ASYMPTOTICALLY OPTIMAL QUANTIZER-GUIDED SINKHORN ALIGNMENT FOR MITIGATING CODEBOOK COLLAPSE

Anonymous authors
Paper under double-blind review

ABSTRACT

Codebook collapse remains a fundamental limitation of vector quantization (VQ), causing poor codebook utilization and increased quantization error. While recent methods improve utilization through distribution alignment, directly matching the codebook to the encoder distribution can deviate from the asymptotically optimal codeword distribution, leading to excessive codeword concentration in high-density regions and insufficient coverage of lower-density regions. We introduce AOptSA, a theory-driven quantization framework motivated by asymptotically optimal quantizer theory. Its key insight is to align the codebook with a density-reweighted target distribution, rather than the raw encoder distribution, thereby better balancing probability mass and spatial coverage. AOptSA realizes this target online via importance sampling from encoder outputs and minimizes Sinkhorn divergence between the sampled target and codebook distributions. The resulting alignment adapts throughout training without altering nearest-neighbor quantization or latent dimensionality. Across image benchmarks, AOptSA achieves full codebook utilization, consistently improves reconstruction quality over strong baselines, and maintains its gains as codebook size increases.

1 INTRODUCTION

Vector Quantization (VQ) is a key technique for discrete representation learning that maps continuous representations into a finite set of discrete codewords. It has been widely adopted in image generation to learn compact visual representations and convert images into sequences of discrete tokens. More recently, VQ-based visual tokenizers have been adopted in token-based multimodal models, enabling images and text to be represented and processed within a unified sequence-modeling framework (Ramesh et al., 2021).

Increasing the VQ codebook size increases the nominal capacity of the discrete latent space by providing a larger set of candidate codewords. In vanilla VQ models, however, simply enlarging the codebook may yield diminishing returns because larger codebooks are more susceptible to codebook collapse, in which only a small subset of codewords is selected and updated while many others remain inactive (Kaiser et al., 2018; Takida et al., 2022). This underutilization reduces the effective capacity of the codebook and can result in suboptimal assignments and higher quantization error, thereby degrading the learned discrete representations (Huh et al., 2023). Consequently, realizing the representational benefits of large codebooks requires maintaining broad and effective codeword utilization.

To address codebook collapse, several methods have been proposed, including low-dimensional mappings (Yu et al., 2022), structured quantization (Jegou et al., 2011; Défossez et al., 2023), stochastic quantization (Roy et al., 2018; Baykal et al., 2024), and distribution alignment (Huh et al., 2023; Fang et al., 2025; 2026). Low-dimensional and structured approaches simplify codebook learning but often reduce representation fidelity, while stochastic methods improve codeword exploration at the cost of increased quantization error and computational overhead. Distribution alignment offers a more direct solution by reducing the mismatch between the codebook and encoder outputs. However, existing methods typically align the codebook with the raw encoder-output distribution. Asymptotically optimal quantizer theory suggests that this distribution can deviate

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.

from the asymptotic codeword distribution, particularly when the quantization-space dimensionality is not sufficiently large relative to the distortion order. This mismatch may lead to less effective spatial allocation of codewords and motivate a density-reweighted alignment target.

To address this limitation, we revisit the codebook collapse problem from the perspective of asymptotically optimal quantizer theory (Graf & Luschgy, 2000). This framework provides a principled basis for deriving an asymptotic target distribution for codeword allocation. Building upon this foundation, we further propose AOptSA, a dynamic distribution alignment method that constructs target samples via importance sampling over the encoder-output distribution (Robert & Casella, 2004). This method offers two key advantages: it requires no explicit distributional assumptions, making it adaptable to complex data distributions, and it can continuously adapt to dynamically evolving encoder outputs. We adopt Sinkhorn divergence for distribution alignment. As a fully differentiable metric with high computational efficiency and geometric fidelity (Cuturi, 2013; Genevay et al., 2018; Feydy et al., 2019), it is well suited for modeling complex data. Owing to these designs, AOptSA improves codebook utilization and reconstruction quality by regularizing the global structural arrangement of the codebook, while preserving the original feature dimensionality during quantization.

Extensive experiments demonstrate that AOptSA maintains high codebook utilization and steady quantization quality across diverse datasets and codebook sizes. These findings indicate that AOptSA scales well with increasing codebook sizes.

The main contributions of this work are as follows:

- We revisit codebook collapse from the perspective of asymptotically optimal quantizer theory and refine the alignment target adopted by existing distribution-matching methods. Our analysis shows that the empirical distribution of asymptotically optimal codebooks converges to a density-reweighted distribution induced by the encoder outputs, rather than the raw encoder-output distribution itself. The distinction is more pronounced when the quantization-space dimensionality is small relative to the distortion order.
- We propose AOptSA, a dynamic distribution-alignment method that constructs target samples through iterative importance sampling directly from encoder outputs. This data-driven sampling procedure allows AOptSA to track evolving feature distributions without assuming a predefined distributional prior. Sinkhorn divergence is then used to align the codebook with the sampled target distribution while preserving the geometric fidelity of the latent space.
- Extensive experiments demonstrate that AOptSA achieves a favorable balance between codebook utilization and reconstruction quality. By retaining the original latent dimensionality and improving the global organization of codewords, AOptSA increases codebook perplexity and consistently enhances reconstruction performance on image datasets. Moreover, it scales effectively to large codebooks, maintaining high codebook utilization and improving quantization quality as codebook capacity increases.

2 Related Work

Codebook collapse remains a persistent challenge in vector quantization (VQ), especially for large codebooks, where only a small fraction of codewords are frequently activated during training. This issue leads to insufficient codebook utilization, limited effective capacity, and increased quantization error. A variety of mitigation strategies have been developed to alleviate this problem.

As relatively indirect remedies, several methods modify the latent space or quantization structure through low-dimensional, factorized, product, residual, or multi-stage representations (Yu et al., 2022; Mentzer et al., 2024; Yu et al., 2024; Jegou et al., 2011; Défossez et al., 2023; El-Nouby et al., 2023; Guo et al., 2023; 2024; Lee et al., 2022; Chen et al., 2025; Zhang et al., 2024b). These methods ease training difficulty by limiting feature dimensions and regularizing codebook structure, thereby alleviating the collapse dilemma inherent to large-capacity codebooks. Although they mitigate severe codeword imbalance, such artificial constraints weaken latent representation capacity, leading to limited quantization accuracy and degraded reconstruction performance.

Unlike these indirect constraints, another line of research directly modifies the assignment mechanism of nearest-neighbor quantization. In conventional VQ, each encoder feature is deterministically assigned to its nearest codeword, and only the selected codeword receives a direct update, which can progressively suppress rarely used codewords. To alleviate this bias, several methods introduce stochasticity into codeword assignment through distance-based categorical sampling or Bayesian uncertainty modeling (Roy et al., 2018; Kaiser et al., 2018; Sønderby et al., 2017; Takida et al., 2022; Baykal et al., 2024). OptVQ (Zhang et al., 2024a) replaces local nearest-neighbor assignment with a globally balanced optimal-transport matching, allowing the assignment of each feature to account for the overall feature-codebook structure. These methods improve codeword participation by reformulating the assignment process, although they alter the original nearest-neighbor quantization rule and may introduce additional computational overhead.

Beyond randomized assignment, SimVQ (Zhu et al., 2025) mitigates codebook collapse by reparameterizing code vectors with a learnable linear basis. By freezing the coefficient codebook and optimizing the basis matrix, it shifts the optimization from isolated code-vector updates to joint adaptation of the transformed codebook space. However, SimVQ still retains nearest-neighbor assignment and offers no principled construction of an optimal codebook; consequently, its codebook organization remains largely data-driven and empirically learned.

More recently, distribution alignment methods have attributed codebook collapse to mismatch between the codebook and encoder-output distributions. Huh et al. (2023) address this issue through a shared affine transformation that updates all codewords using global statistics, including rarely selected ones. Wasserstein-VQ (Fang et al., 2025) assumes Gaussian feature and codebook distributions and minimizes their closed-form quadratic Wasserstein distance through mean and covariance matching. MMD-VQ (Fang et al., 2026) removes the Gaussian assumption by using characteristic-kernel maximum mean discrepancy (MMD) to align codebook and feature distributions. However, both methods directly match the codebook to the raw encoder-output distribution. Asymptotically optimal quantizer theory suggests that this distribution can differ from the asymptotic codeword distribution, particularly when the quantization-space dimensionality is not sufficiently large relative to the distortion order.

In contrast, our work revisits distribution alignment through asymptotically optimal quantizer theory and refines the alignment target used by prior methods. Rather than directly matching the codebook to the raw encoder-output distribution, we adopt a density-reweighted target, with the distinction becoming more pronounced at lower quantization dimensions relative to the distortion order.

3 Background

3.1 An Overview of Vector Quantization

Vector-quantized (VQ) neural networks combine classical vector quantization with deep neural encoders and decoders for efficient discrete representation learning. A typical VQ model comprises three components: an encoder, a vector quantizer, and a decoder (van den Oord et al., 2017). The quantizer is implemented as a learnable codebook $Q = \{e_1, \dots, e_K\}$, where K is the codebook size and each codeword has dimension d. Given an input x, the encoder produces a latent feature $z_e(x)$, which is quantized by the nearest codeword e_k in the codebook:

$$k = \arg\min_j \|z_e(x) - e_j\|_2^2 \tag{1}$$

The selected embedding $z_q(x) = e_k$ is decoded to reconstruct $\hat{x}$. Because nearest-neighbor quantization is non-differentiable, gradients are passed to the encoder using the straight-through estimator (STE) (van den Oord et al., 2017; Bengio et al., 2013).

The training objective typically consists of three terms: a reconstruction loss, a quantization loss for updating the codebook, and a commitment loss that encourages consistency between the encoder outputs and the selected codewords. The overall objective is

$$\mathcal{L}_{\mathrm{VQ-VAE}} = \|x - \hat{x}\|_2^2 + \|\mathrm{sg}[z_e(x)] - z_q(x)\|_2^2 + \beta\|z_e(x) - \mathrm{sg}[z_q(x)]\|_2^2 \tag{2}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator and β controls the weight of the commitment term.

3.2 Asymptotically Optimal Quantizer Theory

To motivate the target distribution used in AOptSA, we review a classical result from asymptotically optimal quantizer theory (Graf & Luschgy, 2000). Let $Z \in \mathbb{R}^d$ denote an encoder-output random vector with distribution P_z and density h_z.

For a fixed encoder-output distribution P_z and an r-order distortion criterion, classical vector quantization seeks a codebook $Q = \{e_k\}_{k=1}^K$ that minimizes the expected quantization distortion

$$D_r(Q; P_z) = \mathbb{E}_{Z \sim P_z}\left[\min_{e \in Q} \|Z - e\|_2^r\right]. \tag{3}$$

Assume that P_z is absolutely continuous with respect to the d-dimensional Lebesgue measure and that $\mathbb{E}[\|Z\|^{r+\delta}] < \infty$ for some $\delta > 0$, where r denotes the order of the quantization distortion. Let Q^* be an asymptotically optimal set of centers for P_z under the distortion objective above. As the number of centers tends to infinity, the empirical distribution of Q^*, denoted by P_e, weakly converges to a limiting distribution P_r (Graf & Luschgy, 2000):

$$P_e = \frac{1}{K}\sum_{e \in Q^*} \delta_e \xrightarrow{\mathcal{D}} P_r, \;\; as \; K \to \infty \tag{4}$$

where δ_e denotes the Dirac measure concentrated at e, and the limiting distribution P_r has density:

$$h_r(z) \propto [h_z(z)]^{d/(d+r)} \tag{5}$$

For squared error ($r = 2$), this becomes:

$$h_r(z) \propto [h_z(z)]^{d/(d+2)} \tag{6}$$

Wasserstein-VQ (Fang et al., 2025) notes that h_r approaches h_z when d is sufficiently large relative to r. However, its 8-dimensional quantization space with $r = 2$ gives $\frac{d}{d+r} = 0.8$, so matching the codebook directly to the encoder-output distribution may deviate from the asymptotic target.

This distinction motivates AOptSA. For practical codebooks, we use Eq. 5 as a theoretically motivated target. When d is not sufficiently large relative to r, directly matching the empirical codebook distribution to P_z is generally inconsistent with this target. Because $\frac{d}{d+r} < 1$, density reweighting attenuates sharp peaks, allocating relatively fewer codewords to very dense regions and more to lower-density regions with substantial geometric extent. This balances probability mass and geometric coverage.

Importantly, the asymptotic result above characterizes optimal quantizers for a fixed distribution and a specified distortion criterion. It does not imply that P_r is the finite-codebook optimum of the full jointly trained VQGAN objective, where the encoder distribution evolves during training and reconstruction, perceptual, and adversarial losses are optimized simultaneously. In AOptSA, we therefore use P_r as a theoretically motivated regularization target for organizing the codebook under a dynamically evolving encoder distribution.

4 Methodology

AOptSA mitigates codebook collapse by aligning the empirical codebook distribution with a density-reweighted target motivated by asymptotically optimal quantizer theory. It constructs target samples online through importance sampling and minimizes a Sinkhorn-based alignment loss.

4.1 Target Distribution

Unlike methods that match the codebook directly to P_z or its moments, AOptSA targets the density-reweighted distribution P_r motivated by the asymptotic result in Eqs. 5 and 6.

Because the encoder distribution evolves and its density is unknown, the target cannot be fixed in advance. At each iteration, AOptSA estimates density from current encoder outputs, reweights them, and samples a target set. This online surrogate tracks the evolving representation while approximating the asymptotic codebook distribution.

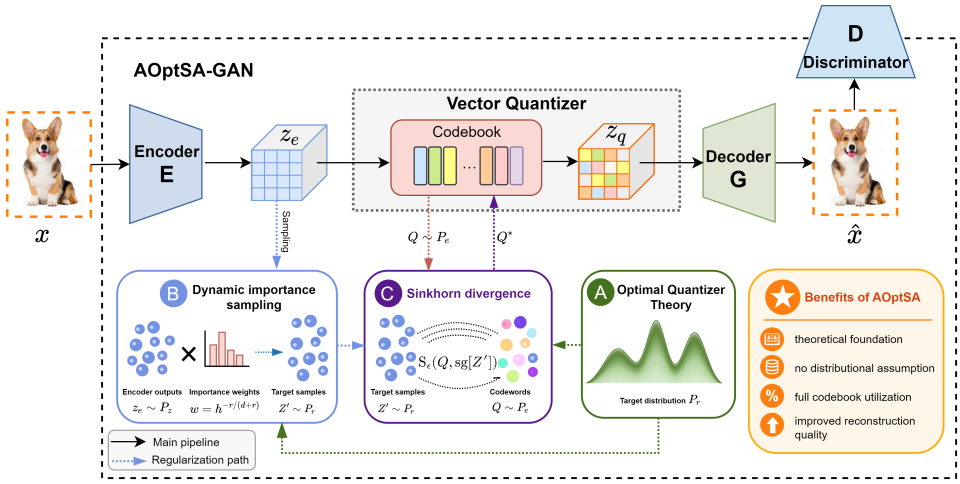


Figure 1: Overview of AOptSA-GAN. Guided by sampling weights motivated by asymptotically optimal quantizer theory, AOptSA reweights encoder outputs to construct a target sample set Z' that approximates P_r. It then minimizes the Sinkhorn divergence $\mathrm{S}_\epsilon(Q, \mathrm{sg}[Z'])$ to align the empirical codebook distribution with this target, thereby encouraging the empirical codebook distribution to better match the density-reweighted target motivated by asymptotically optimal quantizer theory.

4.2 Dynamic Distribution Alignment via Importance Sampling and Sinkhorn Divergence

Because direct sampling from P_r is unavailable, we use Monte Carlo importance sampling (Robert & Casella, 2004). Encoder outputs are assigned weights:

$$w_{z_i} = \frac{h_r(z_i)}{h_z(z_i)} \propto \frac{[h_z(z_i)]^{d/(d+r)}}{h_z(z_i)} = [h_z(z_i)]^{-r/(d+r)} \tag{7}$$

Sampling according to these weights yields a target set $Z' = \{z'_i | z'_i \in \mathbb{R}^d\}_{i=1}^N$ that approximates P_r. In practice, $h_z(z_i)$ can be estimated with a non-parametric density estimator, such as kNN (Fukunaga & Hostetler, 1973).

Given a codebook $Q = \{e_k | e_k \in \mathbb{R}^d\}_{k=1}^K$, we represent the codebook and the sampled target by their empirical distributions, defined as follows:

$$P_e = \frac{1}{K}\sum_{k=1}^{K} \delta_{e_k}, \qquad \widehat{P}_r = \frac{1}{N}\sum_{i=1}^{N} \delta_{z'_i} \tag{8}$$

We align P_e and $\widehat{P}_r$ by minimizing the Sinkhorn divergence $\mathrm{S}_\epsilon(Q, \mathrm{sg}[Z'])$ (Cuturi, 2013; Genevay et al., 2018; Feydy et al., 2019). Unlike moment matching or local regularization, Sinkhorn divergence constrains global codebook-space geometry while remaining computationally tractable through entropic regularization (Courty et al., 2017; Peyré & Cuturi, 2019).

Detailed density estimation, importance sampling, and Sinkhorn implementation are provided in Appendix B.

4.3 Integrating with VQ-based Models

AOptSA regularizes the codebook directly while leaving the encoder, decoder, and nearest-neighbor quantization rule unchanged. It requires no low-dimensional projection, making it applicable to VQ-VAE, VQGAN, and related models. We describe its integration with VQGAN below.

As illustrated in Figure 1, AOptSA-GAN follows the standard encoder-quantizer-decoder architecture (Esser et al., 2021). The encoder produces $z_e(x)$, which is quantized normally. Importance sampling constructs Z', whose empirical distribution $\widehat{P}_r$ approximates P_r, and Sinkhorn divergence

aligns P_e with $\widehat{P}_r$. The original features are passed to the quantizer without projection or other changes.

The final optimization objective for AOptSA is:

$$\mathcal{L}_{VQGAN} = \mathcal{L}_{VQVAE} + \mathcal{L}_{Per} + \lambda \mathcal{L}_{GAN} + \gamma \mathrm{S}_{\epsilon}\left(Q, \mathrm{sg}[Z']\right) \tag{9}$$

where $\mathcal{L}_{VQVAE}$ denotes the standard reconstruction and quantization objective, $\mathcal{L}_{Per}$, $\mathcal{L}_{GAN}$, and $\mathrm{S}_{\epsilon}\left(Q, \mathrm{sg}[Z']\right)$ denote the perceptual loss (Johnson et al., 2016), adversarial loss, and Sinkhorn distribution alignment loss, $\mathrm{sg}[Z']$ denotes the stop-gradient operation. The coefficient γ is a hyperparameter that controls the regularization strength of the Sinkhorn divergence. The combined objective improves codebook utilization and reconstruction quality.

5 Experiments

We evaluate AOptSA on three image reconstruction benchmarks and study its dynamic alignment, target distribution, codebook dimensionality, and scalability. Unless stated otherwise, all methods use the same VQGAN backbone.

5.1 Experimental Settings

5.1.1 Datasets

We evaluate AOptSA on ImageNet-1K (Deng et al., 2009), ImageNet-100 (Tian et al., 2020), and FFHQ (Karras et al., 2019). ImageNet-1K contains approximately 1.28M training and 50K validation images across 1,000 classes, while ImageNet-100 is a 100-class subset with about 130K training and 5K test images. FFHQ contains 70K face images, which we split into 90% training and 10% test sets. Together, these datasets cover large-scale object-centric, reduced-scale object-centric, and face-domain reconstruction settings.

5.1.2 Baselines and Model Settings

We compare AOptSA with representative methods for mitigating codebook collapse, including VQGAN-EMA (Razavi et al., 2019), FSQ (Mentzer et al., 2024), Wasserstein-VQ (Fang et al., 2025), and SimVQ (Zhu et al., 2025). All methods use the same backbone architecture as SimVQ and differ only in their quantization module. Quantization dimensionality follows each method's design; hence, SimVQ, which shares AOptSA's 128-dimensional quantization space, provides the most directly controlled comparison. Unless otherwise stated, the codebook size is 65,536; for FSQ, we use the recommended configuration $[8, 8, 8, 5, 5, 5]$ to approximately match this capacity.

Following SimVQ, input images are resized to $128 \times 128 \times 3$, and the encoder outputs $16 \times 16 \times 128$ latent features with a downsampling factor of 8. We set the Sinkhorn-loss coefficient γ in Eq. 9 to 1.0 and train for 100 epochs on ImageNet-100, 150 epochs on FFHQ, and 6 epochs on ImageNet-1K. All experiments are conducted on a single NVIDIA RTX 5880 Ada GPU with 48 GB memory. Additional implementation details are provided in Appendix C.

5.1.3 Evaluation Metrics

We evaluate codebook usage using utilization and perplexity (Fang et al., 2025), where utilization measures the fraction of active codewords and perplexity reflects the balance of codeword usage. Since full utilization does not necessarily imply balanced or effective codeword usage, we jointly consider perplexity, quantization distortion, and reconstruction quality. Reconstruction quality is measured using rFID, LPIPS, PSNR, and SSIM, with lower rFID/LPIPS and higher PSNR/SSIM indicating better quality. We additionally report mean squared quantization error (MSQE) in the ablation studies.

5.2 Comparison With Existing Methods

Tables 1 and 2 show that AOptSA achieves full codebook utilization and the strongest reconstruction performance on both ImageNet-100 and FFHQ. Importantly, FSQ, Wasserstein-VQ, SimVQ, and

Table 1: Quantitative comparison of VQGAN-based models on ImageNet-100 in terms of codebook utilization and reconstruction performance. All images are 128 × 128 resolution, and codebook size is set to 65,536.

Method	Dim	$u(\%)\uparrow$	Perplexity ↑	rFID ↓	LPIPS ↓	PSNR ↑	SSIM(%) ↑
Vanilla VQGAN	128	1.0%	658	10.27	0.099	22.87	66.25
VQGAN-EMA	128	28.8%	17132	8.57	0.071	24.41	71.55
FSQ	6	100.0%	44643	8.82	0.071	24.49	72.09
Wasserstein-VQ	8	100.0%	**63399**	9.09	0.070	23.81	73.43
SimVQ	128	100.0%	50814	7.94	0.067	24.81	74.20
AOptSA (ours)	128	**100.0%**	61987	**7.20**	**0.063**	**25.24**	**76.23**

Table 2: Quantitative comparison of VQGAN-based models on FFHQ in terms of codebook utilization and reconstruction performance. All images are 128 × 128 resolution, and codebook size is set to 65,536.

Method	Dim	$u(\%)\uparrow$	Perplexity ↑	rFID ↓	LPIPS ↓	PSNR ↑	SSIM(%) ↑
Vanilla VQGAN	128	0.7%	331	5.35	0.052	25.10	80.30
VQGAN-EMA	128	14.8%	9067	4.55	0.034	27.11	85.50
FSQ	6	100.0%	49014	4.60	0.034	27.36	86.35
Wasserstein-VQ	8	100.0%	59068	4.52	0.028	27.30	85.73
SimVQ	128	100.0%	50956	3.95	0.030	27.85	87.44
AOptSA (ours)	128	**100.0%**	**61856**	**3.73**	**0.027**	**28.09**	**88.33**

AOptSA all reach 100% utilization, yet exhibit substantially different reconstruction quality. This indicates that eliminating codebook collapse alone is insufficient for effective quantization; how codewords are allocated in the latent space also matters.

Compared with SimVQ, Wasserstein-VQ attains higher perplexity but weaker reconstruction, possibly due in part to its low-dimensional quantization space. In contrast, SimVQ and AOptSA preserve the original 128-dimensional latent space and achieve stronger reconstruction. AOptSA further improves over SimVQ in both perplexity and reconstruction quality, suggesting that theory-guided codeword allocation remains beneficial even after full codebook utilization is achieved. We further examine the effects of alignment strategy, target distribution, codebook size, and quantization dimensionality below.

5.3 Ablation Studies and Scalability

5.3.1 Effect of Dynamic Distribution Alignment

We first examine whether distribution alignment should be applied continuously during training. Vanilla VQGAN uses no alignment, SDInitial applies Sinkhorn-based alignment only during the first epoch, and AOptSA maintains alignment throughout training. As shown in Table 3, SDInitial already improves utilization from 1.0% to 55.4%, but remains substantially below AOptSA, which reaches 100% utilization and achieves the best reconstruction metrics. Figure 2 further shows that the encoder distribution evolves throughout training. These results indicate that one-time alignment is insufficient; continuous alignment allows the codebook to track the evolving representation distribution.

5.3.2 Choice of Alignment Targets

We next compare direct matching to encoder outputs (AOptSA-Direct) with the density-reweighted target used by AOptSA. Both variants achieve 100% utilization, but AOptSA obtains higher perplexity, lower MSQE, and lower rFID (Table 4). Since utilization is identical, the improvement cannot be attributed to activating additional codewords. Instead, the density-reweighted target reduces the

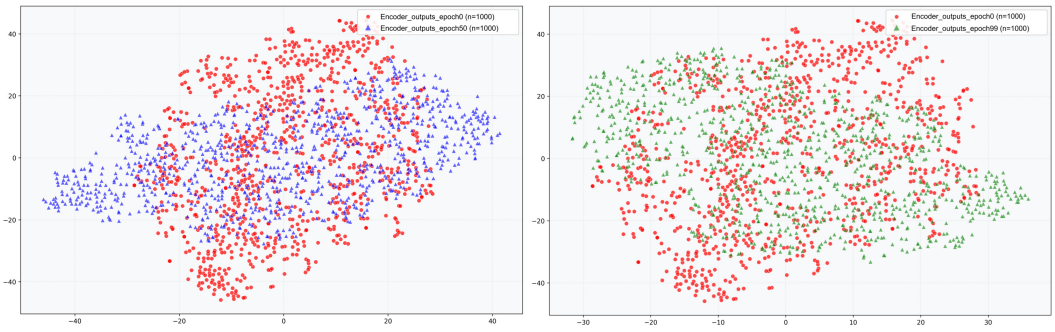


Figure 2: t-SNE visualization of the encoder-output distribution at different training epochs. Red circles denote 1,000 encoder outputs sampled at epoch 0, blue triangles denote 1,000 outputs sampled at epoch 50, and green triangles denote 1,000 outputs sampled at epoch 99. The left panel compares epochs 0 and 50, and the right panel compares epochs 0 and 99.

Table 3: Ablation study on dynamic distribution alignment. All models are trained on ImageNet-100 for 100 epochs.

Method	**Utilization** ↑	**LPIPS**↓	**PSNR** ↑	**SSIM(%)**↑
VQGAN	1.0%	0.099	22.87	66.25
SDInitial	55.4%	0.065	24.92	73.97
AOptSA	**100.0%**	**0.063**	**25.24**	**76.23**

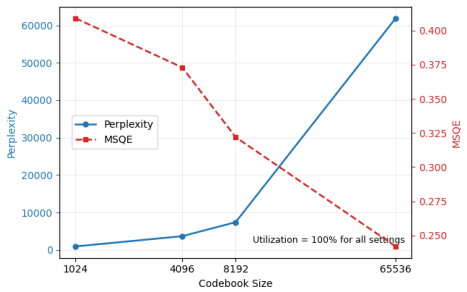


Figure 3: Effect of codebook size on quantization performance. As the codebook size increases, AOptSA consistently maintains 100% utilization, while perplexity increases and MSQE decreases, indicating better codeword usage balance and improved quantization quality.

concentration induced by the raw encoder distribution and provides better coverage of lower-density regions, resulting in lower quantization distortion and better reconstruction.

5.3.3 Effect of Codebook Size

We further evaluate the scalability of AOptSA by varying the codebook size from 1,024 to 65,536. All models are trained on ImageNet-100 for 50 epochs under the same controlled setting. We report codebook utilization, perplexity, and MSQE to assess whether larger codebooks can be effectively utilized and whether quantization error decreases accordingly. As shown in Figure 3, increasing codebook size consistently improves quantization performance (higher perplexity, lower MSQE), confirming that AOptSA can effectively exploit the extra codebook capacity rather than wasting newly introduced codewords. Crucially, this benefit continues as codebooks become very large, demonstrating the strong scalability of the proposed method.

Table 4: Ablation study on the distribution alignment target. Both variants use a 65,536-entry codebook and are trained on ImageNet-100 for 50 epochs.

Method	Perplexity ↑	MSQE↓	rFID ↓
AOptSA-Direct	60712	0.269	8.23
AOptSA	**61890**	**0.242**	**8.05**

Table 5: Ablation study of quantization dimensionality on ImageNet-1K

Method	Dim	Utilization ↑	rFID ↓
AOptSA-8	8	100.0%	3.86
AOptSA-64	64	100.0%	3.74
AOptSA-128	128	100.0%	**3.72**

5.3.4 Effect of Codebook Dimensionality

We evaluate the effect of codebook dimensionality by setting the quantization dimension to 8, 64, or 128, while fixing the codebook size to 8,192 for all variants. Starting from AOptSA-GAN, we introduce a linear projection layer that maps the 128-dimensional encoder outputs to the corresponding quantization space before vector quantization. Each variant is trained on ImageNet-1K for six epochs, and we evaluate codebook utilization and reconstruction FID (rFID). As shown in Table 5, all three variants achieve 100% codebook utilization, indicating that AOptSA remains effective in preventing codebook collapse across different quantization dimensions. Meanwhile, rFID consistently decreases as the codebook dimensionality increases, from 3.86 at 8 dimensions to 3.72 at 128 dimensions, demonstrating that higher-dimensional quantization spaces better preserve reconstruction quality.

Taken together, the ablations show that AOptSA benefits from three complementary properties: continuous adaptation to evolving encoder features, density-reweighted rather than direct distribution alignment, and effective use of large, high-dimensional codebooks.

6 Conclusion

We revisit codebook collapse in vector quantization through the lens of asymptotically optimal quantizer theory and introduce AOptSA, which aligns the codebook with a density-reweighted target distribution rather than directly matching the raw encoder distribution. AOptSA constructs this target online through importance sampling and performs continuous Sinkhorn-based alignment while preserving the original nearest-neighbor quantization rule and latent dimensionality.

Experiments on ImageNet and FFHQ show that AOptSA achieves full codebook utilization together with improved reconstruction quality. The ablation studies further show that effective quantization depends not only on activating codewords, but also on their distribution, continuous adaptation to evolving encoder features, and preservation of latent-space fidelity. AOptSA remains effective as the codebook size increases, providing a scalable and theoretically motivated approach to large-codebook vector quantization. Future work will focus on extending AOptSA to larger-scale multimodal quantization tasks.

AI use statement

In this work, we used generative AI tools to assist with language editing, manuscript organization, and improving the clarity and conciseness of the presentation. We also used these tools to provide feedback on experimental design and the presentation of experimental analyses. All methodological decisions, theoretical derivations, implementations, experiments, and scientific conclusions were independently developed, conducted, and verified by the authors. AI-generated suggestions were

critically evaluated and adopted only after author review. The authors take full responsibility for the final content of this work.

Ethics statement

This work studies vector quantization and image reconstruction using publicly available benchmark datasets. We do not collect new personal data or conduct human-subject experiments. The proposed method is intended as a general representation-learning technique, and any downstream use should follow applicable dataset licenses and responsible-use practices.

Reproducibility statement

We make several efforts to support reproducibility. The main paper provides the complete formulation of AOptSA and the experimental protocol. Detailed implementation choices, including kNN-based density estimation, importance sampling, weighted reservoir construction, and Sinkhorn alignment settings, are reported in Appendix B. Hardware and software environments, together with dataset-specific architectures and training hyperparameters, are provided in Appendix C. The datasets and evaluation metrics used in our experiments are described in Sec. 5.1. In addition, we submit the source code used for all experiments as supplementary material, together with the corresponding configuration files required to reproduce the reported results.

References

Gulcin Baykal, Melih Kandemir, and Gozde Unal. Edvae: Mitigating codebook collapse with evidential discrete variational autoencoders. *Pattern Recognition*, 156:110792, 2024. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2024.110792. URL https://www.sciencedirect.com/science/article/pii/S0031320324005430.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL https://arxiv.org/abs/1308.3432.

Sanyuan Chen, Chengyi Wang, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural codec language models are zero-shot text to speech synthesizers. *IEEE Transactions on Audio, Speech and Language Processing*, 33:705–718, 2025. doi: 10.1109/TASLPRO.2025.3530270.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017. doi: 10.1109/TPAMI.2016.2615921.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=ivCd8z8zR2. Featured Certification, Reproducibility Certification.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alaaeldin El-Nouby, Matthew J. Muckley, Karen Ullrich, Ivan Laptev, Jakob Verbeek, and Herve Jegou. Image compression with product quantized masked image modeling. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=Z2L5d9ay4B.

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.

Xianghong Fang, Litao Guo, Hengchao Chen, Yuxuan Zhang, XiaofanXia, Dingjie Song, Yexin Liu, Hao Wang, Harry Yang, Yuan Yuan, and Qiang Sun. Enhancing vector quantization with distributional matching: A theoretical and empirical study, 2025. URL https://arxiv.org/abs/2506.15078.

Xianghong Fang, Yuan Yuan, Dehan Kong, and Tim G. J. Rudner. VQ-transplant: Efficient VQ-module integration for pre-trained visual tokenizers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eETr3lrOQB.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2681–2690. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/feydy19a.html.

Keinosuke Fukunaga and L Hostetler. Optimization of k nearest neighbor density estimates. *IEEE Transactions on Information Theory*, 19(3):320–326, 1973. doi: 10.1109/TIT.1973.1055003.

Aude Genevay, Gabriel Peyre, and Marco Cuturi. Learning generative models with sinkhorn divergences. In Amos Storkey and Fernando Perez-Cruz (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1608–1617. PMLR, 09–11 Apr 2018. URL https://proceedings.mlr.press/v84/genevay18a.html.

Siegfried Graf and Harald Luschgy. *Foundations of quantization for probability distributions*. Springer Berlin, Heidelberg, Berlin, Germany, 2000.

Haohan Guo, Fenglong Xie, Xixin Wu, Frank K. Soong, and Helen Meng. Msmc-tts: Multi-stage multi-codebook vq-vae based neural tts. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1811–1824, 2023. doi: 10.1109/TASLP.2023.3272470.

Haohan Guo, Fenglong Xie, Dongchao Yang, Hui Lu, Xixin Wu, and Helen Meng. Addressing index collapse of large-codebook speech tokenizer with dual-decoding product-quantized variational auto-encoder. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 548–553, 2024. doi: 10.1109/SLT61566.2024.10832228.

Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14096–14113. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/huh23a.html.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2011. doi: 10.1109/TPAMI.2010.57.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision*, pp. 694–711, 2016.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2390–2399. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kaiser18a.html.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11523–11532, June 2022.

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.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 02 2019. doi: 10.1561/2200000073.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/ramesh21a.html.

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.

Christian P Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, New York, NY, USA, 2nd edition, 2004.

Aurko Roy, Ashish Vaswani, Arvind Neelakantan, and Niki Parmar. Theory and experiments on vector quantized autoencoders, 2018. URL https://arxiv.org/abs/1805.11063.

Casper Kaae Sønderby, Ben Poole, and Andriy Mnih. Continuous relaxation training of discrete latent variable image models. In *Bayesian Deep Learning Workshop, NeurIPS*, volume 201, 2017.

Yuhta Takida, Takashi Shibuya, Weihsiang Liao, Chieh-Hsin Lai, Junki Ohmura, Toshimitsu Uesaka, Naoki Murata, Shusuke Takahashi, Toshiyuki Kumakura, and Yuki Mitsufuji. SQ-VAE: Variational Bayes on discrete representation with self-annealed stochastic quantization. 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. 20987–21012. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/takida22a.html.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 776–794, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58621-8.

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.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=pfNyExj7z2.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David Ross, and Lu Jiang. Language model beats diffusion - tokenizer is key to visual generation. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 765–783, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/036912a83bdbb1fd792baf6532f102d8-Paper-Conference.pdf.

Borui Zhang, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Preventing local pitfalls in vector quantization via optimal transport, 2024a. URL https://arxiv.org/abs/2412.15195.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speechtokenizer: Unified speech tokenizer for speech language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 31798–31818, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/86d1ab582afb247ccaa84bec4a7e24f7-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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22968–22977, October 2025.

A Appendix: Additional Analysis on Codebook Collapse

This section provides additional analyses to better understand how the relationship between the encoder-output and codebook distributions affects codebook utilization and quantization quality. We first conduct controlled synthetic experiments to isolate the effects of support coverage and spatial codeword allocation, and then examine whether similar behavior appears in learned representations on ImageNet-100.

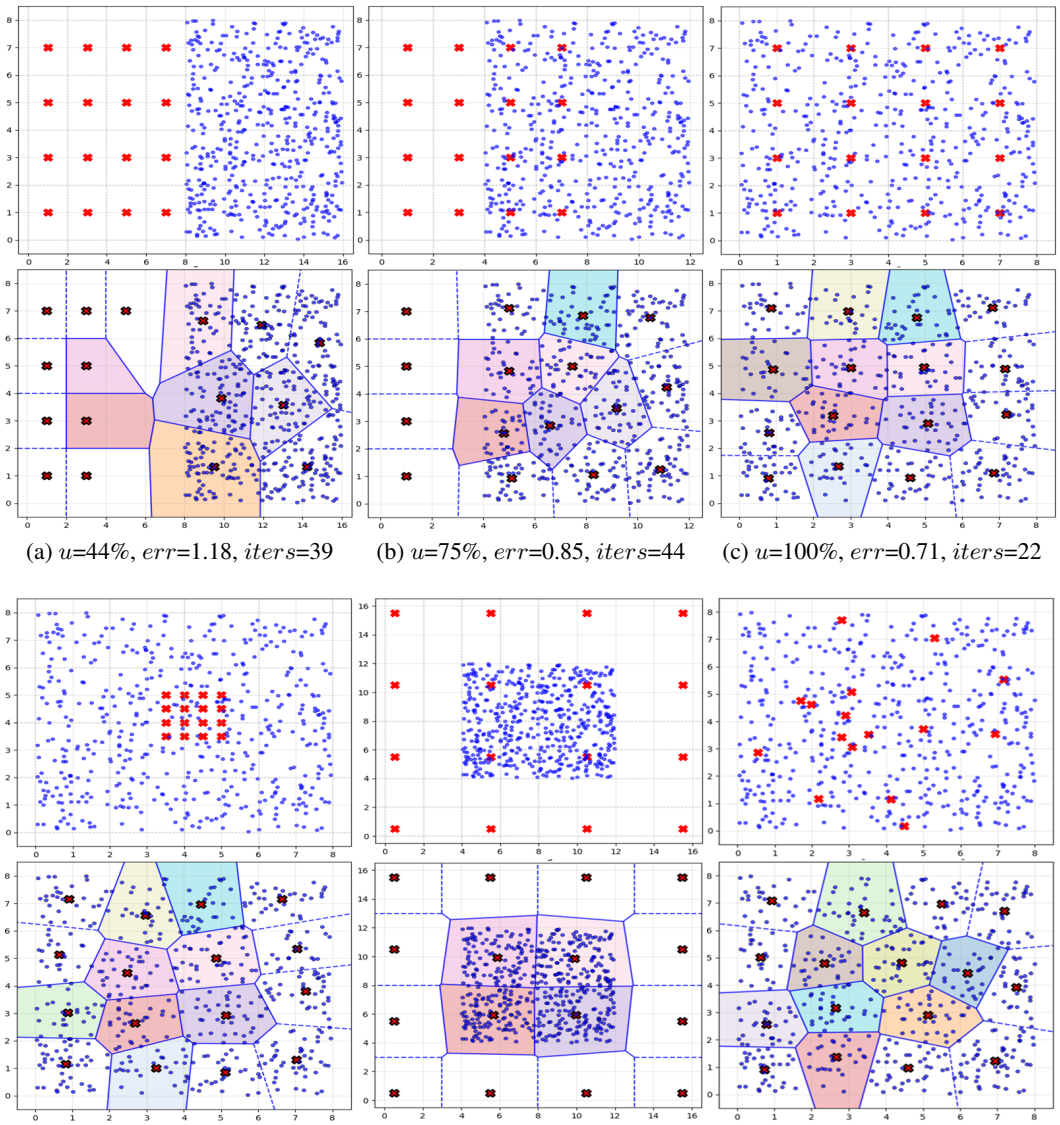
A.1 Synthetic Analysis of Distribution Matching

To isolate the effect of distribution mismatch on codebook learning, we construct a set of controlled synthetic experiments using 16 codebook vectors and 500 uniformly sampled encoder outputs. The codebook is optimized using the exponential-moving-average (EMA) k-means update. After convergence, we report codebook utilization u, mean squared quantization error (MSQE, denoted by err), and the number of optimization iterations iters. Voronoi diagrams are used to visualize the spatial partition induced by the learned codewords. Figure 4 compares six representative configurations with different relationships between the encoder-output and codebook distributions. In Figures 4(a)–(c), both distributions are uniform, while their supports are respectively disjoint, partially overlapping, and well aligned. As the overlap increases, codebook utilization improves, quantization error decreases, and optimization converges more rapidly. In particular, the well-aligned setting in Figure 4(c) achieves full utilization and the lowest MSQE among these three cases.

Figures 4(d) and 4(e) further examine the effect of support scale. Although the encoder-output and codebook distributions share a common center, their support radii differ. A support mismatch can either leave parts of the encoder distribution insufficiently covered or place codewords outside the effective feature support. Both cases lead to inefficient codeword allocation, manifested as increased quantization distortion, slower optimization, or reduced utilization.

Finally, Figure 4(f) considers a setting in which the two supports are broadly aligned but their distributional shapes differ: the encoder outputs are uniformly distributed, whereas the codewords follow a Gaussian-like allocation. Despite substantial support overlap, this mismatch in spatial density slows convergence, indicating that support coverage alone is not sufficient for efficient quantization.

Taken together, these controlled experiments highlight two complementary factors for stable codebook learning: (i) *coverage*, whereby the codebook should adequately cover the effective support of the encoder distribution; and (ii) *allocation*, whereby codewords should be appropriately distributed within that support. When both conditions are better satisfied, codebook utilization increases, quantization error decreases, and optimization becomes more efficient.



(a) u=44%, err=1.18, $iters$=39 (b) u=75%, err=0.85, $iters$=44 (c) u=100%, err=0.71, $iters$=22

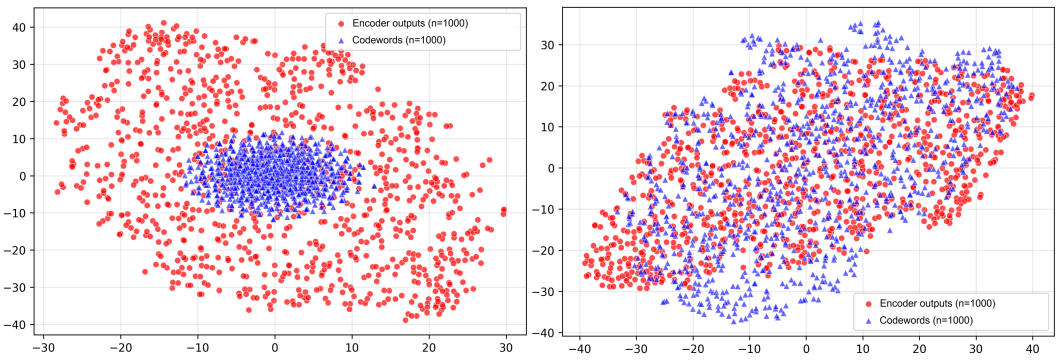
(d) u=100%, err=0.72, $iters$=52 (e) u=25%, err=1.54, $iters$=17 (f) u=100%, err=0.72, $iters$=64

Figure 4: Effects of distribution mismatch on codebook utilization, quantization error, and convergence in controlled synthetic experiments. In each panel, the upper plot shows the initial codewords (red crosses) and encoder outputs (blue dots), while the lower plot shows the EMA k-means-updated codewords (black crosses) and their Voronoi cells. Panels (a)–(c) show disjoint, partially overlapping, and aligned uniform supports; (d) and (e) show codebook supports that are respectively narrower and wider than the encoder-output support; and (f) shows a mismatch between Gaussian codewords and uniform encoder outputs. u, err and $iters$ denote codebook utilization, mean squared quantization error, and convergence iterations.

A.2 Visualization on Real Encoder Features

We next examine whether the observations from the synthetic experiments also appear in learned representations. Figure 5 visualizes encoder outputs and codewords on ImageNet-100 after 100 epochs of training using t-SNE.

For Vanilla VQGAN, the t-SNE visualization shows that the sampled codewords are concentrated in a relatively compact region compared with the sampled encoder outputs. In contrast, the AOptSA



(a) Encoder outputs and codewords in VQGAN (b) Encoder outputs and codewords in AOptSA

Figure 5: t-SNE visualization of encoder outputs and codewords after training Vanilla VQGAN and AOptSA for 100 epochs on ImageNet-100. For each model, 1,000 encoder outputs and 1,000 codewords are randomly sampled.

visualization exhibits a broader spatial distribution of codewords and greater apparent overlap with the encoder features. While t-SNE does not preserve the global geometry or density of the original high-dimensional space, these qualitative patterns are consistent with the higher codebook utilization achieved by AOptSA.

This visualization provides qualitative evidence consistent with the synthetic analysis: effective codebook learning appears to depend not only on activating more codewords, but also on improving their spatial organization relative to the encoder distribution. This observation is consistent with the motivation of AOptSA, which continuously aligns the codebook toward a density-reweighted target distribution during training.

B Appendix: Implementation Details of AOptSA

This section provides implementation details of the online target construction and Sinkhorn-based distribution alignment used in AOptSA. At each training iteration, detached encoder features are used to update an unweighted reference buffer for density estimation and a separate importance-weighted target reservoir. The former is used to estimate the current encoder-output density, while the latter maintains samples that approximate the density-reweighted target distribution. The codebook vectors are then aligned with samples from the target reservoir through Sinkhorn divergence. Gradients are stopped through both buffer updates, density estimation, importance weighting, and reservoir sampling, and the alignment loss updates only the codebook vectors.

B.1 Importance Sampling Procedure

Let a training batch produce spatial encoder features of size $B \times H \times W \times d$. We flatten them into $N = BHW$ feature vectors $\{z_i\}_{i=1}^{N}$ and detach them from the computational graph. AOptSA maintains two separate buffers with different purposes: an unweighted reference buffer for estimating the encoder distribution P_z, and a weighted target reservoir for constructing samples that approximate the density-reweighted target distribution P_r.

Reference buffer and density estimation. The reference buffer stores detached encoder features without importance reweighting and is used only for estimating the density of the current encoder-output distribution. For each feature z_i, we randomly sample at most 1,000 features from the reference buffer as reference points for a k-nearest-neighbor (kNN) density estimator. Let $R_{i,k}$ denote the Euclidean distance from z_i to its k-th nearest reference feature. The local density is estimated up to a proportionality constant as

$$\hat{h}_z(z_i) \propto \frac{1}{R_{i,k}^d}, \tag{10}$$

where d denotes the latent dimensionality. We use $k = 50$ for ImageNet and $k = 25$ for FFHQ. Because the reference buffer contains unweighted encoder features, this estimator reflects the raw encoder-output distribution rather than the density-reweighted target distribution.

Importance weighting. According to the asymptotic target distribution motivated in Sec. 3.2, the desired target density satisfies

$$h_r(z) \propto h_z(z)^{\frac{d}{d+r}}, \tag{11}$$

where r is the distortion order. Using the encoder distribution as the proposal distribution, the corresponding importance weight for z_i is

$$\tilde{w}_i = \frac{h_r(z_i)}{h_z(z_i)} \propto \left[\hat{h}_z(z_i)\right]^{-\frac{r}{d+r}}. \tag{12}$$

Thus, features from lower-density regions receive relatively larger importance weights, increasing their probability of being retained in the target reservoir.

Weighted target reservoir. Separately from the reference buffer, we maintain an epoch-wise weighted reservoir with capacity K, equal to the codebook size. The reservoir is updated after every training batch and reset at the beginning of each epoch. It stores target samples according to the importance weights above and is used to construct the empirical approximation of P_r.

We use Efraimidis–Spirakis weighted reservoir sampling. For each candidate feature z_i with positive sampling weight w_i, we draw $u_i \sim \mathcal{U}(0, 1)$ and assign the random key

$$\kappa_i = u_i^{1/w_i}. \tag{13}$$

The target reservoir retains the K samples with the largest keys. This procedure accumulates target samples online while favoring features according to their importance weights. The separation between the unweighted reference buffer and the weighted target reservoir ensures that density estimation is performed with respect to the encoder-output distribution P_z, while the second buffer is used exclusively to approximate the density-reweighted target distribution P_r.

Since encoder features are detached before either buffer is updated, no gradients are propagated through the reference-buffer update, kNN density estimation, importance-weight computation, or weighted reservoir sampling.

B.2 Sinkhorn Divergence

Let $Q = \{e_k\}_{k=1}^K$ denote the codebook and $Z' = \{z'_i\}_{i=1}^K$ the sampled target set maintained by the weighted reservoir. We align their empirical distributions using the balanced, debiased Sinkhorn divergence implemented in `GeomLoss`.

We use the quadratic Euclidean ground cost

$$c(e, z) = \frac{1}{2}\|e - z\|_2^2, \tag{14}$$

which is consistent with the squared-error distortion considered in the asymptotic quantization analysis and with the Euclidean nearest-neighbor assignment used by the underlying VQ model.

In `GeomLoss`, we set `p=2`, `debias=True`, and `reach=None`. The blur parameter is set to 0.8, corresponding to an entropic regularization strength of $\epsilon = 0.64$, and `scaling=0.5` is used to balance numerical accuracy and computational cost. Because both empirical measures contain K support points, we use the memory-efficient online backend rather than explicitly materializing the full pairwise cost matrix.

The Sinkhorn alignment loss is evaluated every ten training steps. The sampled target features are treated as stop-gradient targets, and gradients from the alignment objective are propagated only to the codebook vectors. Therefore, AOptSA regularizes the spatial organization of the codebook without modifying the encoder through the auxiliary alignment objective.

Table 6: Hardware and software environment

Configuration	Values
GPU	1 × NVIDIA RTX 5880 Ada GPU
GPU memory	48 GB
Python version	3.13.5
PyTorch	2.6.0+cu124
CUDA	12.4
cuDNN	9.1.0
Lightning	2.2.5
GeomLoss	0.2.6

Table 7: Experimental configurations on image datasets. The shared Sinkhorn implementation settings are provided in Appendix B.2.

Configuration	ImageNet-1K	ImageNet-100	FFHQ
Input size	$128 \times 128 \times 3$	$128 \times 128 \times 3$	$128 \times 128 \times 3$
Latent size	$16 \times 16 \times 128$	$16 \times 16 \times 128$	$16 \times 16 \times 128$
Quantization dimensions	8/64/128	128	128
Batch size	64	64	64
Training epochs	6	100	150
Optimizer	AdamW	AdamW	AdamW
Learning rate	1e-4	1e-4	1e-4
Warmup epochs	0	0	0
Commitment coefficient (β)	0.25	0.25	0.25
Adversarial coefficient (λ)	1.0	1.0	1.0
Sinkhorn divergence coefficient (γ)	1.0	1.0	1.0
Number of kNN neighbors (k)	50	50	25

C Experimental Details

This section provides the hardware and software environment, together with dataset-specific training configurations used in our experiments. The Sinkhorn-divergence settings shared across datasets are described separately in Appendix B.2.

C.1 Hardware and Software Environment

All experiments are conducted on a single NVIDIA RTX 5880 Ada GPU with 48 GB of memory. The software environment is summarized in Table 6.

C.2 Dataset-Specific Training Configurations

Table 7 summarizes the training configurations used for ImageNet-1K, ImageNet-100, and FFHQ. All datasets use input resolution $128 \times 128 \times 3$, latent feature size $16 \times 16 \times 128$, batch size 64, and AdamW optimization with a learning rate of 10^{-4}. The commitment-loss coefficient β, adversarial-loss coefficient λ, and Sinkhorn-alignment coefficient γ are fixed to 0.25, 1.0, and 1.0, respectively.

For ImageNet-1K, we vary the quantization dimension among $\{8, 64, 128\}$ for the dimensionality study, while ImageNet-100 and FFHQ use 128-dimensional quantization. ImageNet-1K, ImageNet-100, and FFHQ are trained for 6, 100, and 150 epochs, respectively. For kNN-based density estimation, we use $k = 50$ on ImageNet-1K and ImageNet-100 and $k = 25$ on FFHQ.

Table 8: Runtime and peak GPU memory comparison on ImageNet-100. Peak memory is measured under the same training configuration.

Method	Total Time (h)	Time/Epoch (h)	Peak Memory (GB)
SimVQ	189.94	1.899	34.868
AOptSA	190.67	1.907	35.025



Figure 6: Reconstructed FFHQ samples. (Top) Original inputs; (Middle) Vanilla VQGAN reconstruction; (Bottom) AOptSA (ours) reconstruction. All images are 128 × 128 resolution.

D Additional Experimental Results

This section provides additional experimental results that complement the quantitative evaluations in the main paper. We first report the computational overhead of AOptSA relative to SimVQ and then present qualitative reconstruction results on FFHQ and ImageNet-100.

D.1 Runtime and Memory Analysis

To evaluate the computational overhead of AOptSA, we compare its end-to-end training cost with SimVQ on ImageNet-100. Both methods use the same encoder–decoder backbone, hardware environment, batch size, and training configuration, and are trained for 100 epochs. Their quantization modules differ: SimVQ employs a learnable linear reparameterization, whereas AOptSA introduces online density estimation, weighted reservoir sampling, and Sinkhorn-based distribution alignment. We therefore report both wall-clock training time and peak allocated GPU memory as end-to-end measurements of the two methods rather than as isolated costs of individual components.

As shown in Table 8, SimVQ requires 189.94 hours in total, corresponding to 1.899 hours per epoch, while AOptSA requires 190.67 hours in total, or 1.907 hours per epoch. Thus, AOptSA introduces only 0.73 additional training hours over 100 epochs. Its peak allocated GPU memory is 35.025 GB, compared with 34.868 GB for SimVQ, corresponding to an increase of only 0.157 GB (approximately 0.45%). These results indicate that AOptSA introduces only a small additional computational and memory overhead in the evaluated setting.

D.2 Qualitative Reconstruction Results

We further provide qualitative reconstruction results on FFHQ and ImageNet-100. Figures 6 and 7 compare the original images with reconstructions produced by Vanilla VQGAN and AOptSA. All images are evaluated at a resolution of 128×128.

These visualizations complement the quantitative reconstruction metrics reported in the main paper and provide qualitative examples of the reconstruction behavior of AOptSA across both face-domain and object-centric datasets.



Figure 7: Reconstructed ImageNet-100 samples. (Top) Original inputs; (Middle) Vanilla VQGAN reconstruction; (Bottom) AOptSA (ours) reconstruction. All images are 128 × 128 resolution.