **Bold** indicates the winner within each codebook size; underline indicates the overall best. Without warm-up, codebook utilization collapses as K grows and codebook effective dimension is pinned at 4. AE Warm-up maintains near-full utilization at every K and raises codebook effective dimension to 11–21. Additionally, D (codebook d_{eff}), Mel, PESQ, and STOI improve monotonically with K. Cold-start at $K = 2^{16}$ is omitted given the severity of collapse at $K = 2^{14}$.

Method	$\lvert\mathcal{C}\rvert$	Util↑	D ↑	Mel↓	UTMOS↑	PESQ↑	STOI↑	V/UV F1↑
WavTokenizer	4096	100.0%	4	0.480/0.548	**4.051/3.563**	**2.336/2.054**	**0.912/0.878**	**0.940/0.916**
AE Warm-up	4096	100.0%	11	**0.469/0.526**	3.980/3.484	2.275/2.032	0.907/0.873	0.931/0.905
WavTokenizer	8192	16.9%	4	0.510/0.584	**4.009/3.547**	2.179/1.937	0.902/0.867	**0.934/0.908**
AE Warm-up	8192	100.0%	17	**0.448/0.504**	3.991/3.480	**2.425/2.147**	**0.915/0.882**	0.933/0.907
WavTokenizer	16384	8.4%	4	0.497/0.566	4.007/**3.539**	2.209/1.977	0.904/0.870	0.936/0.912
AE Warm-up	16384	100.0%	19	**0.432/0.486**	**4.041**/3.536	**2.514/2.229**	**0.920/0.889**	**0.942/0.917**
AE Warm-up	65536	99.7%	21	0.428/0.480	4.050/3.521	2.626/2.328	0.925/0.895	0.940/0.915

4.2 WavTokenizer on LibriTTS

Setup. To validate our theory in the audio domain and on an larger-scale setting, we train WavTokenizer (Ji et al., 2025) on LibriTTS (Zen et al., 2019) with full reconstruction and adversarial losses. We use four codebook sizes ($K \in \{4096, 8192, 16384, 65536\}$) and a fixed training budget of 100 epochs in two regimes: (i) cold-start WavTokenizer training (the baseline) and (ii) AE Warm-up for 43 epochs before initializing the codebook with k-means and training as a VQ-VAE with dead code respawn. All other hyperparameters match the original WavTokenizer; details are in Appendix B.2.

Key findings. Table 2 confirms the same pattern as VQGAN. Cold-start training collapses codebook to $d_{\text{eff}} = 4$ regardless of K, utilizing less than 1% of the available dimensionality. Codebook utilization collapses from 100% at $K{=}2^{12}$ to 8.4% at $K{=}2^{14}$. Performance does not merely plateau but *degrades* with K: test-clean PESQ drops from 2.34 at $K{=}2^{12}$ to 2.18 at $K{=}2^{13}$. The exception is $K{=}2^{12}$, where a 4-dimensional latent already tiles the codebook and warm-up's extra dimensions ($d_{\text{eff}}{=}11$) are surplus capacity—consistent with the theory.

AE Warm-up restores monotone scaling: codebook d_{eff} rises from 11 to 21 as K grows from 2^{12} to 2^{16}, all at near-full utilization. The reconstruction metrics also improve: Mel loss drops from 0.469 to 0.428 on test-clean, and PESQ improves from 2.275 to 2.626. We did not attempt cold-start at $K{=}2^{16}$ given the severity of collapse already at $K{=}2^{14}$.

5 Discussion and Limitations

The bottleneck in VQ-VAE training is the span of the latent subspace the encoder delivers to the quantizer, not the number of codes. AE warm-up addresses this directly, and because it acts before quantization begins, it is agnostic to the downstream quantizer: residual VQ (Zeghidour et al., 2021; Kumar et al., 2023; Zhang et al., 2024; Défossez et al., 2024), product quantization (Zhang et al., 2025), and structural reparameterizations such as SimVQ (Zhu et al., 2025) are all compatible. Indeed, methods that constrain latent dimension can improve performance but remain bottlenecked by their low-dimensional subspace (Appendix C.3.1), while methods that encourage higher dimension also benefit from warm-up (Appendix C.3.2).

Scope and future work. Our theory assumes linear encoders and decoders with independent Gaussian inputs — enough to expose the mechanism, but restrictive. Extending to nonlinear encoders and predicting the loss floor quantitatively in realistic models are natural next steps. The method also raises practical questions: how far can codebook size be pushed once warm-up is in place, and which existing codebook-geometry techniques continue to work in the higher-d_{eff} regime warm-up unlocks?

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, assist in the writing of proofs, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, and support data visualizations presented in the paper. We have not used generative AI tools to provide critical ingredients for proving mathematical claims, propose or refine hypotheses, assist with translation, and interpret results. Generating synthetic data sets is not applicable to this work. Additionally, we used AI to assist in drafting and editing parts of this paper. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide complete proofs for our theoretical results in Appendix A and detailed experimental settings, including architectures, training procedures, datasets, and evaluation protocols, in Appendix B. Code for reproducing our experiments is available at anonymous.4open.science/r/vqvae_latent_span_collapse-51BB.

REFERENCES

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, 2024. URL https://arxiv.org/abs/2410.00037.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, June 2021.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yBlVlS2Fd9.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 27980–27993. Curran Associates, Inc., 2023. doi: 10.52202/075280-1214. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/58d0e78cf042af5876e12661087bea12-Paper-Conference.pdf.

Adrian Lancucki, Jan Chorowski, Guillaume Sanchez, Ricard Marxer, Nanxin Chen, Hans J. G. A. Dolfing, Sameer Khurana, Tanel Alumäe, and Antoine Laurent. Robust training of vector quantized bottleneck models. *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2020. URL https://api.semanticscholar.org/CorpusID:218673823.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 51772–51783, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/e2dd53601de57c773343a7cdf09fae1c-Paper-Conference.pdf.

Max Morrison, Rithesh Kumar, Kundan Kumar, Prem Seetharaman, Aaron Courville, and Yoshua Bengio. Chunked autoregressive GAN for conditional waveform synthesis. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=v3aeIsY_vVX.

Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2025.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf.

Maria Refinetti and Sebastian Goldt. The dynamics of representation learning in shallow, non-linear autoencoders. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18499–18519. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/refinetti22a.html.

A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 2, pp. 749–752 vol.2, 2001. doi: 10.1109/ICASSP.2001.941023.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. pp. 4521–4525, 09 2022. doi: 10.21437/Interspeech.2022-439.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014. URL https://arxiv.org/abs/1312.6120.

Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4214–4217, 2010. doi: 10.1109/ICASSP.2010.5495701.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*, pp. 776–794, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58620-1. doi: 10.1007/978-3-030-58621-8_45. URL https://doi.org/10.1007/978-3-030-58621-8_45.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

Yuqing Wang, Zhijie Lin, Yao Teng, Yuanzhi Zhu, Shuhuai Ren, Jiashi Feng, and Xihui Liu. Bridging continuous and discrete tokens for autoregressive visual generation. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18596–18605, 2025. URL https://api.semanticscholar.org/CorpusID:277151126.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021. URL https://api.semanticscholar.org/CorpusID:236149944.

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

Fang Zhang, Yongxin Zhu, Yihao Liu, Bin Fu, and Linli Xu. Unveiling and addressing dimensional collapse in vector quantization models via codebook regularization. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=faDJYtfjWh.

Jiayou Zhang, Yifan Shen, Guangyi Chen, Le Song, and Eric P. Xing. Dimensional collapse in VQVAEs: Evidence and remedies. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=vJtnJfS2mQ.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, Los Alamitos, CA, USA, June 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00068. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00068.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speechtokenizer: Unified speech tokenizer for speech language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AF9Q8Vip84.

Wenhao Zhao, Qiran Zou, Rushi Shah, and Dianbo Liu. Representation collapsing problems in vector quantization, 2024. URL https://arxiv.org/abs/2411.16550.

Wenhao Zhao, Qiran Zou, Rushi Shah, Yudi Wu, Zhouhan Lin, and Dianbo Liu. Early quantization shrinks codebook: A simple fix for diversity-preserving tokenization, 2026. URL https://arxiv.org/abs/2603.17052.

Chuanxia Zheng and Andrea Vedaldi. Online clustered codebook. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22798–22807, October 2023.

Lei Zhu, Fangyun Wei, Yanye Lu, and Dong Chen. Scaling the codebook size of vq-gan to 100,000 with a utilization rate of 99%. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 12612–12635. Curran Associates, Inc., 2024. doi: 10.52202/079017-0401. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/1716d022edeac750e57a2986a7135e13-Paper-Conference.pdf.

Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. Addressing representation collapse in vector quantized models with one linear layer. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22968–22977, 2025. doi: 10.1109/ICCV51701.2025.02132.

A Proofs for Section 3

A.1 Dense RD-AE flow

We derive (6)–(7) using the notation of Section 3.1. All channel parameters $(c_j, \tau_j^2, D_j, D^\star)$ are treated as constant throughout, consistent with the assumption that the quantizer remains optimal for the current latent statistics.

Channel in the eigenbasis. We denote the covariance matrix of the encoder latent space as Σ_z, and write the diagonal decomposition $\Sigma_z = W_1 \Sigma W_1^\top = U \Lambda U^\top$. In coordinates $\tilde{z} = U^\top z$, the components $\tilde{z}_j \sim \mathcal{N}(0, \lambda_j)$ are independent, so the rate-R channel of Section 3.1 applies coordinatewise with parameters (c_j, τ_j^2, D_j) as in (9).

Returning to the original basis, $\mathbb{E}[z_q \mid z] = Mz$ with $M := U \operatorname{diag}(c_1, \ldots, c_d) U^\top$. Since $z_{q,j}$ is the MMSE estimate of $\tilde{z}_j$, the residual $n := z - z_q$ satisfies $z_q \perp n$ by Gaussianity, with $n \sim \mathcal{N}(0, U \operatorname{diag}(D_1, \ldots, D_d) U^\top)$.

Joint moments. The decomposition $z = z_q + n$ with $z_q \perp n$ gives

$$\mathbb{E}[z_q z_q^\top] = U \operatorname{diag}(c_j \lambda_j) U^\top =: \Gamma_q, \qquad \mathbb{E}[z z_q^\top] = \Gamma_q, \qquad \mathbb{E}[x z_q^\top] = \Sigma W_1^\top M, \tag{18}$$

where $M = U \operatorname{diag}(c_1, \ldots, c_d) U^\top$. The first two follow because $z_q \perp n$ implies $\mathbb{E}[z z_q^\top] = \mathbb{E}[z_q z_q^\top]$. The third uses $z_q = Mz + (\text{noise})$ and $z = W_1 x$, with M symmetric.

Decoder gradient. From $\mathcal{L}_{\text{rec}} = \mathbb{E}\|x - W_2 z_q\|^2 = \operatorname{tr}\Sigma - 2\operatorname{tr}(W_2\mathbb{E}[z_q x^\top]) + \operatorname{tr}(W_2^\top W_2 \Gamma_q)$, we treat the channel parameters (M, Γ_q) as constant and differentiate:

$$\nabla_{W_2}\mathcal{L}_{\text{rec}} = -2\Sigma W_1^\top M + 2W_2\Gamma_q. \tag{19}$$

$\mathcal{L}_{\text{com}}$ does not depend on W_2, so $\nabla_{W_2}\mathcal{L}_{\text{com}} = 0$.

Encoder reconstruction gradient under STE. The straight-through estimator uses $\partial z_q/\partial z := I$ in differentiation. Equivalently, replace z_q by the surrogate $z_q^{\text{surr}} = z + \text{sg}(z_q - z) = W_1 x + s$ where $s := z_q - W_1 x$ is treated as a constant. The reconstruction loss becomes

$$\mathcal{L}_{\text{rec}}^{\text{STE}} \;=\; \mathbb{E}\|x - W_2 W_1 x - W_2 s\|^2, \tag{20}$$

identical in value to $\mathcal{L}_{\text{rec}}$ but with W_1-dependence routed only through $W_1 x$. Differentiating in W_1 with s constant,

$$\nabla_{W_1}\mathcal{L}_{\text{rec}}|^{\text{STE}} \;=\; -2W_2^\top\mathbb{E}[(x - W_2 z_q)x^\top] \;=\; -2W_2^\top\Sigma + 2W_2^\top W_2\mathbb{E}[z_q x^\top], \tag{21}$$

where the residual $x - W_2W_1x - W_2s = x - W_2z_q$ uses the true z_q. Substituting $\mathbb{E}[z_q x^\top] = MW_1\Sigma$,

$$\nabla_{W_1}\mathcal{L}_{\text{rec}}|^{\text{STE}} \;=\; 2(W_2^\top W_2 M W_1 - W_2^\top)\Sigma. \tag{22}$$

Encoder commitment gradient. The commitment term $\mathcal{L}_{\text{com}} = \mathbb{E}\|z - \text{sg}(z_q)\|^2$ treats z_q as constant during differentiation, so it only depends on W_1 through $z = W_1 x$:

$$\mathcal{L}_{\text{com}} \;=\; \operatorname{tr}(W_1\Sigma W_1^\top) - 2\mathbb{E}[z^\top \text{sg}(z_q)] + \text{const}. \tag{23}$$

Differentiating, $\nabla_{W_1}\operatorname{tr}(W_1\Sigma W_1^\top) = 2W_1\Sigma$ and $\nabla_{W_1}\mathbb{E}[z^\top\text{sg}(z_q)] = \mathbb{E}[\text{sg}(z_q)x^\top]$ has population mean $MW_1\Sigma$, giving

$$\nabla_{W_1}\mathcal{L}_{\text{com}} \;=\; 2(W_1 - MW_1)\Sigma. \tag{24}$$

Assembling. Combining (19), (22), (24), we arrive at the flow in (6)–(7):

$$\dot{W}_1 \;=\; -2(W_2^\top W_2 M W_1 - W_2^\top)\Sigma \;-\; 2\beta(W_1 - MW_1)\Sigma, \tag{25}$$

$$\dot{W}_2 \;=\; 2\Sigma W_1^\top M - 2W_2\Gamma_q. \tag{26}$$

A.1.1 Dense water-level evolution

In the main text, we examined the water-level dynamics under diagonal initialization of W_1 and W_2 (Eq. (14), derivation in §A.3), which characterizes how $D*$ rises during RD-AE training to remove modes from the active set . We derive the water-level dynamics under dense initialization here.

Recall that the water-fill constraint $\sum_{j\in A}\frac{1}{2}\log_2(\lambda_j/D^\star) = R$ couples $D^\star$ to the eigenvalues of $\Sigma_z(t)$, which themselves evolve through $W_1(t)$. To track $\dot{\lambda}_j$, differentiate the eigenvalue identity $\Sigma_z u_j = \lambda_j u_j$ and the normalization $u_j^\top u_j = 1$:

$$\dot{\Sigma}_z\, u_j + \Sigma_z \dot{u}_j \;=\; \dot{\lambda}_j u_j + \lambda_j \dot{u}_j, \qquad u_j^\top \dot{u}_j = 0. \tag{27}$$

Left-multiplying by $u_j^\top$ and using $u_j^\top\Sigma_z = \lambda_j u_j^\top$ gives

$$\dot{\lambda}_j \;=\; u_j^\top \dot{\Sigma}_z\, u_j, \qquad \dot{\Sigma}_z \;=\; \dot{W}_1\Sigma W_1^\top + W_1\Sigma\dot{W}_1^\top. \tag{28}$$

The water-level evolution follows by differentiating the rate constraint at fixed active set:

$$\frac{\dot{D}^\star}{D^\star} \;=\; \frac{1}{k}\sum_{j\in A}\frac{\dot{\lambda}_j}{\lambda_j} \;=\; \frac{1}{k}\sum_{j\in A}\frac{u_j^\top\dot{\Sigma}_z u_j}{\lambda_j}. \tag{29}$$

Between water-crossings, A and k are constant, and (25)–(29) close the dynamics of $(W_1, W_2, D^\star)$. At a crossing, A updates discretely, k decrements by one, and $D^\star$ is continuous.

A.2 RD-AE flow derivation (diagonal case)

We derive the diagonal RD-AE flow (12) stated in Section 3.2. Throughout we assume that $W_1 = \text{diag}(u_j)$, $W_2 = \text{diag}(v_j)$. The RD channel of rate R bits acts coordinatewise on $z = W_1 x$.

The channel parameters $c_j, \tau_j^2, D^\star$ are treated as constant while differentiating the loss, as in §A.1. However, c_j and $D^\star$ change over time as the latent variances $\lambda_j = u_j^2\sigma_j^2$ evolve, and this coupling is derived separately in §A.3.

Key moments. In the diagonal case, the dense decomposition of §A.1 reduces to $z_j = z_{q,j} + n_j$ with $n_j \sim \mathcal{N}(0, D_j)$ independent of $z_{q,j}$, giving

$$\mathbb{E}[z_{q,j}^2] = c_j\lambda_j, \qquad \mathbb{E}[z_j\, z_{q,j}] = c_j\lambda_j, \qquad \mathbb{E}[x_j\, z_{q,j}] = c_j u_j \sigma_j^2. \tag{30}$$

Decoder gradient. The decoder direct gradient updates. From $\hat{x}_j = v_j z_{q,j}$,

$$\frac{\partial \mathcal{L}_{\text{rec}}}{\partial v_j} \;=\; -2\,\mathbb{E}\big[(x_j - v_j z_{q,j}) z_{q,j}\big] \;=\; -2\big(\mathbb{E}[x_j z_{q,j}] - v_j \mathbb{E}[z_{q,j}^2]\big) \;=\; -2c_j u_j \sigma_j^2 (1 - u_j v_j), \tag{31}$$

using $c_j\lambda_j = c_j u_j^2 \sigma_j^2$ in the last step. The commitment term does not depend on v_j, so $\partial\mathcal{L}_{\text{com}}/\partial v_j = 0$.

Encoder reconstruction gradient under STE. The straight-through estimator redefines the backward pass to treat $\partial z_{q,j}/\partial z_j := 1$, while the forward pass uses the actual $z_{q,j}$ from the channel. Concretely, STE replaces $z_{q,j}$ with a surrogate $z_{q,j}^{\text{surr}} = z_j + \text{sg}(z_{q,j} - z_j)$, which equals $z_{q,j}$ in value but has identity Jacobian in z_j. Under this surrogate, the reconstruction $\hat{x}_j = v_j z_{q,j}^{\text{surr}}$ satisfies $\partial\hat{x}_j/\partial u_j = v_j\, \partial z_j/\partial u_j = v_j x_j$, and

$$\left.\frac{\partial \mathcal{L}_{\text{rec}}}{\partial u_j}\right|_{\text{STE}} \;=\; -2v_j\,\mathbb{E}\big[(x_j - v_j z_{q,j}) x_j\big] \;=\; -2v_j\big(\sigma_j^2 - v_j \mathbb{E}[x_j z_{q,j}]\big) \;=\; -2v_j\sigma_j^2(1 - c_j u_j v_j). \tag{32}$$

The value of $z_{q,j}$ inside the residual uses the true channel output, not the surrogate — only the backward Jacobian is replaced.

Encoder commitment gradient. The commitment term $\mathcal{L}_{\text{com}} = \mathbb{E}\|z - \text{sg}(z_q)\|^2$ treats z_q as constant during differentiation. Since $z_j = u_j x_j$, differentiating with respect to u_j gives:

$$\frac{\partial \mathcal{L}_{\text{com}}}{\partial u_j} \;=\; 2\,\mathbb{E}\big[(z_j - z_{q,j}) x_j\big] \;=\; 2\big(u_j\sigma_j^2 - \mathbb{E}[x_j z_{q,j}]\big) \;=\; 2u_j\sigma_j^2(1 - c_j) \;=\; \frac{2D_j}{u_j}, \tag{33}$$

where the last equality uses $D_j = \lambda_j(1 - c_j) = u_j^2\sigma_j^2(1 - c_j)$.

Assembling the flow. The gradient flow is $\dot{u}_j = -\big(\partial\mathcal{L}_{\text{rec}}/\partial u_j|^{\text{STE}} + \beta\,\partial\mathcal{L}_{\text{com}}/\partial u_j\big)$ and $\dot{v}_j = -\partial\mathcal{L}_{\text{rec}}/\partial v_j$. Substituting gives

$$\dot{u}_j \;=\; 2\sigma_j^2\, v_j(1 - c_j u_j v_j) \;-\; \frac{2\beta D_j}{u_j}, \qquad \dot{v}_j \;=\; 2\sigma_j^2\, c_j u_j (1 - u_j v_j), \tag{34}$$

as stated in the main text.

A.3 Water-level dynamics in the diagonal case

Here, we derive (14) from Section 3.2, which characterizes how the reverse water-filling variance threshold rises throughout RD-AE training. As described in the main text, this is extremely consequential: it progressively decreases the active set of modes which are represented by the quantizer.

Given a rate budget R, a classical result (Polyanskiy & Wu, 2025, Ex. I.9) relates the rate-distortion water level $D^\star$ to the eigenvalues λ_j via the constraint:

$$\textstyle\sum_j \frac{1}{2}\log_2^+(\lambda_j/D^\star) \;=\; R, \tag{35}$$

We define the set of active modes to be $A := \{i : \lambda_i > D*\}$. The sum (35) then reduces to

$$\sum_{i\in A} \tfrac{1}{2}\log_2(\lambda_i/D^\star) \;=\; R. \tag{36}$$

We derive its dependence on u_i (equivalently, on $\lambda_i = u_i^2\sigma_i^2$) by implicit differentiation.

Rewriting (36) as $\sum_{i\in A}\log_2\lambda_i - k\log_2 D^\star = 2R$, where $k = |A|$, and differentiating in u_ℓ for $\ell \in A$:

$$\frac{1}{\lambda_\ell \ln 2}\cdot\frac{\partial\lambda_\ell}{\partial u_\ell} - \frac{k}{D^\star \ln 2}\cdot\frac{\partial D^\star}{\partial u_\ell} = 0. \tag{37}$$

Using $\partial\lambda_\ell/\partial u_\ell = 2u_\ell\sigma_\ell^2 = 2\lambda_\ell/u_\ell$, this gives

$$\frac{\partial D^\star}{\partial u_\ell} = \frac{2D^\star}{k\,u_\ell} \qquad \text{for } \ell \in A. \tag{38}$$

For $\ell \notin A$, mode ℓ does not enter (36) and $\partial D^\star/\partial u_\ell = 0$.

The time-derivative form follows by the chain rule along a flow trajectory:

$$\frac{\dot D^\star}{D^\star} = \sum_{\ell\in A}\frac{1}{D^\star}\frac{\partial D^\star}{\partial u_\ell}\dot u_\ell = \frac{2}{k}\sum_{\ell\in A}\frac{\dot u_\ell}{u_\ell}. \tag{39}$$

As active modes grow, $D^\star$ rises via (36). When $D^\star$ reaches λ_j for the weakest active mode, that mode transitions to inactive. Once inactive, it remains so: at $\beta = 1$, inactive modes are frozen ($\dot r_j = 0$ from (13)), so λ_j cannot grow back above $D^\star$.

A.4 Per-mode reduction at $\beta = 1$ and balanced init

We specialize the flow to $\beta = 1$ and balanced initialization $u_j(0) = v_j(0) = \varepsilon$. We show that (i) balance is preserved on active modes, (ii) the active-mode activation $r_j = u_j v_j = u_j^2$ obeys a Saxe-like logistic, and (iii) inactive modes at balance are frozen.

Balance preservation on active modes. From the flow with $\beta = 1$ and $j \in A$:

$$\dot u_j - \dot v_j = 2\sigma_j^2 v_j(1 - c_j u_j v_j) - \frac{2D_j}{u_j} - 2\sigma_j^2 c_j u_j(1 - u_j v_j). \tag{40}$$

Evaluating at $u_j = v_j =: a_j$ and using $D_j = D^\star = (1 - c_j)\lambda_j = (1 - c_j)a_j^2\sigma_j^2$ on active modes,

$$\dot u_j - \dot v_j\big|_{u_j = v_j} = 2\sigma_j^2 a_j(1 - c_j a_j^2) - 2\sigma_j^2(1 - c_j)a_j - 2\sigma_j^2 c_j a_j(1 - a_j^2) = 0. \tag{41}$$

Hence $u_j(t) = v_j(t)$ for all t on active modes; write $a_j(t)$ for this common value.

Active-mode logistic. At balance, $\dot u_j = \dot v_j$, so we compute either. From $\dot v_j = 2\sigma_j^2 c_j u_j(1 - u_j v_j)$ with $u_j = v_j = a_j$:

$$\dot a_j = 2\sigma_j^2 c_j a_j(1 - a_j^2). \tag{42}$$

With $r_j = a_j^2$, $\dot r_j = 2a_j\dot a_j$, giving

$$\dot r_j = 4\sigma_j^2 c_j\, r_j(1 - r_j) \qquad \text{for } j \in A, \tag{43}$$

which is Saxe's logistic (4) with timescale $1/(4\sigma_j^2)$ stretched to $1/(4\sigma_j^2 c_j)$ by the channel gain.

Inactive modes. For $j \notin A$, $c_j = \tau_j^2 = 0$ and $D_j = \lambda_j = u_j^2\sigma_j^2$. The flow reduces to

$$\dot u_j = 2\sigma_j^2 v_j - \frac{2u_j^2\sigma_j^2}{u_j} = 2\sigma_j^2(v_j - u_j), \qquad \dot v_j = 0. \tag{44}$$

At balance $u_j = v_j$, $\dot u_j = 0$ and $\dot v_j = 0$, so inactive modes at balance are stationary.

A.5 Plain-AE mode activation (closed form)

We derive the closed-form solution of the plain-AE flow from Saxe et al. (2014) for completeness and reference in the warm-up theorem proof. The plain-AE gradient flow from the decoupled loss is

$$\dot u_j = 2\sigma_j^2 v_j(1 - u_j v_j), \qquad \dot v_j = 2\sigma_j^2 u_j(1 - u_j v_j). \tag{45}$$

From $\dot u_j - \dot v_j = -2\sigma_j^2(1 - u_j v_j)(u_j - v_j)$, the balance gap $u_j - v_j$ decays to zero from any initial condition; in particular, balanced initialization $u_j(0) = v_j(0) = \varepsilon$ is preserved. Writing $a_j(t)$ for the common value and $r_j = a_j^2$,

$$\dot a_j = 2\sigma_j^2 a_j(1 - a_j^2), \qquad \dot r_j = 4\sigma_j^2 r_j(1 - r_j). \tag{46}$$

Separating variables, $\mathrm{d}r_j/[r_j(1-r_j)] = 4\sigma_j^2\,\mathrm{d}t$, and integrating from $r_j(0) = \varepsilon^2$ gives

$$r_j(t) \;=\; \left(1 + \tfrac{1-\varepsilon^2}{\varepsilon^2} e^{-4\sigma_j^2 t}\right)^{-1}. \tag{47}$$

A.6 Proof of Theorem 1

The proof has three steps: (i) identify the active set at the switch time, (ii) show warmup-inactive modes stay frozen, (iii) derive the loss bound.

State at the switch. Plain-AE training from balanced initialization $W_1(0) = W_2(0) = \varepsilon I$ preserves diagonality and balance (§A.5), so at time T_{wu} the weights are $W_1 = W_2 = \mathrm{diag}(\sqrt{g_j(T_{\mathrm{wu}})})$ and the latent eigenvalues are $\lambda_j(T_{\mathrm{wu}}) = g_j(T_{\mathrm{wu}})\,\sigma_j^2$. Since σ_j^2 is strictly decreasing in j and $g_j(T_{\mathrm{wu}})$ is strictly decreasing in j (larger σ_j^2 gives faster Saxe activation), $\lambda_j(T_{\mathrm{wu}})$ is strictly decreasing in j.

The active set at the switch is therefore a top prefix $\{1, \dots, m\}$, where m is the maximal integer satisfying the water-fill consistency condition $\lambda_m(T_{\mathrm{wu}}) > D_0^\star$, with $D_0^\star = \left(\prod_{i\le m} \lambda_i(T_{\mathrm{wu}})\right)^{1/m} 2^{-2R/m}$. Rearranging yields $R > \frac{1}{2}\sum_{i<m} \log_2(\sigma_i^2 g_i/\sigma_m^2 g_m)$, so $m = m^{\mathrm{wu}}(T_{\mathrm{wu}}, R)$.

Warmup-inactive modes are frozen forever. For $j > m^{\mathrm{wu}}$, mode j is inactive at the switch ($c_j = \tau_j^2 = 0$) and balanced. By §A.4, $\dot{u}_j = \dot{v}_j = 0$ at balance, so λ_j is constant. The water level $D^\star$ is non-decreasing along the flow (§A.3), so $\lambda_j < D^\star$ for all $t \ge T_{\mathrm{wu}}$, and mode j remains inactive.

The active set shrinks as a prefix. Active modes have $\dot{r}_j = 4\sigma_j^2 c_j r_j(1-r_j) \ge 0$ (§A.4), so λ_j is non-decreasing on the active set. Combined with monotonic $D^\star$, modes can only exit the active set, not enter. Since λ_j remains ordered (larger σ_j^2 drives faster growth, preserving the decreasing sequence), the active set at any later time is a prefix $\{1, \dots, k\}$ with $k \le m^{\mathrm{wu}}$. Any surviving mode converges to $r_j = 1$, giving $A_\infty = \{1, \dots, k_\infty\}$ with $k_\infty \le m^{\mathrm{wu}}$ and $\lambda_j = \sigma_j^2$ on A_∞.

Loss bound. At final state, the reconstruction loss is

$$\mathcal{L}_{\mathrm{rec}}^\infty \;=\; k_\infty\, D_\infty^\star \;+\; \sum_{j>k_\infty} \sigma_j^2 \;=\; k_\infty \big(\textstyle\prod_{i\le k_\infty} \sigma_i^2\big)^{1/k_\infty} 2^{-2R/k_\infty} \;+\; \sum_{j>k_\infty} \sigma_j^2, \tag{48}$$

since each surviving mode pays water-fill distortion $D_\infty^\star = \delta_{k_\infty}(R)$ and each inactive mode pays σ_j^2 (the decoder receives zero on inactive coordinates). This is the Shannon distortion of an idealized scheme that activates the top k_∞ modes and ignores the rest. Since $k_\infty \le m^{\mathrm{wu}}$ and the Shannon optimal scheme for the top m^{wu} source at rate R activates all m^{wu} modes (with distortion $m^{\mathrm{wu}}\delta_{m^{\mathrm{wu}}}(R) + \sum_{j>m^{\mathrm{wu}}} \sigma_j^2$), the k_∞-mode scheme is suboptimal, and

$$\mathcal{L}_{\mathrm{rec}}^\infty \;\ge\; m^{\mathrm{wu}}\, \delta_{m^{\mathrm{wu}}}(R) \;+\; \sum_{j>m^{\mathrm{wu}}} \sigma_j^2. \tag{49}$$

□

A.6.1 Numerical validation of Theorem 1

Figure 6 compares the theorem to simulation across warm-up durations in the linear RD-AE setting. Left panel: $m^{\mathrm{wu}}(T_{\mathrm{wu}}, R)$ from (16) against the observed final active-set size k_∞ in RD-AE simulation, for $R \in \{12, 14, 16\}$ bits. Right panel: the loss bound (17) against the observed final $\mathcal{L}_{\mathrm{rec}}^\infty$, with the rate-distortion floor $D(R)$ as reference. The figure shows T_{wu} in the regime where the bound is informative; at $T_{\mathrm{wu}} = 0$ the bound reduces to $D(R)$ and the observed loss exceeds it. The bound tightens as T_{wu} grows, and both k_∞ and $\mathcal{L}_{\mathrm{rec}}^\infty$ approach the Shannon limit.

A.7 Collapse for general commitment weight β

Equations (12) and (14) hold for any $\beta > 0$, and we set $\beta = 1$ in Section 3.2 because it simplifies the analysis. In the following we outline the general argument that explains how modes drop out at other values of β. It leads to qualitatively same result at meaningful values of β (away from 0) and only behaves differently when β is very small, in particular restoring modes at $\beta = 0$.

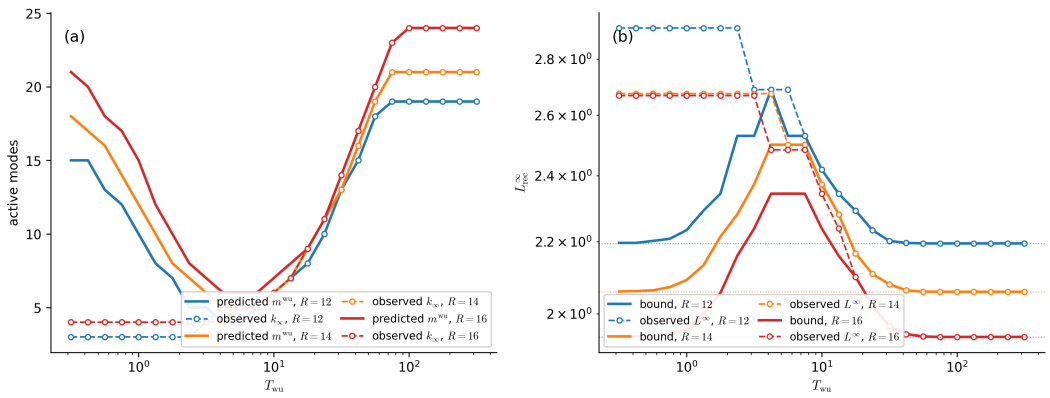


Figure 6: **Warmup theorem: predictions vs. simulation.** Balanced initialization $W_1(0) = W_2(0) = \varepsilon I$ with $\varepsilon = 0.1$, $\beta = 1$, $d = 64$, $\sigma_j^2 = j^{-1}$, three rates $R \in \{12, 14, 16\}$ bits. **(a)** Active-mode count at convergence. Solid lines: predicted $m^{\mathrm{wu}}(T_{\mathrm{wu}}, R)$ from (16). Dashed lines: observed k_∞ from simulated RD-AE flow. **(b)** Reconstruction loss. Solid lines: loss bound from (17). Dashed lines: observed $\mathcal{L}_{\mathrm{rec}}^{\infty}$. Dotted horizontal lines mark the Shannon floor $D(R)$ for each rate. As T_{wu} grows, predictions and observations both approach the Shannon limit; at small T_{wu} the bound is vacuous and the observed collapse is severe.

On an inactive mode, (12) reads $\dot{v}_j = 0$, $\dot{u}_j = 2\sigma_j^2(v_j - \beta u_j)$: the decoder is frozen, and the encoder only relaxes to v_j/β. An invariance argument for v_j/u_j with $u_j v_j < 1$ gives $v_j \le \max(1, \beta)\, u_j$ from balanced initialization. Hence a mode dropped at t_0 keeps $\lambda_j \le \max(1, \beta^{-2})\, D^\star(t_0)$ as long as its inactive, and can re-enter only while $D^\star$ is below this cap.

This explains why dimensional collapse persists beyond $\beta = 1$. A mode dropped early, when $D^\star(t_0) = O(\varepsilon^2)$, can increase its latent variance by at most the factor $\max(1, \beta^{-2})$ while inactive. Meanwhile, if the surviving modes converge, their variances approach σ_i^2/β and the water level approaches $D_\infty^\star = W/\beta$, where W is the water level of $\{\sigma_i^2\}_{i \in \mathcal{A}_\infty}$ at rate R. Comparing the inactive-mode cap with this learned water level gives

$$\frac{\bar{\lambda}_j}{D_\infty^\star} \le \frac{D^\star(t_0)}{W} \max(\beta, \beta^{-1}).$$

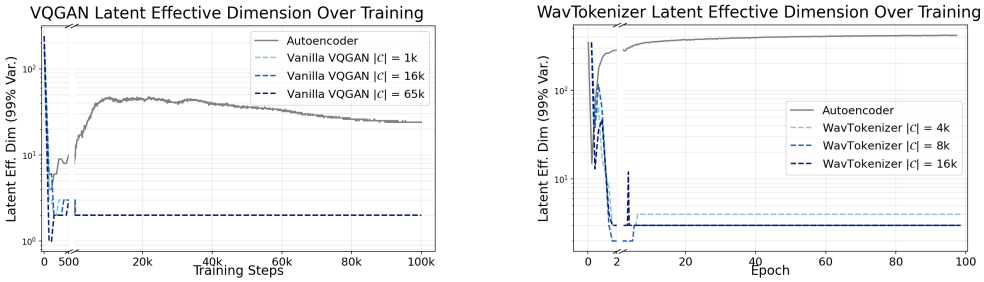
For fixed positive β, the cap therefore remains at initialization scale while the learned water level is independent of that scale. In particular, for $0 < \beta \le 1$, the cap is much smaller whenever $D^\star(t_0) \ll \beta W$; for modes dropped near initialization, $D^\star(t_0) \approx \varepsilon^2 W_0$, where W_0 is the water level of the full spectrum $\{\sigma_i^2\}$ at rate R, so the condition reads $\beta \gg \varepsilon^2 W_0 / W$. Thus the additional encoder growth allowed by $\beta < 1$ need not rescue early weak modes: their bounded recovery is surpassed by the growth of the water level. This provides the mechanism for a similar degree of collapse at moderate β, although re-entry during the initial rise remains possible.

In the setting of Fig. 2, k_∞ changes from 7 at $\beta = 0.1$ to shown 5 at $\beta = 1$, versus 19 Shannon-optimal active modes, confirming persistent dimensional collapse. At $\beta = 0$, this scale separation no longer follows: inactive encoders grow linearly without a finite cap, see Appendix C.2 ablation for that.

B Experiment Details

B.1 VQGAN Experiments

Architecture and loss. We use the f16 VQGAN encoder–decoder from Esser et al. (2021) with the `imagenet_vqgan.yaml` configuration (128 base channels, multipliers $[1, 1, 2, 2, 4]$, two residual blocks per level, self-attention at 16×16, 256-channel bottleneck, 256-dimensional codebook embedding space). At 128×128 input resolution each image is tokenized to an 8×8 grid of 64 discrete tokens. As per the original VQGAN training recipe, the discriminator is disabled for the



(a) VQGAN on ImageNet-100 (20k)

(b) WavTokenizer on LibriTTS.

Figure 7: **Without VQ, real architectures exhibit Saxe-like sequential activation when trained with Adam. VQ causes dimensional collapse.** Latent effective dimension (d_{eff}, 99%-variance threshold) during autoencoder and standard VQ-VAE training. In both (a) VQGAN and (b) WavTokenizer, standard VQ training pins latent dimension at 2-4, while dimension rises as modes are learned sequentially when training as an autoencoder.

first 250k training steps, the remaining loss is $\mathcal{L} = \mathcal{L}_{L_1} + \mathcal{L}_{\text{LPIPS}} + \lambda_{\text{cb}}\,\mathcal{L}_{\text{VQ}}$ with $\lambda_{\text{cb}} = 1.0$ and commitment weight $\beta = 0.25$, following Esser et al. (2021).

Training. All runs use the default VQGAN optimizer settings: Adam with $(\beta_1, \beta_2) = (0.5, 0.9)$, no weight decay, and constant effective learning rate 2.88×10^{-4}. We emphasize that any "warm-up" referenced in this paper is the AE phase on the model, not an LR schedule. ImageNet-100 runs are distributed across $4\times$ A100 40 GB GPUs at batch size 16/GPU (effective batch size 64), completing in approximately 5.5 hours of wall-clock time. ImageNet-1K runs are distributed across $2\times$ H200 GPUs at batch size 8/ GPU (effective batch size 16).

Codebook handling. *Vanilla VQGAN* initializes the codebook from $\mathcal{U}(-1/K, 1/K)$ and updates it via straight-through gradients. *VQGAN w/ Respawn* and all *AE Warm-up* recipes use k-means initialization on encoder outputs (100 Lloyd iterations over the full training set) with EMA updates, and respawn any code whose EMA usage (decay 0.99) falls below 0.01 from a random encoder output in the current batch.

Data. Our ImageNet-1K experiments are performed at 256×256 resolution. Our ImageNet-100 experiments sample 20,000 training images uniformly at random from the 100-class subset standardized by Tian et al. (2020), and retain the entire 5,000-image validation set. ImageNet-100 images are center-cropped and resized to 128×128.

Evaluation. We report L1 and LPIPS (VGG-16 backbone) as the standard pixel and perceptual reconstruction terms; Rec Loss is their sum. CB Util% is the fraction of codes receiving at least one assignment on the validation set, and CB Dim is the 99%-variance PCA effective dimension of the codebook entries. rFID uses the InceptionV3 backbone.

B.2 WavTokenizer Experiments

Architecture. We use the official WavTokenizer 75 FPS architecture from Ji et al. (2025) with a 512-dimensional pre-quantization latent and a single codebook. Unless otherwise noted, all architectural and loss hyperparameters are inherited from the released 75 FPS checkpoint configuration, including the convolutional encoder/decoder backbone, the multi-period and multi-scale STFT discriminators, and the full reconstruction loss suite (time-domain, mel-spectrogram, feature matching, adversarial). We vary only the codebook size $K \in \{4096, 8192, 16384, 65536\}$ and the training regime (cold-start vs. AE warm-up).

Training. All runs train for 100 epochs on $2\times$ H200 GPUs with a per-GPU batch size of 40; at 75 FPS this corresponds to 9,000 latent vectors per GPU per batch. The optimizer, learning-rate schedule, discriminator schedule, and loss weightings follow the released WavTokenizer recipe for

the 75FPS model with codebook size 4096. All runs use the same random seed (the WavTokenizer default); we did not run multiple seeds due to compute constraints–each run takes approximately 7 days on 2× H200 GPUs.

AE warm-up protocol. For warm-start runs, we first train the encoder and decoder as a plain autoencoder for 43 epochs, utilizing both reconstruction and discriminator loss. This single warm-up checkpoint is reused across all codebook sizes. At epoch 43, we enable quantization: we initialize the codebook via scikit-learn k-means on encoder outputs for $K \in \{4096, 8192, 16384\}$, and fall back to random initialization for $K = 65536$, where k-means becomes prohibitive at this codebook count. Training then proceeds for the remaining 57 epochs under the standard recipe with EMA codebook updates (decay 0.99) and respawn enabled. We did not sweep over warm-up duration due to compute constraints; 43 epochs was chosen as the point at which the latent effective dimension d_{eff} visibly plateaus in Figure 7b, matching the stopping rule advocated in Section 4.

Codebook respawn. We respawn codes whose EMA usage falls below a codebook-size-dependent threshold, aggregating usage counts across both GPUs each epoch for stability at large K. Cold-start baselines use the WavTokenizer default threshold of 1.0. Warm-start runs scale the threshold inversely with K to keep the expected number of respawned codes per epoch roughly constant:

Codebook size K	Dead-code threshold
4,096	1.0
8,192	0.5
16,384	0.25
65,536	0.06

Data. We train on both train-clean-360 and train-other-500 sampled at 24 kHz, with audio clips of 3s, following the original WavTokenizer training recipe (Ji et al., 2025). Validation and model selection use 100 randomly sampled batches of combined dev-clean and dev-other; final evaluation reports metrics on both LibriTTS test-clean and test-other. For each training run, we select the checkpoint with the best validation loss.

Evaluation. We report five reconstruction metrics on test-clean and test-other: mel-spectrogram L1 loss (Mel), UTMOS (Saeki et al., 2022), PESQ (Rix et al., 2001), STOI (Taal et al., 2010), and voiced/unvoiced F1 (V/UV F1) computed following the methodology of Morrison et al. (2022). With the exception of L1 loss, these match the metrics reported in Ji et al. (2025). Codebook utilization is computed over the full evaluation set, and codebook effective dimension $\dim(\mathcal{C})$ is the number of PCA components needed to explain 99% of the variance across codebook entries, matching the definition used for VQGAN in Appendix C.

C Additional results on VQGAN

Table 3 shows the final codebook utilization, d_{eff}, L1, LPIPS, and rFID for our ImageNet-1K experiments.

Figure 8 shows the training trajectories for the experiments across all codebook sizes and warm-up runs, and Figure 9 shows the effect of the number of AE warm-up steps on final reconstruction loss after 100k total training steps. We observe that for all codebook sizes, the effect of adding warm-up steps plateaus at around 20k steps, and increasing warm-up duration past this point does not improve final loss.

C.1 Water-filling predicts final VQ-VAE codebook dimension

Theorem 1 bounds the surviving codebook dimension by $m_{\text{wu}}(T_{\text{wu}}, R)$, the number of latent PCA components that lie above the Shannon water level at rate $R = \log_2 K$. The theorem is stated for the linear-Gaussian RD-AE; here we test whether the this holds for VQGAN.

Table 3: **VQGAN on ImageNet-1K with $|\mathcal{C}| = 2^{14}$.** L1, LPIPS, and rFID: lower is better. D: codebook d_{eff} (higher is better). Vanilla VQGAN uses uniform codebook initialization and no respawn; VQGAN w/ Respawn adds k-means initialization and respawn; T_{wu} additionally pre-trains the encoder–decoder as a plain autoencoder for the indicated number of steps before introducing the codebook. Bold indicates the best value among VQ models.

Variant	Util%	$D\uparrow$	L1$\downarrow$	LPIPS$\downarrow$	rFID$\downarrow$
Vanilla VQGAN	8.1	4	0.143	0.304	7.91
w/ Respawn	100.0	5	0.136	0.284	7.07
$T_{\text{wu}} = 1$k	100.0	5	0.131	0.280	7.22
$T_{\text{wu}} = 6$k	100.0	11	0.136	0.268	5.97
$T_{\text{wu}} = 10$k	100.0	13	**0.127**	**0.260**	5.88
$T_{\text{wu}} = 20$k	100.0	16	0.131	**0.260**	**5.35**
$T_{\text{wu}} = 30$k	100.0	**18**	0.129	**0.260**	5.37

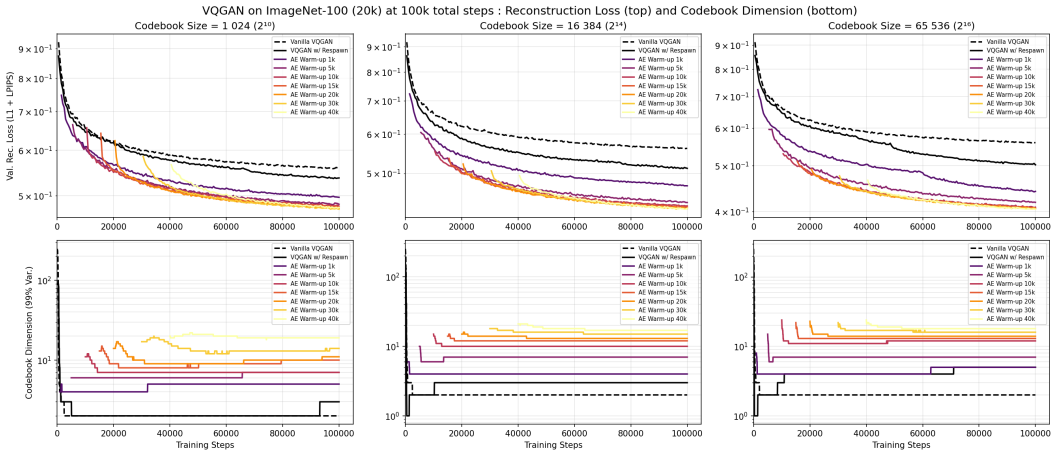


Figure 8: **VQGAN on ImageNet-100: codebook size $\times$ warm-up duration.** Training runs spanning $K \in \{1\text{k}, 16\text{k}, 65\text{k}\}$ and $T_{\text{wu}} \in \{0, 1\text{k}, 5\text{k}, 10\text{k}, 20\text{k}, 30\text{k}, 40\text{k}\}$ steps of AE Warm-up, all with k-means init and respawning of the codebook (solid lines), compared with the vanilla VQGAN training recipe with random codebook init and no respawn (dashed line). *Top row:* validation reconstruction loss (L1 + LPIPS) during training. *Bottom row:* codebook effective dimension d_{eff} (99% variance threshold) during training. Vanilla VQGAN collapses to $d_{\text{eff}} = 2$ and hits a K-independent loss floor regardless of codebook size. Adding dead code respawn (No Warm-up) partially rescues codebook utilization and modestly raises d_{eff}, but still plateaus in validation loss well above the warm-up runs. Additional AE warm-up steps further increases the codebook effective dimension and lowers the reconstruction floor.

Method. For each AE warm-up checkpoint, we compute the PCA spectrum of the pre-quantization latent over the full ImageNet-100 (20k) training set. We then compute reverse water-filling at rates $R \in \{10, 14, 16\}$ bits, matching the codebook sizes used in Table 1. The number of PCA components whose eigenvalue exceeds $D^\star$ gives the predicted active-mode count m_{wu}. Figure 10 visualizes the spectrum and water lines at three checkpoints; Figure 11 compares m_{wu} to the trained VQ-VAE codebook effective dimension at every (T_{wu}, K) pair.

Key findings. The water-filling prediction sits strictly above the trained codebook dimension in all 24 settings, validating the theorem's bound qualitatively in the nonlinear regime. As predicted, the predicted dimension grows with T_{wu} (as more weak modes are learned during AE training) and with K (higher rate lowers the water line, admitting more modes). Both trends are reproduced by the trained VQ-VAEs.

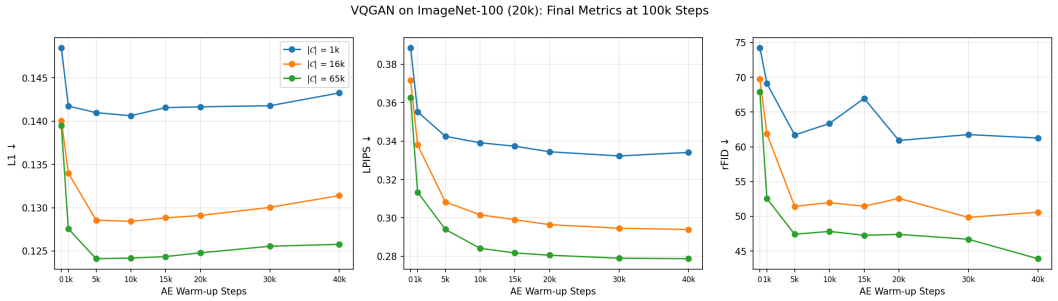


Figure 9: **AE warm-up vs. codebook-size scaling in VQGAN.** Final reconstruction metrics on the ImageNet-100 validation set at step 100,000—L1 (left), LPIPS (center), and rFID (right)—as a function of AE warm-up length $T_{\text{wu}} \in \{0, 1\text{k}, 5\text{k}, 10\text{k}, 15\text{k}, 20\text{k}, 30\text{k}, 40\text{k}\}$, for three codebook sizes $K \in \{2^{10}, 2^{14}, 2^{16}\}$. $T_{\text{wu}} = 0$ corresponds to the VQGAN w/ Respawn baseline (k-means initialization plus dead-code respawn, no AE phase). At $T_{\text{wu}} = 0$ the three curves are tightly clustered: increasing K by $64\times$ yields only a modest reduction in any of the three metrics. As T_{wu} grows, the curves separate into the expected ordering $K = 2^{16} < 2^{14} < 2^{10}$ and the gap between them widens, indicating that warm-up not only lowers the reconstruction floor but also restores the scaling benefit of a larger codebook. Returns diminish past $T_{\text{wu}} \approx 20\text{k}$, matching the plateau of the AE-phase effective-dimension trajectory in Figure 7.

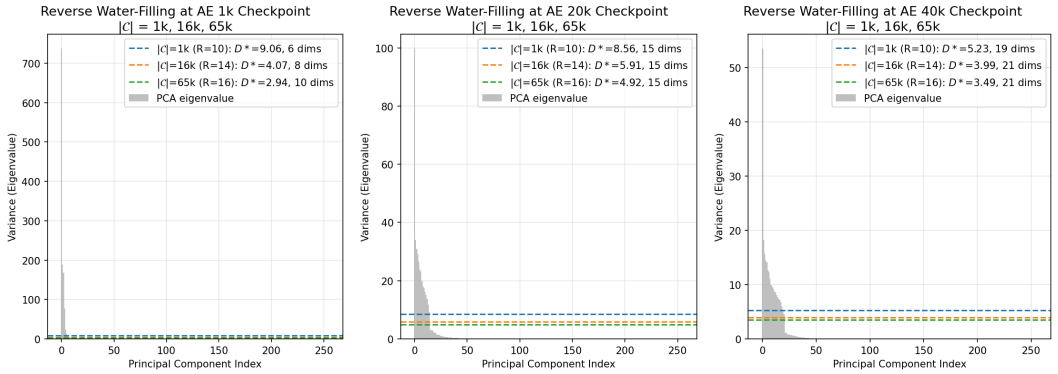


Figure 10: **AE latent PCA spectrum and water levels at three warmup durations.** Gray bars: PCA eigenvalues of the pre-quantization latent at AE warmup steps 1k, 20k, and 40k (left to right), computed over the full ImageNet-100 (20k) training set. Dashed horizontal lines: Shannon water level $D^\star$ at rates $R = \log_2 K$ for codebook sizes $K \in \{2^{10}, 2^{14}, 2^{16}\}$, computed by reverse water-filling (Eq. 8) on the displayed spectrum. The number of PCA components above each line is the predicted active-mode count $m_{\text{wu}}(T_{\text{wu}}, R)$ from Theorem 1. Two trends visible across panels: as T_{wu} grows, more modes lift above each line (weak modes activate during AE training); at fixed T_{wu}, larger K pushes the water line lower and admits more modes.

C.2 VQGAN WITH $\beta = 0$

The linear analysis in Section 3 identified $\beta = 0$ as a degenerate case in which inactive modes are not strictly frozen: at $\beta = 0$, the encoder ODE reads $\dot{u}_j = 2\sigma_j^2 v_j$, which is positive for any $v_j > 0$, so inactive modes can in principle grow back into the active set. Figure 12 and Table 4 test this directly by removing the commitment term from VQGAN.

Vanilla VQGAN at $\beta = 0$. Without commitment loss, codebook utilization drops to 3.7%, 0.3%, and 0.0% at $K = 2^{10}, 2^{14}, 2^{16}$ respectively. The codebook effective dimension is reported as 248–254, but this is a degenerate artifact: random codebook initialization spreads codewords across the embedding space, and with no commitment term to pull encoder outputs toward existing codes, almost no codes are ever assigned. The unassigned codes remain at their random init values, inflating the PCA span without improving reconstruction.

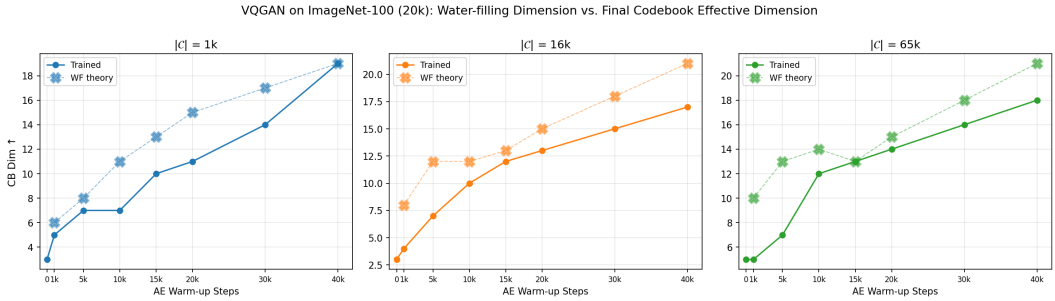


Figure 11: **Water-filling on the AE latent spectrum upper-bounds the trained VQ-VAE codebook dimension.** For each AE warm-up checkpoint $T_{\text{wu}} \in \{0, 1\text{k}, 5\text{k}, 10\text{k}, 15\text{k}, 20\text{k}, 30\text{k}, 40\text{k}\}$ and each codebook size $K \in \{2^{10}, 2^{14}, 2^{16}\}$ (one panel per K), we compare the water-filling prediction $m_{\text{wu}}(T_{\text{wu}}, R)$ from the AE PCA spectrum (dashed lines) against the codebook effective dimension of the corresponding VQ-VAE trained from that checkpoint to 100k total steps (solid lines). The predicted dimension lies above the trained dimension at every point and across all three codebook sizes: water-filling on the AE latent is an empirical upper bound on what the downstream VQ-VAE achieves. Both quantities increase monotonically with T_{wu}, and the gap between them is roughly constant.

Table 4: **ImageNet-100 (20k) warm-up ablation, with $\beta = 0$ baselines.** L1, LPIPS, and rFID: lower is better. Dim: codebook d_{eff} (higher = more of the codebook span is used). Bold indicates the best value within each codebook size. Vanilla VQGAN uses uniform codebook init and no respawn; VQGAN w/ Respawn adds k-means init and respawn; $T_{\text{wu}} = \{1\text{k}, \ldots, 40\text{k}\}$ additionally pre-trains the encoder–decoder as a plain autoencoder for the indicated number of steps before introducing the codebook. The $\beta = 0$ rows remove the commitment loss; without it, vanilla VQGAN collapses to near-zero codebook utilization while the encoder is free to span many latent dimensions, and respawn alone partially compensates for utilization but still falls short of warm-up at large K. All runs share the same 100k-step budget, so warm-up trades VQ steps for AE steps.

	$\lvert\mathcal{C}\rvert = 2^{10}$					$\lvert\mathcal{C}\rvert = 2^{14}$					$\lvert\mathcal{C}\rvert = 2^{16}$				
Recipe	Util%	Dim↑	L1↓	LPIPS↓	rFID↓	Util%	Dim↑	L1↓	LPIPS↓	rFID↓	Util%	Dim↑	L1↓	LPIPS↓	rFID↓
Vanilla VQGAN	82.7	2	0.1530	0.4051	87.82	8.0	2	0.1532	0.4083	81.01	2.2	2	0.1525	0.4064	83.59
w/ $\beta = 0$	3.7	248	0.2100	0.4959	161.42	0.3	253	0.3856	0.5903	300.06	0.0	254	0.4399	0.5911	368.17
w/ Respawn	100.0	3	0.1484	0.3885	74.21	99.9	3	0.1401	0.3717	69.74	47.0	5	0.1395	0.3626	67.89
w/ $\beta = 0$	99.8	22	0.1566	0.3723	78.01	99.8	5	0.1362	0.3383	61.32	98.6	5	0.1300	0.3220	54.54
$T_{\text{wu}} = 1\text{k}$	100.0	5	0.1417	0.3553	69.11	100.0	4	0.1340	0.3380	61.86	98.4	5	0.1276	0.3134	52.55
$T_{\text{wu}} = 5\text{k}$	100.0	7	0.1410	0.3425	61.70	100.0	7	0.1286	0.3083	51.43	98.5	7	**0.1241**	0.2940	47.41
$T_{\text{wu}} = 10\text{k}$	100.0	7	**0.1406**	0.3391	63.35	99.9	10	**0.1284**	0.3016	51.97	98.0	12	0.1242	0.2841	47.83
$T_{\text{wu}} = 15\text{k}$	100.0	10	0.1416	0.3373	66.97	99.9	12	0.1288	0.2990	51.44	97.7	13	0.1244	0.2817	47.27
$T_{\text{wu}} = 20\text{k}$	100.0	11	0.1416	0.3344	**60.91**	99.8	13	0.1291	0.2965	52.58	97.6	14	0.1248	0.2806	47.40
$T_{\text{wu}} = 30\text{k}$	100.0	14	0.1418	**0.3323**	61.75	99.8	15	0.1300	0.2946	**49.84**	97.2	16	0.1255	0.2790	46.70
$T_{\text{wu}} = 40\text{k}$	100.0	19	0.1433	0.3341	61.26	99.8	17	0.1314	**0.2939**	50.59	97.0	18	0.1258	**0.2787**	**43.89**

Respawn at $\beta = 0$. With k-means initialization and dead-code respawn, $\beta = 0$ raises the codebook dimension at small K (22 vs. 3 at $K = 2^{10}$), as the linear analysis predicts, but at larger K it stays at $\sim$5, like the $\beta = 0.25$ baseline. Its reconstruction relative to that baseline is mixed (worse L1 and rFID at $K = 2^{10}$, better at $K \geq 2^{14}$), and performs worse than AE Warm-up at every codebook size.

Role of the commitment loss. The commitment term is intended to stabilize the encoder–codebook relationship by anchoring encoder outputs to codeword positions. However, as we show in Section 3, this loss term suppresses weak modes and leads to domension collapse. Removing the commitment loss prevents the suppression but breaks the stabilization, leading to higher codebook dimension but significantly worse reconstruction loss and codebook utilization, as shown in Table 4. AE warm-up sidesteps this trade-off entirely: weak modes activate *before* VQ is introduced, and we see that training with warm-up outperforms $\beta = 0$ across every codebook size.

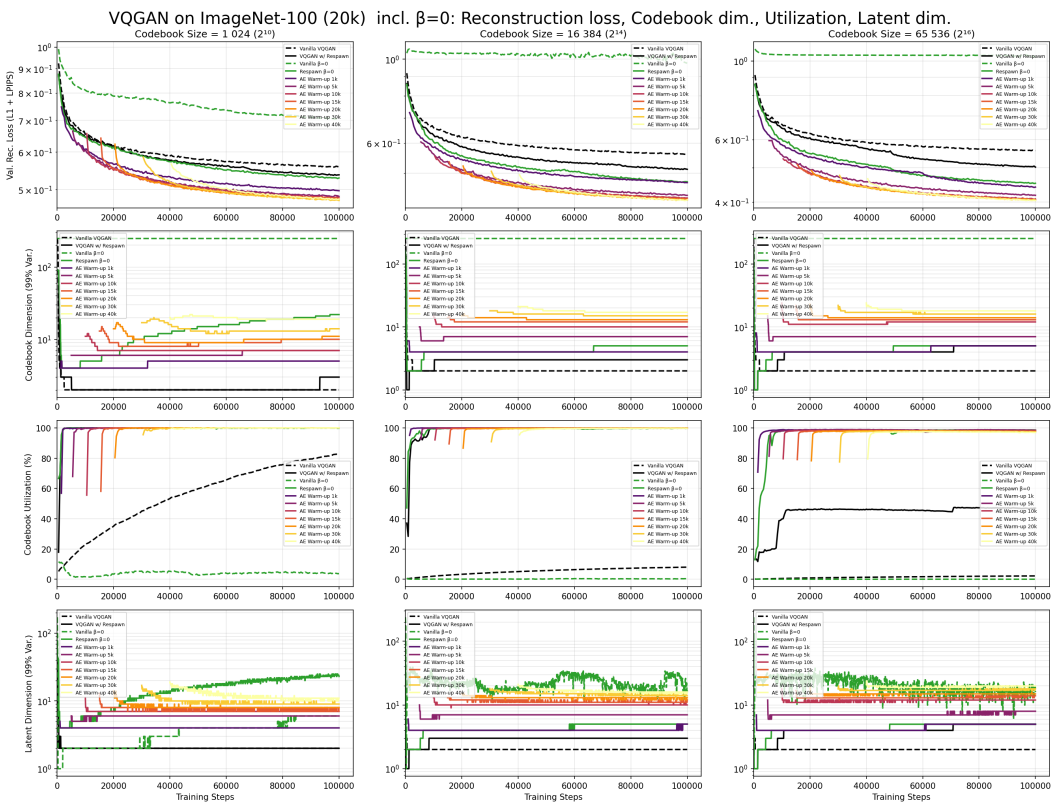


Figure 12: **VQGAN on ImageNet-100 (20k) with the commitment term removed.** Training curves at $\beta = 0$ (vanilla and respawn variants, dashed) overlaid on the standard $\beta > 0$ runs from Figure 8. Rows, top to bottom: validation reconstruction loss (L1 + LPIPS), codebook effective dimension, codebook utilization, latent effective dimension. Columns: codebook sizes $K \in \{2^{10}, 2^{14}, 2^{16}\}$. Vanilla VQGAN at $\beta = 0$ collapses to near-zero codebook utilization at every K — almost all codewords are dead from initialization onward, since without the commitment term pulling the encoder toward the codebook, randomly initialized codes are never assigned. Respawn at $\beta = 0$ keeps codes alive and admits a higher latent effective dimension at small K (consistent with the linear-theory escape route in Section 3.2), but remains well behind every AE warm-up run with $T_{\text{wu}} \geq 5\text{k}$.

C.3 Other VQ-VAE Training Methods

C.3.1 Methods that constrain codebook dimension

The following Tables 5, 6, 7 use a different random seed from Tables 1 and 4, so the shared Respawn and warm-up rows differ. Absolute values, codebook d_{eff} in particular, vary across seeds, but the warm-up gain is stable: the rFID improvement over Respawn at $T_{\text{wu}} = 40\text{k}$ is 12.9/19.1/24.0 in Table 1 and 15.5/18.2/23.2 in Table 5 for $K = 2^{10}/2^{14}/2^{16}$. Because the warm-up checkpoints also differ across seeds, the water-filling predictions in Appendix C.1 apply only to Table 1, and the codebook d_{eff} values in Tables 5–7 should not be compared against them.

Some popular codebook training methods have been shown to improve reconstruction performance while purposely constraining dimension. We find that that these methods do result in a performance improvement over the original VQGAN recipe, but are still bottlenecked by their latent dimension.

FSQ (Mentzer et al., 2024) is a product quantization method that projects the latent representation down to a low-dimensional space and quantizes each coordinate independently. We follow the standard recipe for $K = 1k, 16k, 65k$ and compare against the VQGAN w/ Respawn recipe as well as our AE warm-up recipe with $T_{\text{wu}} = 40\text{k}$ in Table 5. At every matched codebook size, FSQ

Table 5: **Comparison with FSQ on ImageNet-100.** L1, LPIPS, and rFID: lower is better. D: codebook d_{eff} (higher is better). VQGAN w/ Respawn uses k-means initialization and respawn; $T_{\text{wu}} = 40$k additionally pre-trains the encoder–decoder as a plain autoencoder for 40k steps before introducing the codebook. For FSQ, the effective codebook size is determined by the number of quantization levels.

Family	$\lvert\mathcal{C}\rvert$	Util%	$D\uparrow$	L1↓	LPIPS↓	rFID↓
VQGAN w/ Respawn ($T_{\text{wu}} = 0$)	1k (2^{10})	100.0	3	0.141	0.361	69.9
	16k (2^{14})	100.0	4	0.130	0.342	59.3
	65k (2^{16})	99.0	4	0.129	0.341	62.3
VQGAN $T_{\text{wu}} = 40$k	1k	100.0	17	0.143	0.320	54.4
	16k	99.9	23	0.127	0.274	41.1
	65k	95.8	27	0.122	0.260	39.1
FSQ	1k ([8,5,5,5], $d = 4$)	∼100*	4*	0.142	0.349	65.3
	16k ([8,8,8,6,5], $d = 5$)	∼100*	5*	0.129	0.315	52.9
	65k ([8,8,8,5,5,5], $d = 6$)	∼100*	6*	0.127	0.309	53.3

Table 6: **Comparison with VQGAN-LC on ImageNet-100.** L1, LPIPS, and rFID: lower is better. D: codebook d_{eff} (higher is better). All models are trained for 100k total steps.

	$D = 4$					$D = 8$					$D = 16$				
Variant	Util%	$D\uparrow$	L1↓	LPIPS↓	rFID↓	Util%	$D\uparrow$	L1↓	LPIPS↓	rFID↓	Util%	$D\uparrow$	L1↓	LPIPS↓	rFID↓
VQGAN w/ Respawn	100.00	4	0.131	0.352	58.05	100.00	4	0.130	0.342	70.68	100.00	4	0.132	0.344	66.80
w/ $T_{\text{wu}} = 40$k	100.00	4	0.129	0.334	60.06	99.99	8	0.124	0.300	54.04	100.00	13	0.126	0.284	43.43
VQGAN-LC	96.89	4	0.130	0.343	58.25	99.26	7	0.127	0.294	48.59	99.67	6	0.127	0.300	42.00
w/ $T_{\text{wu}} = 40$k	97.69	4	0.132	0.334	57.12	99.39	8	0.125	0.301	50.37	99.40	11	0.126	0.281	40.46

consistently improves LPIPS and rFID over VQGAN w/ Respawn, but falls short of our T_{wu} run, suggesting that FSQ's low-dimensional constraint is a limiting factor.

VQGAN-LC (Zhu et al., 2024) initializes the codebook with representations from a converged image encoder that is trained without quantization. Our theory says that such an encoder should be able to represent low-variance feature directions and avoid dimensional collapse. Their method then quantizes these features and trains a projector to embed this rich feature representation into a lower dimensional space. This abstractly mirrors our proposed method of training the VQ-VAE itself without quantization during warm-up and then adding quantization–and corroborates our view that lower-variance feature directions that are missing from an early-quantization model are important. Indeed, VQGAN-LC also shows rFID improvement as ambient dimension scales in Table 6.

C.3.2 Methods that encourage higher codebook dimension

To address dimensional collapse in VQ-VAES, Zhang et al. (2025) proposed DCVQ, a product quantization method that splits the latent vector into subvectors of dimension $d_s = 8$ and quantizes each subvector independently based on cosine similarity. We compare DCVQ with a a latent dimension of 16 (resulting in 2 separate $d_s = 8$ subspaces that are quantized independently with $K_s = 256$ codes each) against VQGAN w/ Respawn and our AE Warm-up recipe in Table 7. We find that DCVQ results in better reconstruction metrics than the VQGAN w/ Respawn recipe as well as higher dimension, but each 8-dimensional subspace exhibits mild collapse: the product codebook utilizes 8/16 total dimensions. but underperforms $T_{\text{wu}} = 20$k. Additionally, DCVQ also benefits from AE Warmup: when training as an autoencoder for the first 20k steps, DCVQ w/ $T_{\text{wu}} = 20$k shows improved LPIPS and rFID, and also utilizes all 16/16 available codebook dimensions.

D Additional results on WavTokenizer

Figure 13 shows training trajectories for the WavTokenizer runs reported in Table 2.

Table 7: **Comparison with DCVQ on ImageNet-100 with $|\mathcal{C}| = 2^{16}$.** L1, LPIPS, and rFID: lower is better. D: codebook d_{eff} (higher is better). All models are trained for 100k total steps.

Variant	Util%	$D\uparrow$	L1$\downarrow$	LPIPS$\downarrow$	rFID$\downarrow$
VQGAN w/ Respawn	99.0	4	0.129	0.341	62.33
w/ $T_{\text{wu}} = 20$k	96.5	25	0.122	0.262	39.59
DCVQ (2×256, $d_s = 8$)	92.4	8	0.125	0.295	49.32
w/ $T_{\text{wu}} = 20$k	91.3	16	0.132	0.270	40.16

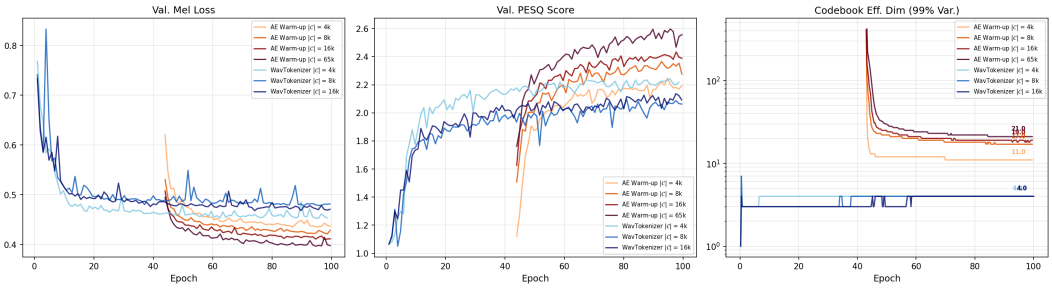


Figure 13: **WavTokenizer on LibriTTS: cold-start training fails to benefit from larger codebooks; AE Warm-up restores codebook scaling.** Cold-start WavTokenizer (blue curves, $K \in \{4\text{k}, 8\text{k}, 16\text{k}\}$) versus AE-VQ (warm curves, $K \in \{4\text{k}, 8\text{k}, 16\text{k}, 65\text{k}\}$). *From left:* validation mel loss, PESQ, codebook d_{eff} (99% variance). AE Warm-up runs are trained for 43 epochs before introducing VQ. Cold-start runs show flat loss and low effective dimension regardless of K; AE-VQ runs show monotone improvement with K and substantially higher effective dimension in both latent and codebook spaces.

We notice that, when switching from autoencoder training to VQ training at epoch 43, PESQ sharply degrades, but quickly recovers and surpasses the no-warm-up runs.

Cold-start runs plateau in Mel loss quickly, after the first 20–30 epochs. AE Warm-up runs continue to improve Mel loss after the quantizer is enabled; the additional training budget appears productive only when the latent has enough effective dimension.