IVQ: Structured and Lightweight Vector Quantization via Binary Hierarchical Composition Inspired by *IChing*

Heda Zuo [1] **Junxian Wu** [1] **Fengjie Lu** [1] **Pei Chen** [1] **Lingyun Sun** [1] **Weitao You** [1]

Abstract

Vector Quantization (VQ) has been widely used in visual and audio representation due to its effectiveness in compressing high-dimensional signals. However, existing VQ methods often rely on large and unstructured codebooks, which leads to inefficient code utilization and frequent codebook collapse. In this paper, we propose *IChing* Vector Quantization (IVQ), a lightweight and structured VQ framework inspired by *IChing*. IVQ introduces binary hierarchical composition and geometric symmetry relations into the codebook design, enabling a compact set of structured codes to represent the latent space while maintaining high utilization without codebook collapse. Experimental results show that IVQ achieves superior quality with significantly smaller codebooks and consistently higher utilization rates compared to several VQ variants in audio representation. Auxiliary experiments on visual reconstruction and cross-modal generation further validate the universality and robustness of IVQ. Codes are released at https://github.com/chouliuzuo/IVQ.

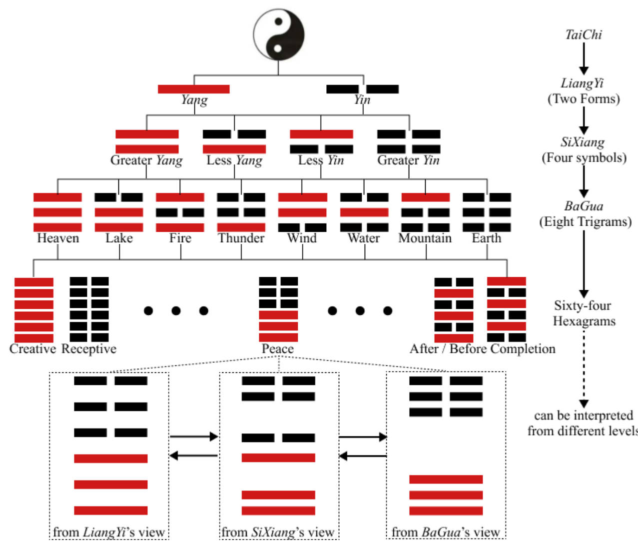

Figure 1. The core concept of *IChing*. *TaiChi* produce *LiangYi*, *LiangYi* produce *SiXiang*, *SiXiang* produce *BaGua*, which again produce the sixty-four hexagrams. For each hexagram, it can also interpreted from views of different levels, whether from top to down or from bottom to up.

1. Introduction

Vector Quantization (VQ) (Gray, 1984; Buzo et al., 1980) has long been a fundamental technique in signal processing, enabling the compression of signals while preserving high fidelity. It is particularly effective for data (e.g. video and music) with high spatial or temporal complexity by reducing the redundancy. In typical VQ, the latent space is discretized into a codebook of codewords, each representing a cluster of nearby latent vectors.

However, VQ approaches (Van Den Oord et al., 2017; Yu et al., 2021) and the variants (e.g., RVQ (Kumar et al., 2023), PVQ (Jegou et al., 2010)) suffer from a fundamental limitation: the codebook is usually treated as **an unstructured set of independent vectors**, where training losses focus mainly on optimizing the gap between embeddings and codes. This lack of intrinsic topology often leads to codebook collapse, where a significant portion of codes (“**dead codes**”) are never activated due to poor initialization and vanishing gradients. Tricks like random initialization or restarting to a batch value can increase the probability of being activated, but cannot fundamentally solve the problem and may lead to **redundancy** (The representation range of code vectors exhibit a high degree of overlap). Consequently, to ensure expressivity with the rise of large-scale models, existing methods (Défossez et al., 2022; Yu et al., 2023) often simply enlarge the codebook size instead of optimizing its internal structure, incurring high computational costs. Therefore, some downstream researches are limited to rely on pre-trained large backbones rather than task-specific re-training, restricting the potential for fine-grained adaptation.

We argue that an ideal codebook should not be an arbitrary

[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China. Correspondence to: Weitao You <weitao_you@zju.edu.cn>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

collection of codes but a structured system with logical dependencies. To achieve this, we draw inspiration from the binary quantization system of *IChing* (*The Book of Changes*), and propose an ultra lightweight and structured codebook that improves the efficiency of VQ through binary hierarchical composition and geometric relational constraints. Far from being merely a philosophical text, *IChing* represents one of the earliest quantization systems that discritizes complex phenomena into symbolic representations based on binary codes of *Yin* and *Yang*. Its underlying principles offer several insights that naturally align with the goals of VQ:

1. Binary Hierarchical Composition: *IChing* discritizes diverse phenomena based on a simple binary system. Moreover, it expands from the binary *Yin–Yang* foundation into Four Images, Eight Trigrams, Sixty-Four Hexagrams through overlapping. This hierarchical organization inspires a layered codebook structure, where each code can be made up from different granularity of base codes (shown in Figure 1, a hexgram can be composed of six-overlaps of *LiangYi*, tri-overlaps of *SiXiang* or overlap of *BaGua*, which is similar to how the decimal number 12 can be represented as 1100 in binary, 30 in quaternary, and 14 in octal). Such hierarchy allows a small set of base codes to compose a large codebook containing multi-level information, significantly reducing the complexity and risk of collapse.

2. Geometric Symmetry Relations: In *IChing*, hexagrams are not independent codes but are connected through geometric relationships, such as inverted, opposite, and contrapositive forms, which resemble logical Converse, Inverse, and Contrapositive relations. These structured relationships inspire the geometric relational constraints among codes, if codes are abstractly treated as geometric figures. By enforcing these structures connections, IVQ mitigates codebook collapse problem since codes can receive gradient through their related codes beside themselves.

Based on the above insights, we propose a hierarchical and structured quantization scheme-*IChing* Vector Quantization (IVQ)-for representation. For example, each quantization layer represents a different semantic granularity of hexagrams: 64-Hexagram, 8-Trigram × 2 composition, 4-Image × 3 composition, and 2-*YinYang* × 6 composition. The complete codebook can thus be constructed with only 104 codes (64 + 8×2 + 4×3 + 2×6). We could further compact to 78 codes (64 + 8 + 4 + 2) when keeping a shared subspace in each composition, and finally, a four-layer design based solely on the *Yin–Yang* requires merely 8 codes (2×4) to represent the whole hidden semantic space.

Because each hexagram in *IChing* is unique and consistent, we encourage a hierarchical consistency loss to maintain alignment among different quantization granularities. Furthermore, leveraging the relational structure of the hexagrams, we design a relational consistency loss according to geometric symmetry relations among corresponding codes. These losses collectively ensure the structural integrity of the IVQ codebook and **allow gradient propagation between codes even when not activated**, so that keep a full utilization which effectively prevent codebook collapse.

Furthermore, we can also apply the IVQ codebook to visual encoding processes rather than relying on pretrained large models. These IVQ encoders can be used for visual and music reconstruction in single modal. Moreover, IVQ also contributes to cross-modal downstream task like video-to-music generation. We first discretize the video inputs into *IChing* codes, which are then directly dequantized within the music codebook through an autoregressive generation process to produce the corresponding music.

Experimental results show that IVQ preserves representation quality in both audio and visual domains while substantially reducing the complexity of codebook training. Owing to its compact and structured design, we further observe that a single IVQ codebook can be shared across modalities, enabling discrete codes quantized from one modality to be dequantized in another. Moreover, despite introducing no additional task-specific modifications, IVQ-based models even achieve competitive performance on the video-to-music generation task. These findings provide preliminary evidence of IVQ in learning efficient and universal multi-modal representations.

Our main contributions are summarized as follows: 1) Inspired by *IChing*, we propose IVQ, a Residual-Product Quantization structure based on binary logic, which achieves a breakthrough by introducing an ultra-compact and structured codebook. 2) To prevent codebook collapse, we propose hierarchical and relational consistency loss according to binary hierarchical composition and geometric symmetry relations. 3) Extensive experiments demonstrate that IVQ can be comparable or even superior to existing baselines in audio representation using a much more compact codebook, and the versatile applicability to visual and cross-modal domains is also preliminarily confirmed.

2. Related Work

2.1. Vector Quantization

Vector Quantization (VQ) was originally introduced in (Buzo et al., 1980; Gray, 1984) as a cornerstone technique for compressing complex signals while preserving fidelity. It has been widely adopted across modalities, including images (Van Den Oord et al., 2017; Esser et al., 2021), videos (Zhang et al., 2024), and audio (Copet et al., 2024; Agostinelli et al., 2023). The core idea of VQ is to discretize continuous latent spaces into compact codebooks, then use several center codes to represent for nearby vectors, allowing efficient storage and representation. Subsequent improvements have yielded two major branches: parallel quantization like Product VQ (Jegou et al., 2010) and sequential quantization like Additive VQ (Babenko & Lempitsky, 2014) and Residual VQ (Chen et al., 2010), which respectively focus on subspace partitioning and refinement. To mitigate the non-differentiability and gradient collapse inherent in VQ (Huh et al., 2023), several differentiable variants have been developed, such as Soft Convex VQ (Gautam et al., 2024), EMA (Łańcucki et al., 2020), and Noise Substitution VQ (Vali & Bäckström, 2022). More recently, VQ has also been optimized by simplification and acceleration (Mentzer et al., 2023; Yu et al., 2023). Although VQ is widely used in large models for complex signal processing, it still suffers from unstructured codebooks with low utilization rate or heavy redundancy, which leads to heavy computational costs in training.

2.2. Audio representation

The general concept of audio representation is to use an encoder that compresses high-dimensional temporal signals into a low-dimensional latent representation, which is the foundation of downstream tasks like data reconstruction and generation. In early ages, audio representation models are mainly inspired by discrete encoding models in images like VQ-VAEs (Van Den Oord et al., 2017; Razavi et al., 2019) and VQ-GANs (Esser et al., 2021; Yu et al., 2021). Recently, neural acoustic codecs (Zeghidour et al., 2021; Défossez et al., 2022; Kumar et al., 2023) have demonstrated remarkable capabilities in reconstructing high-quality audio. Then, models like WavTokenizer (Ji et al., 2024) and MuCodec (Xu et al., 2025) further enhance the compression in low-bitrate tokenization. Researches in this area (Li et al., 2024; Liu et al., 2024) is still balancing the trade-off between high sample rate and low compression bitrate while maintaining the reconstruction quality. Discrete audio tokens have also enabled large-scale generative model like (Borsos et al., 2023; Agostinelli et al., 2023). Despite these advances, some audio representation models increase capacity through larger but unstructured codebooks, which still do not address codebook collapse and redundancy issues. The lack of explicit structural constraints in codebook design limits efficient reuse and poses challenges for downstream tasks, motivating the exploration of compact and structured quantization schemes.

3. Method

3.1. Preliminary

3.1.1. Vector Quantization

A vector-quantized network (VQN) is a neural-network consisting of a VQ layer $h(\cdot,\cdot)$:

$$\hat{y} = D(h(E(x), C)) = D(h(z_e, C)) = D(z_q) \quad (1)$$

The VQ layer $h(\cdot,\cdot)$ quantizes the encoded embedding $z_e = E(x)$ by selecting the nearest vector from a codebook of n vectors $C = \{c_1, c_2, ...c_n\}$. The individual vector c_i is referred to as the code-vector and the index i as the code. The process $h(\cdot,\cdot)$ can be written as:

$$z_q = c_i,\ i = \arg\min_j\ d(z_e, c_j) \quad (2)$$

Euclidean distance is the standard distance measure for $d(\cdot,\cdot)$, where $d(x, x') = \|x - x'\|_2$. Then, z_q is sent to predict the output $\hat{y}$ through the decoder, and the loss is computed with the target y through $\mathcal{L}(y, \hat{y})$:

$$\min_{E,D,h} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}}[\mathcal{L}(D(h(E(x))), y)] \quad (3)$$

The above equation is not continuously differentiable since there is an $\arg\min$ operator, the straight through estimation (STE) cancels the not-differentiable parameter and using z_e to represent z_q in the back propagation:

$$\frac{\partial\mathcal{L}}{\partial E} = \frac{\partial\mathcal{L}}{\partial\hat{y}}\frac{\partial\hat{y}}{\partial z_q}\cancel{\frac{\partial z_q}{\partial z_e}}\frac{\partial z_e}{\partial E} \approx \frac{\partial\mathcal{L}}{\partial E} \quad (4)$$

The commitment loss is also be considered which evaluate the accuracy of quantization:

$$\mathcal{L}_{commit} = (1-\beta)\ d(sg(z_e), z_q) + \beta\ d(z_e, sg(z_q)) \quad (5)$$

where β is the commitment loss weight and $sg(\cdot)$ represents the stop gradient function. It is obvious that there will be no gradient or upgrade if a c_j has never been used if the codebook is non-structured, which causes lots of dead codes with a low utilization rate, that is, **codebook collapse**.

3.1.2. Residual / Product Vector Quantization

To balance representational capacity and computational efficiency, various extensions of VQ have been developed. Residual Vector Quantization (RVQ) performs multi-stage quantization, encoding a vector through successive residual refinements using multiple codebooks:

$$z_{q,k} = h(z_r, C_k),\ k \in [1, K],\ z_r = z_r - z_{q,k} \quad (6)$$

where z_r is initialized as z_e, and $z_q = \sum_i^K z_{q,k}$ after K iterations. RVQ can represent n^K vectors with space complexity $n \times K$. It still causes an extra space and time complexity of K times, and is prone to codebook collapse when n is large. Unfortunately, as it lacks subspace decomposition, it often requires a large n to achieve comprehensive coverage of the original high-dimensional latent space.

Product Vector Quantization (PVQ), on the other hand, divides the latent vector z_e into K subspaces and applies independent quantization to each piece:

$$z = Concat_i^K\{z_k\},\ z_{q,k} = h(z_{e,k}, C_k) \quad (7)$$

Algorithm 1 *IChing* Vector Quantization (Bottom-Up)

Require: Embedding z_e, block sizes $B = \{6, 3, 2, 1\}$, codebooks $C = \{C_0, C_1, C_2, C_3\}$

$z_r \leftarrow z_e$
Initialize list $Q \leftarrow \emptyset$
for $i = 0$ to 3 **do**
  Split z_r into $B[i]$ blocks: $\{z_r^{(j)}\}_{j=1}^{B[i]}$
  for $j = 1$ to $B[i]$ **do**
    $\text{idx}^{(j)} \leftarrow \arg\min_k \|z_r^{(j)} - C_i[k]\|$
    $z_q^{(j)} \leftarrow C_i[\text{idx}^{(j)}]$
  end for
  $\hat{z}_{q,i} \leftarrow \text{Concat}(z_q^{(1)}, \ldots, z_q^{(B[i])})$
  Append $\hat{z}_{q,i}$ to Q
  $z_r \leftarrow z_r - \hat{z}_{q,i}$
end for
$z_q \leftarrow \sum_i \hat{z}_{q,i}$
output: Quantized representation z_q

PVQ can also represent n^K vectors without extra space cost, since the subspace dimension is reduced from D to D/K. However, it weakens correlations among subspaces in vector due to block separation and also increases time complexity.

Despite their limitations, RVQ and PVQ provide foundational insights into hierarchical quantization, which motivates the design of *IChing* Vector Quantization (IVQ). IVQ synergistically unifies their advantages and further introduces explicit structural relationships among codes, so that bridges the gap in RVQ's representational capacity within compact codebooks and resolves the fragmented subspace correlations inherent in PVQ. This approach not only reduces computational complexity but also robustly prevents codebook collapse while ensuring representation quality.

3.2. IChing Vector Quantization

As illustrated in Figure 1, *IChing* quantizes diverse phenomena and events into 64 discrete hexagrams, each representing a unique combination of *Yin* and *Yang*. Moreover, *IChing*'s philosophical framework embodies three intrinsic principles — **simplicity** (*JianYi*), **variability** (*BianYi*), and **invariance** (*BuYi*). Such philosophical insights can inspire a rethinking of the VQ process, offering potential solutions to long-standing challenges such as large unstructured codebooks with codebook collapse or codebook redundancy.

3.2.1. Mathematical Abstraction of *IChing*

To formalize the structural prior of IVQ, we abstract the core concepts of *IChing*—the **Binary Hierarchical Composition** and the **Geometric Symmetry Relations**—into a mathematical framework.

Firstly, we define the basic quantization unit *LiangYi* as a binary state $\mathcal{B}_1 = \{y^0, y^1\}$, where $y \in \mathbb{R}^d$ denotes a base code. Following the *IChing* hierarchy, a hexagram is constructed through a recursive binary composition process: 1) *LiangYi*: The base level distinguishes *Yin* and *Yang*. 2) *SiXiang*: Formed by the Cartesian product of two bits of *LiangYi*, $\mathcal{B}_2 = \mathcal{B}_1 \times \mathcal{B}_1$, resulting in $2^2 = 4$ codes. 3) *BaGua*: the Cartesian product of *Sixiang* and *LiangYi*, $\mathcal{B}_3 = \mathcal{B}_2 \times \mathcal{B}_1$. 4) Hexagram: A complete semantic unit $\mathcal{B} \in \mathbb{R}^D$ can be seen as composed of two *Bagua*s / three *SiXiang*s / six *LiangYi*s ($2^6 = 64$ codes). Details in Sec. 3.2.2.

In *IChing*, the relationship between codes is defined by geometric and logical symmetries. We map these to specific transformations in the latent space, providing a structural regularizer for the codebook: 1) *zong* is an index **inversion** relationship that defined as a spatial reversal, analogous to the Converse in logic. 2) *cuo* is an index **opposition** relationship that defined as a state negation like the Inverse. 3) *cuo-zong* is the **combination** of both, analogous to the Contrapositive. Details in Sec. 3.2.3.

Inspired by these concepts of *IChing*, we consider that a codebook can be generated from a minimal set of base codes through algebraic operations such as the Cartesian product. This strategy significantly reduces computational cost while empowering each code with multi-granular information, thereby preserving representational richness. Furthermore, by establishing logical or geometric dependencies between codes, the codebook space can be structured to ensure full code utilization and robustly prevent codebook collapse and redundancy common in unstructured VQs.

3.2.2. Hierarchical Codebook

In the view of binary hierarchical composition, each code vector can be manifested differently across multi-granular quantization layers as Algorithm 1. This hierarchical organization not only increases the information capacity of individual codes but also establishes consistent relationships among different layers of the codebook.

In detail, we first build a hierarchical codebook $C = \{C_1, C_2, C_3, C_4\}$ based on RVQ. The difference is that the number and dimension of code vectors are distinct in different levels of codebooks. Following *IChing*, we can set the number of each codebook as $n = \{2, 4, 8, 64\}$ from a bottom-up perspective while $n = \{64, 8, 4, 2\}$ from a top-down perspective. As a result, the dimension of each code vector is set into D/K_i with $K = \{6, 3, 2, 1\}$ pieces (bottom-up) for PVQ since $n_i^{K_i} = 64$ and $D_i \times K_i = D$.

Thus, in quantization, for each input vector z_e, it will be split into K_i pieces and each piece will be quantized with C_i as shown in Figure 2 (c). Then, the residual of each layer will be quantized in the next layer and the quantized vector

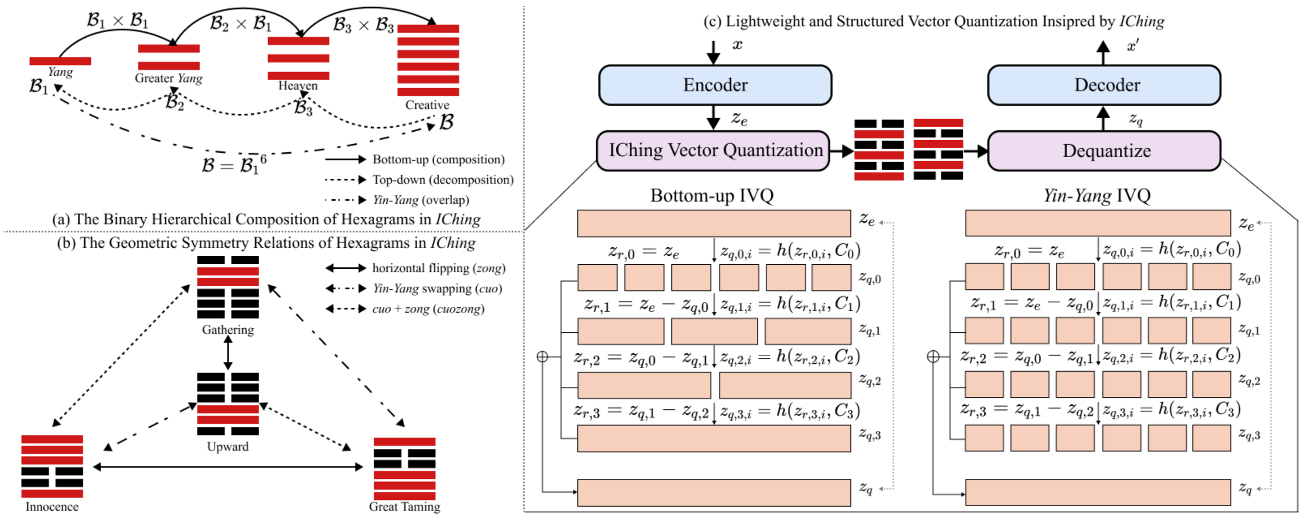

Figure 2. The main concept of Binary Hierarchical Composition and Geometric Symmetry Relations in *IChing*, which inspires the lightweight and structured vector quantization in IVQ.

z_q is the sum of output in each layer. It can be written as:

$$z_{q,i} = Concat_j^K\{h(z_{r,i,j}, C_i)\},\ z_q = \sum_{i=1}^{4} z_{q,i} \tag{8}$$

The hierarchical codebook can be regarded as a combination form of RVQ and PVQ, however, we simplify both of them under the guidance of *IChing*. For RVQ, where each quantization layer typically maintains the same dimension and codebook size, we reduce the number of codes per layer. For PVQ, where each piece employs an independent codebook, we allow all subspaces to share a unified codebook, since *Yin–Yang* remains invariant across different pieces.

We can calculate how we simplify the space complexity. For naive RVQ, the codebook size is $4 \times 64 \times D$, and for hybrid form of RVQ and PVQ, the codebook size is $64\times D+8\times D/2\times 2+4\times D/3\times 3+2\times D/6\times 6 = 78\times D$. But the codebook size for IVQ is only $64 \times D + 8 \times D/2 + 4\times D/3+2\times D/6 < 74\times D$. And we could further simplify it by keeping each layer in *Yin-Yang* and the codebook size is only $4 \times 2 \times D/6 < 2 \times D$. All of these codebooks have the capability to form 64^4 kinds of codes (4-layer).

Additionally, we build the hierarchical relationship between codebooks of different layers. Since each hexagram is the same no matter being composed from which granularity, code vectors in same hexagram index should be consistent across different granularity. Therefore, we decode 64 hexagrams from each layer of codebook and propose a hierarchical similarity loss inspired by (Radford et al., 2021):

$$\mathcal{L}_{C_\alpha \to C_\beta} = -\sum_{i}^{N} \left[\frac{s(c_\alpha^i, c_\beta^i)}{\sum_j^N s(c_\alpha^i, c_\beta^j)} \right] \tag{9}$$

where $s(c_\alpha, c_\beta) = \frac{c_\alpha^T c_\beta}{\|c_\alpha\|\cdot\|c_\beta\|}$ and c_α, c_β are the hexagram code vectors from codebook $C\alpha, C_\beta$. Thus, we ensure the unity of hexagrams between codebooks of different layers.

3.2.3. Structured Codebook

Following the geometric symmetry relations of *IChing*, we introduce intra-codebook relational structures among codes within each layer. This design primarily aims to prevent codebook collapse and redundancy. When relational dependent codes are activated, a code that is not directly activated can still receive gradient updates through its associated codes during training, preventing it from becoming a "dead code." Meanwhile, the structured relationships can also ensure a disjoint representation range of code vectors.

In this paper, we focus on the relations among opposite and inverted (*cuo*/*zong*) hexagrams in *IChing* shown in Fig. 2 (b). Specifically, a *zong* hexagram is obtained by flipping a hexagram horizontally, thereby reversing the order of lines; a *cuo* hexagram is generated by negating each line, switching *Yin* to *Yang* and vice versa. The *cuo–zong* hexagram is formed by first applying negation (*cuo*) and then flipping (*zong*). Interestingly, the relationships among hexagrams resemble those between a logical proposition and its converse, inverse, and contrapositive in mathematics.

Regarding the mathematical formulation, the *zong* counterpart of index $i = \{b_1, b_2, \cdots, b_6\}, b_j \in \mathcal{B}_1$ can be written as $\mathcal{T}_{zong}(i) = \{b_6, b_5, \cdots, b_1\}$, while *cuo* counterpart can be written as $\mathcal{T}_{cuo}(i) = \{\neg b_1, \neg b_2, \cdots, \neg b_6\}$. Therefore, the combination of *cuo-zong* counterpart can be written as $\mathcal{T}_{cuozong}(i) = \{\neg b_6, \neg b_5, \cdots, \neg b_1\}$. We design relation losses like Equation 9 inspired by the InfoNCE loss. For *zong* relations, we encourage each pair of code vectors corresponding to mutually flipped hexagrams to maintain reversal consistency, that is, $c_{f(i)} = f(c_i)$ where f stands for flipping function. For *cuo* relations, we encourage corresponding code vectors to be maximally dissimilar due to their opposite states. Finally, for *cuo–zong* relations, we encourage their code vectors to be more similar which is analogous to contrapositive pairs. If we denote Equation 9 as $g(\cdot, \cdot)$, then the losses can be written as:

$$\mathcal{L}_{zong} = g(f(c_i), c_{f(i)}) \tag{10}$$

$$\mathcal{L}_{cuo} = -g(c_i, c_{\mathcal{T}_{cuo}(i)}) \tag{11}$$

$$\mathcal{L}_{cuozong} = g(c_i, c_{\mathcal{T}_{cuozong}(i)}) \tag{12}$$

Therefore, the 64 hexagrams are further organized into 20 relational groups, where all codes within the same group are jointly updated whenever any one of them receives a gradient. In practice, since the number of codes in each hierarchical layer (from 2 to 64) is relatively small, this group-wise update mechanism further mitigates the risk of codebook collapse and redundancy. Moreover, the internal structured relationships establish logical consistency within the codebook, reinforcing both training stability and interpretability.

3.2.4. The Expansibility of IVQ

IChing's philosophical framework embodies three intrinsic principles — **simplicity** (*JianYi*), **variability** (*BianYi*), and **invariance** (*BuYi*). The principle of **simplicity** is straightforward, as the entire quantization is constructed by expansion of the binary duality (*Yin-Yang*). The true power of *IChing*, however, lies in its **variability**, which inspires the expansibility of this VQ structure. Among its variability, the aspect of **invariance** is embodied by the hierarchical and structured codebook design, which remains stable and interpretable.

We first discuss the expansibility across modalities. As described in *IChing*, the 64 hexagrams constitute a "universal compact discrete set" to represent vast worldly phenomena. For example, the *Qian* hexagram does not merely label a single object but serves as a high-dimensional abstraction for shared attributes "primacy" across domains like heaven, sovereignty, or leadership. This principle motivates the design of IVQ as a shared discrete latent space that captures high-dimensional information in a modality-agnostic manner. By encoding signals into a highly condensed, binary-composed latent space, the model strips away modality-specific noise and retains only the core structural information. It allows a shared codebook structure across modalities, where quantization is modality-agnostic and dequantization is modality-specific, enabling efficient cross-modal extensibility without modifying the quantization structure.

Beyond the standard 64-state configuration, IVQ offers a highly flexible and reconfigurable framework for diverse dimensionality and precision requirements. The hierarchical composition is not restricted to a fixed depth or binary radix, both the number of layers and the base code of each layer can be extended while preserving the underlying compositional structure. For example, by overlapping two sets of 64 hexagrams, 4,096 extended hexagrams composed of 12 lines can be obtained, where the *Yin–Yang* layer only requires a spatial complexity of $2 \times D/12$. For higher layers, the code number n_i can flexibly take values such as 4, 8 or 16. Given the same latent dimension D, if each layer uses a small codebook size n, the reduction in computational complexity of IVQ becomes more significant as the total number of codes increases.

3.3. Implementation and Task Framework

To evaluate the efficiency and universality of the proposed IVQ, we integrate it into standard encoder–decoder architectures mainly in audio and further across different modalities since we consider it as a task-agnostic quantization module.

Audio Representation. Our primary evaluation is conducted on a neural audio codec framework similar to Encodec (Défossez et al., 2022), which adopts a convolutional encoder, an IVQ module, and a symmetric decoder. The encoder maps input waveforms into continuous latent features z_e, which are discretized to z_q by IVQ module. The decoder reconstructs the audio signal from the quantized representations. This architecture serves as the primary instantiation for evaluating IVQ and is used consistently across all audio experiments. Details are shown in Appendix B.

Extensions to Other Modalities. To verify the universality and robustness of IVQ, we further apply the same quantization module to visual reconstruction and video-to-music generation. In these settings, we simply replace the encoder and decoder with modality-specific networks, while keeping the IVQ module unchanged. For visual reconstruction, we deploy IVQ within VQ-VAE. Following (Yu et al., 2024), the IVQ layer quantizes the 2D feature maps from a Vision Transformer (ViT) or a ResNet backbone. Finally, we also explore the potential of IVQ in bridging heterogeneous modalities by combining a visual encoder and a music decoder with IVQ structure for video-to-music generation.

Video-to-music generation. In IVQV2M, we treat each video V as a sequence of images x with frame rate fr. We obtain the hexagrams (IVQ codes) of x after quantization in visual encoder like $i = \arg\min(x, C_v)$. The hexagrams are dequantized in music codebook since IVQ is unified for multimodals like $z_m = h_m^{-1}(i, C_m)$. Then it will be treated as a sequence of music tokens without cross-modal transformation. The music tokens are generated through a decoder-only model with FeedForward Transfomer blocks (shifted-right music tokens and dequantized latent z_m as inputs), and then reconstructed to audio format through the music decoder as shown in Fig. 4 in Appendix.

Table 1. Evaluation of quantization comparison (KLD: Kullback–Leibler Divergence, LSD: Log-Spectral Distance, PSNR: Peak Signal-to-Noise Ratio, SSIM: Structural Similarity Index Measure, CS: Chroma Similarity.)

Model	Codebook Size	MTG dataset					LibriSpeech dataset				
		KLD↓	LSD↓	PSNR↑	SSIM↑	CS↑	KLD↓	LSD↓	PSNR↑	SSIM↑	CS↑
RVQ	4×64	0.50	1.58	23.79	0.53	0.71	0.31	3.30	23.05	0.54	0.54
PVQ	4×64	0.42	1.57	23.84	0.55	0.76	0.27	**3.22**	23.09	0.56	0.55
VQ	1×4096	0.76	2.19	16.61	0.21	0.09	0.32	3.76	14.09	0.10	0.07
FSQ	1×4375	2.32	2.71	15.47	0.10	0.09	3.27	4.23	14.15	0.06	0.09
LFQ	1×4096	0.78	1.62	22.87	0.44	0.64	0.37	3.46	21.67	0.45	0.41
R-FSQ	4×4375	2.44	2.65	15.12	0.10	0.14	2.15	3.89	14.83	0.07	0.09
R-LFQ	4×64	0.61	1.63	22.60	0.44	0.61	0.25	3.34	21.28	0.44	0.31
IVQ	**4×2**	**0.37**	**1.54**	**24.21**	**0.56**	**0.77**	**0.22**	3.42	**23.47**	**0.57**	**0.57**

4. Experiments

4.1. Implementation details

In audio encoder, we use 32 channels for embedding and 4 CNN blocks with (2,4,5,8) strides. The kernel size is 3 for ResNet while 7 for input and output. 2-layer LSTM is used for sequential modeling and a Conv1D layer for encoding output following (Défossez et al., 2022). We adopt *Yin-Yang* IVQ in quantization, and the structure of decoder is in the reverse order of the encoder, using transposed convolutions instead of strided convolutions. Audio sample rate is 50 tokens per second for 32khz while 75 for 48khz. The audio model is trained on MTG dataset (Bogdanov et al., 2019) on 1×RTX5090 for 150 epoches with a batch size of 48.

For extensions, we adopt 12 layers of residual attention blocks with 768 as hidden dimension in visual encoder. The patch size is 16 for each 256×256 image which will be quantized into 64 tokens with Bottom-up IVQ. The visual model is trained on 4×RTX4090 for 750k steps with a batch size of 32. For video-to-music generation, video sample rate is 1 frame per second and 24 layers with 1024 dimension in the feedforward transformer. It is trained on 1×RTX5090 for 120 epoches with a batch size of 12.

4.2. Metrics and Baselines

Metrics. We employ a broad range of metrics that assess visual and music reconstruction and video-to-music generation. For visual reconstruction, we use PSNR and SSIM, which measure signal-level fidelity, and FID which computes the distributional distance between generated and real data, while IS reflects the sharpness and diversity of the outputs. For music reconstruction, we further incorporate audio-specific similarity measure: FAD and FD quantify feature-space distances between generated and reference music, LSD measures spectral reconstruction error, and CS evaluates pitch-related consistency. For video-to-music generation, multimodal metrics are used, including CMR for overall audio–visual semantic consistency, TA for the synchronization of temporal events. For subjective evaluation, OMQ evaluates the perceptual music quality, while MVC evaluates the global cross-modal correlation. Moreover, we add the Codebook Usage (CU) to evaluate the risk of codebook collapse. We conduct the experiment in both MTG and LibriSpeech test set with 100 samples for each.

Baselines. We adopt RVQ (Chen et al., 2010), PVQ (Jegou et al., 2010), FSQ (Mentzer et al., 2023), LFQ (Yu et al., 2023) and variants like Residual-FSQ and R-LFQ in the same training configuration with the same Encodec (Défossez et al., 2022) framework for quantization comparison (This framework has already included the strategy of restarting dead codes to a value from the batch). For application comparison, we test Encodec, WavTokenizer (Ji et al., 2024) and MuCodec (Xu et al., 2025) in the MTG test set with a same training process using their original configurations. These application models are all based on VQ which can potentially be adapted to IVQ.

4.3. Experimental Results

Quantization Comparison. Table 1 shows that: 1) Naive VQ methods tend to rely on large codebooks to achieve reasonable reconstruction quality, whereas IVQ attains superior performance using an ultra-compact codebook of size 4×2. This is because of many dead codes due to the inappropriate initialization in naive VQ, which highlights the advantage and importance of a compact structured codebook. 2) IVQ consistently performs better in PSNR, SSIM, KLD while achieving smaller codebook size compared with RVQ and PVQ. It is because RVQ lacks expressivity in high-dimensional spaces with sparse codes, while PVQ suffers from inter-block disjointness. IVQ synergistically couples them to achieve a globally coherent representation. 3) LFQs show clear performance degradation while FSQs almost fail the reconstruction despite their large discrete spaces. This is due to the rigid discrete latent space in LFQ and fixed grids in FSQ which cannot be successfully adapted to complex audio signals, suggesting that reducing lookup complexity alone is insufficient without proper structural constraints.

Overall, these results validate the effectiveness of IVQ as a lightweight and structured quantization scheme that achieves a favorable trade-off between representation qual-

Table 2. Evaluation of music reconstruction application. (FAD: Fréchet Audio Distance, FD: Fréchet Distance.)

Model	Codebook Size	CU↑	KLD↓	FAD↓	FD↓	LSD↓	PSNR↑	SSIM↑	CS↑
Encodec	4×2048	79%	0.17	1.60	**0.83**	1.67	21.25	0.45	**0.67**
WavTokenizer	1×4096	99%	0.17	**1.44**	0.96	1.78	21.42	0.43	0.65
MuCodec	1×16384	32%	0.39	2.84	2.43	1.75	20.50	0.47	0.40
SemantiCodec	1×32768	21%	0.23	4.13	2.16	3.78	22.34	**0.60**	0.49
Our (Encodec + IVQ)	**4×2**	**100%**	**0.15**	2.56	1.34	**1.54**	**22.51**	0.47	**0.67**

ity, codebook efficiency, and training stability.

Application Comparison. From Table 2, the results show that: 1) Most music models rely on even larger codebooks, since music contains complex temporal information. However, our IVQ model constructed only on the *Yin–Yang* hierarchy reduces the codebook size by thousands of times; 2) Some large-codebook models exhibit extremely low code utilization, leading to severe waste of storage and training resources. This observation validates the feasibility and necessity of our lightweight IVQ design; 3) Compared with baselines, our model outperforms others on more than half metrics. Although FAD and FD still show a slight gap from the best results, the differences lie within an acceptable trade-off range and define directions for future improvement; 4) Our model is based on Encodec+IVQ, the comparison with Encodec also demonstrates the effective contribution of IVQ to both compression and reconstruction quality.

Visualization. We further visualize the code vectors and embeddings of audio models by applying PCA for dimensionality reduction. As shown in Fig. 3, the code vectors of IVQ exhibit a well-organized and evenly distributed pattern, with no dead codes observed. Under the guidance of IVQ, the encoder embeddings are also concentrated within a compact region surrounding the IVQ code space, indicating a more structured latent organization. In contrast, other models display varying degrees of code overlap and noticeable dead codes, suggesting significant redundancy in the representations. These observations further demonstrate the feasibility and effectiveness of IVQ in maintaining a compact, fully utilized, and semantically coherent codebook.

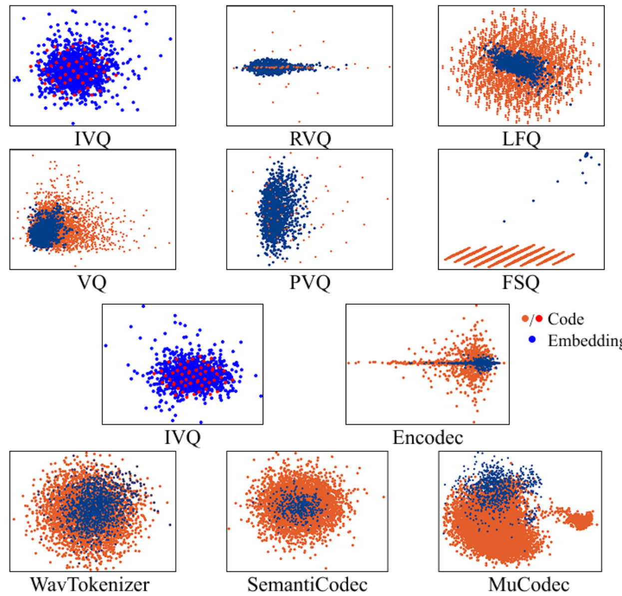

Figure 3. Visualization results of audio VQ models.

4.4. Universality Study

Visual Reconstruction. We adopt IVQ to visual VQ-based frameworks for comparisons on 30k images in ImageNet-1k (Deng et al., 2009) test set. From Table 4, we can find that: 1) Most baselines adopt large codebooks with low code utilization. In contrast, IVQ (bottom-up) reduces the codebook size by 50–100 times with full utilization, which significantly improves the efficiency and reduces the complexity; 2) Our model can perform best in FID and IS which indicates a high-quality and similar reconstruction, while shows a slight gap in PSNR which reflects a trade-off between compactness and reconstruction quality; 3) When applying the IVQ to ViT-VQVAE and VQVAE, the results demonstrate that IVQ not only compresses the codebook and improves code utilization, but also achieves consistent improvements across all metrics compared with naive VQ.

Video-to-Music Generation. Based on the aforementioned visual and audio models, we further conduct experiments on video-to-music generation with IVQ framework. Table 3 shows that our model achieves comparable or even superior performance to baselines, particularly in generative similarity and subjective evaluation. It is worth noting that all baselines rely on large pretrained backbones including ViT and MusicGen, while our approach does not use any pretrained large-scale models. Despite this, the IVQ-based model achieves a performance comparable to those built upon large pretrained systems, demonstrating the strong representational capacity and efficiency brought by the IVQ.

4.5. Ablation Study

Table 5 and 6 show that removal of the hierarchical (BHC) or structured (GSR) design from IVQ leads to performance degradation in visual and audio reconstruction. The effect is more pronounced in PSNR and FID in the visual domain, while all metrics perform much worse in audio. Completely removing the IVQ strategy results in an even larger perfor-

Table 3. Evaluation of video-to-music generation (CMR: Cross Modal Relevance, TA: Temporal Alignment, OMQ: Overall Music Quality, MVC: Music-Video Correspondece).

Model	generative similarity					audio quality		correspondence		subjective	
	KLD↓	FAD↓	FD↓	LSD↓	CS↑	PSNR↑	SSIM↑	CMR↑	TA↑	OMQ↑	MVC↑
VidMuse	1.12	3.78	2.79	2.20	0.09	**15.00**	0.15	0.60	**0.64**	3.03±0.09	3.41±0.10
M^2UGen	1.54	4.49	4.14	2.51	0.09	13.20	0.09	0.60	0.63	3.19±0.08	2.26±0.09
GVMGen	1.50	5.67	3.22	2.47	0.08	13.37	0.12	**0.66**	0.61	3.05±0.09	2.00±0.10
Control V2M	1.14	3.26	**2.00**	**2.14**	0.09	14.54	**0.16**	0.62	**0.64**	2.94±0.09	2.88±0.11
Our (IVQV2M)	**1.05**	**3.15**	2.58	2.36	**0.11**	13.61	**0.16**	0.63	0.61	**3.21±0.08**	**3.46±0.09**

Table 4. Evaluation of visual reconstruction (Size: Codebook Size, FID: Fréchet Inception Distance, IS: Inception Score).

Model	Size	CU↑	FID↓	IS↑	PSNR↑
ViT-VQGAN	8192	4.6%	23.15	87.30	**21.79**
RQ-VAE	8192	**100%**	5.36	107.85	20.23
VQVAE	4096	**100%**	12.26	81.53	16.39
ViT-VQVAE	4096	76%	2.95	151.33	16.87
IVQVAE	**78**	**100%**	8.84	98.66	17.15
ViT-IVQVAE	**78**	**100%**	**2.77**	**159.23**	17.85

Table 5. Ablation study of visual reconstruction.

Model	PSNR↑	SSIM↑	FID↓
bottom-up	**21.20**	**0.56**	**4.33**
w.o. hierarchy	20.93	**0.56**	4.35
w.o. structure	20.93	**0.56**	4.86
w.o. hier&stru	20.88	0.56	4.43
w.o. IVQ	19.68	0.54	4.83
top-down	20.34	0.54	5.37
Yin-Yang	13.40	0.32	60.96

mance drop and requires a much larger codebook to achieve comparable representational capacity.

We further evaluate several IVQ variants, including bottom-up, top-down, and 4-layer *Yin–Yang*–only quantization. Most of these variants outperform the non-IVQ baseline, though their relative strengths vary. We could also find that shared subspace in each composition not only reduces the complexity but also performs better. Considering efficiency, we adopt *Yin–Yang* IVQ for audio due to its smallest codebook size and comparable results. In contrast, we use bottom-up IVQ for visuals since it performs much better.

We also generalize IVQ based on the invariance and variability of *IChing*. Specifically, we modify the hexagram by

Table 6. Ablation study of audio reconstruction.

Model	Codebook Size	KLD↓	FAD↓	CS↑
Yin-Yang	**4×2**	0.68	2.43	**0.82**
non-share	4×2×6	0.86	2.75	0.71
w.o. hierarchy	4×2	0.89	2.93	0.74
w.o. structure	4×2	0.76	2.60	0.77
w.o. hier&stru	4×2	0.77	2.75	0.79
w.o. IVQ	4×64	0.79	2.80	0.75
bottom-up	78	0.82	2.41	0.75
top-down	78	**0.66**	**2.36**	0.79

Table 7. Universality of IVQ in audio reconstruction.

Model	Codebook	CU	KLD↓	FAD↓	CS↑
4-line IVQ	**4×2**	**100%**	**0.82**	**2.94**	**0.76**
w.o. IVQ	4×16	**100%**	0.87	**2.94**	0.75
12-line IVQ	**4×2**	**100%**	0.43	1.77	0.86
w.o IVQ	4×4096	84%	**0.35**	**1.66**	**0.88**

adjusting the number of lines. When reducing each hexagram from 6 to 4 lines, the code number decreases to 16, and it still outperforms 16-code RVQ as shown in Table 7. Conversely, when extending the hexagram to 12 lines, the number of codes increases to 4,096. Although the performance is slightly inferior to 4096-code RVQ, it achieves significant advantages in computational efficiency and codebook utilization, confirming the universality of IVQ.

5. Conclusion

In this paper, we address the major limitations of existing VQ methods, which often rely on large codebooks without optimizing the internal structure, leading to codebook collapse and redundancy. Inspired by the ancient Chinese classic *IChing*, we propose *IChing* Vector Quantization (IVQ), a lightweight yet expressive codebook with hierarchical and structured design. IVQ can be applied to multimodal representation as well as cross-modal tasks. Extensive experiments demonstrate that IVQ achieves improved codebook utilization and performance compared to existing approaches. In future, we will explore variants of IVQ to further enhance its applicability in multimodal tasks.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2023YFF0904900), and the National Natural Science Foundation of China (No. 62577048).

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Agostinelli, A., Denk, T. I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., Huang, Q., Jansen, A., Roberts, A., Tagliasacchi, M., et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

Babenko, A. and Lempitsky, V. Additive quantization for extreme vector compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 931–938, 2014.

Bogdanov, D., Won, M., Tovstogan, P., Porter, A., and Serra, X. The mtg-jamendo dataset for automatic music tagging. ICML, 2019.

Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Roblek, D., Teboul, O., Grangier, D., Tagliasacchi, M., et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31: 2523–2533, 2023.

Buzo, A., Gray, A., Gray, R., and Markel, J. Speech coding based upon vector quantization. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(5):562–574, 1980.

Chen, Y., Guan, T., and Wang, C. Approximate nearest neighbor search by residual vector quantization. *Sensors*, 10(12):11259–11273, 2010.

Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., and Défossez, A. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Défossez, A., Copet, J., Synnaeve, G., and Adi, Y. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Défossez, A., Copet, J., Synnaeve, G., and Adi, Y. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Gautam, T., Pryzant, R., Yang, Z., Zhu, C., and Sojoudi, S. Soft convex quantization: revisiting vector quantization with convex optimization. In *6th Annual Learning for Dynamics & Control Conference*, pp. 273–285. PMLR, 2024.

Gray, R. Vector quantization. *IEEE Assp Magazine*, 1(2): 4–29, 1984.

Huh, M., Cheung, B., Agrawal, P., and Isola, P. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *International Conference on Machine Learning*, pp. 14096–14113. PMLR, 2023.

Jegou, H., Douze, M., and Schmid, C. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

Ji, S., Jiang, Z., Wang, W., Chen, Y., Fang, M., Zuo, J., Yang, Q., Cheng, X., Wang, Z., Li, R., et al. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. *arXiv preprint arXiv:2408.16532*, 2024.

Kumar, R., Seetharaman, P., Luebs, A., Kumar, I., and Kumar, K. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36:27980–27993, 2023.

Łańcucki, A., Chorowski, J., Sanchez, G., Marxer, R., Chen, N., Dolfing, H. J., Khurana, S., Alumäe, T., and Laurent, A. Robust training of vector quantized bottleneck models. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2020.

Li, H., Xue, L., Guo, H., Zhu, X., Lv, Y., Xie, L., Chen, Y., Yin, H., and Li, Z. Single-codec: Single-codebook speech codec towards high-performance speech generation. *arXiv preprint arXiv:2406.07422*, 2024.

Liu, H., Xu, X., Yuan, Y., Wu, M., Wang, W., and Plumbley, M. D. Semanticodec: An ultra low bitrate semantic audio codec for general sound. *IEEE Journal of Selected Topics in Signal Processing*, 2024.

Mentzer, F., Minnen, D., Agustsson, E., and Tschannen, M. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Razavi, A., Van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Tian, Z., Liu, Z., Yuan, R., Pan, J., Liu, Q., Tan, X., Chen, Q., Xue, W., and Guo, Y. Vidmuse: A simple video-to-music generation framework with long-short-term modeling. *arXiv preprint arXiv:2406.04321*, 2024.

Vali, M. H. and Bäckström, T. Nsvq: Noise substitution in vector quantization for machine learning. *IEEE Access*, 10:13598–13610, 2022.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Xu, Y., Chen, H., Yu, J., Tan, W., Lei, S., Lin, Z., Gu, R., and Wu, Z. Mucodec: Ultra low-bitrate music codec for music generation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 689–698, 2025.

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

Yu, L., Lezama, J., Gundavarapu, N. B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Birodkar, V., Gupta, A., Gu, X., et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

Yu, Q., Weber, M., Deng, X., Shen, X., Cremers, D., and Chen, L.-C. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024.

Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., and Tagliasacchi, M. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

Zhang, G., Tang, L., and Zhang, X. Vqnerv: Vector quantization neural representation for video compression. In *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5. IEEE, 2024.

Zhuo, L., Wang, Z., Wang, B., Liao, Y., Bao, C., Peng, S., Han, S., Zhang, A., Fang, F., and Liu, S. Video background music generation: Dataset, method and evaluation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15637–15647, 2023.

Zuo, H., You, W., Wu, J., Ren, S., Chen, P., Zhou, M., Lu, Y., and Sun, L. Gvmgen: A general video-to-music generation model with hierarchical attentions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

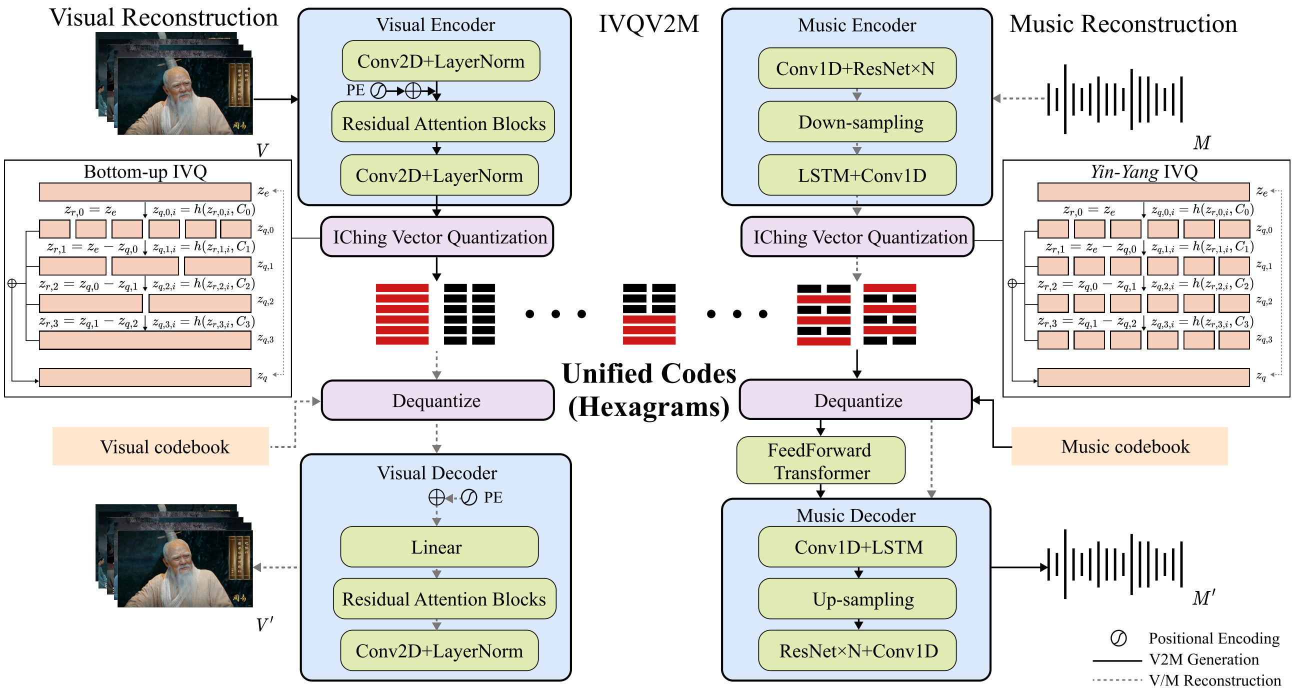

Figure 4. Model architecture with IVQ for visual and music reconstruction and video-to-music generation (IVQV2M).

A. *IChing* Vector Quantization

As shown in Section 3.2, the Bottom-up IVQ operates by applying block sizes $B = \{6, 3, 2, 1\}$ from coarse to fine granularity. For the Top-down variant, we simply reverse the order and set $B = \{1, 2, 3, 6\}$. For the *Yin–Yang* variant, all blocks are fixed to size 6, which not only reduces the total number of codes but also decreases the dimensionality represented by each code from D to $D/6$. We define the IVQ as Algorithm 1.

In cross-modal generation, the input z_e is a visual embedding. Its indices $\text{idx}^{(j)}$, obtained from the visual codebooks C_V and concatenated, map to a corresponding hexagram index. This index can then be dequantized using the music codebooks C_m, effectively converting the visual embedding into music for following music generation.

B. Model Architecture and Implementation Details

Visual encoder and decoder. Inspired by (Yu et al., 2024), we utilize ViT as the backbone. Each image $x \in \mathbb{R}^{H \times W \times C}$ is segmented into patches through a CNN patch embedding layer and then concatenated with latent tokens, which are then sent into the feedforward transformer blocks with positional embeddings to derive z_e. We use latent tokens to enable a more compact representation of the image, which will be quantized to z_q with an IVQ codebook. In decoding process, we incorporate a sequence of mask tokens to the quantized image tokens, and then reconstruct images from them through a ViT decoder.

Music encoder and decoder. For music, we use convolutional networks in encoder for down-sampling and up-sampling since it contains more complex temporal information. In detail, a Conv1D layer is utilized to embed the music and followed by a sequence of Conv1D blocks composed of a single residual unit and a down-sampling layer following (Défossez et al., 2022). Then, a two LSTM layers is used for sequential modeling and a Conv1D layer for encoding output z_e, which is quantized into z_q through IVQ before being sent into decoder. The structure of decoder is in the reverse order of the encoder, using transposed convolutions instead of strided convolutions.

Following previous works, we train our model on ImageNet (Deng et al., 2009) for visual reconstruction while on MTG (Bogdanov et al., 2019) for music. For video-to-music generation, we train the model on GVMGen (Zuo et al., 2025), SymMV (Zhuo et al., 2023) and VidMuse dataset (Tian et al., 2024). In evaluation, we use 30k images in ImageNet validation set for visual reconstruction. For video-to-music generation, since baselines are trained on different datasets, we

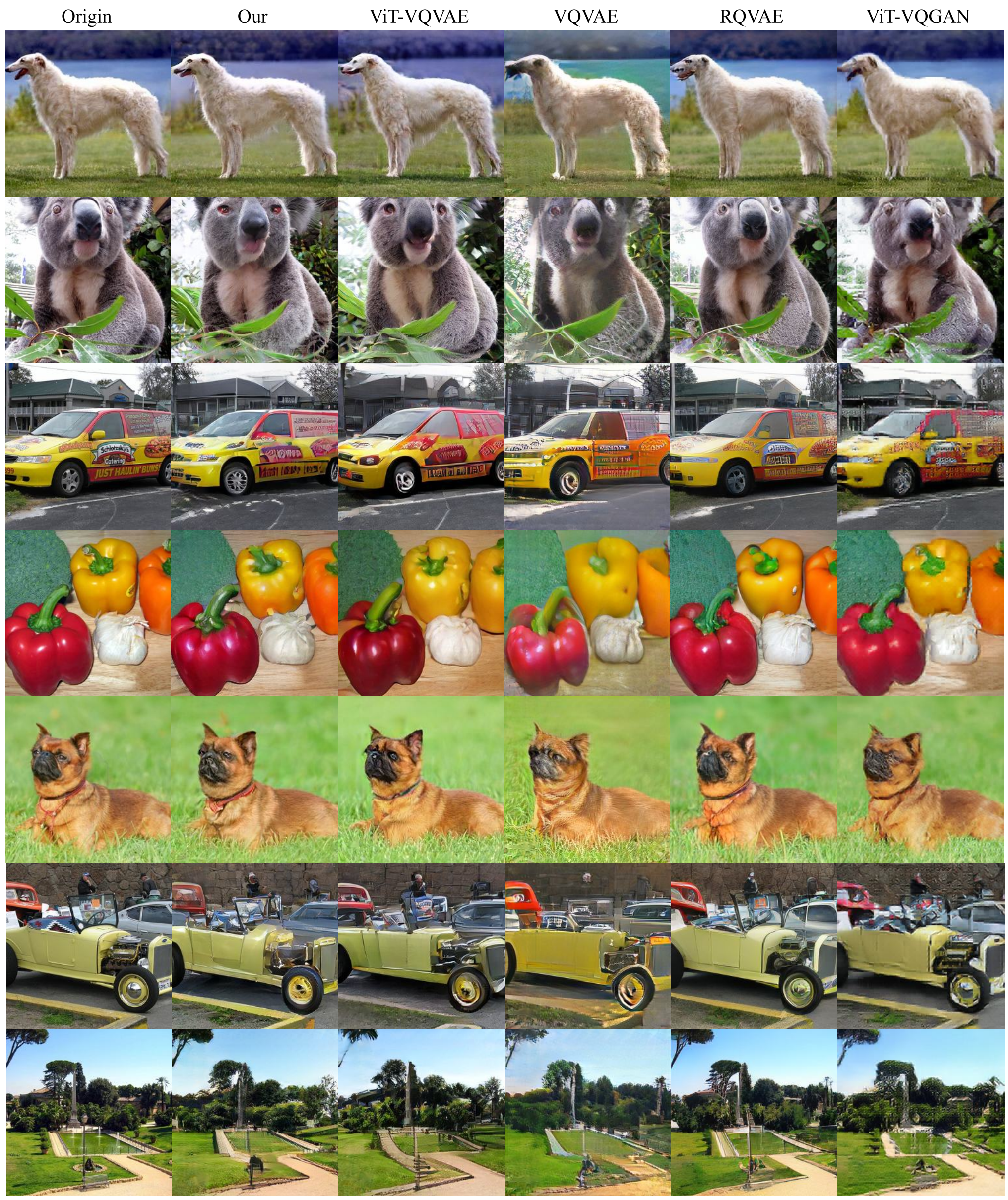

Figure 5. Visual reconstruction comparison.

uniformly select 135 videos for objective evaluation while 30 for subjective from GVMGen, Videmuse and SymMV test set.

C. Experimental Results

We further provide additional qualitative results in Figures 5 and 6. As shown in Fig. 5, our model achieves the highest reconstruction quality in both color fidelity and structural consistency, with particularly strong performance in preserving fine details. In contrast, other models exhibit noticeable degradation—for example, VQVAE suffers from overexposure and color shifts, while VQGAN shows reduced image sharpness and significant artifacts.

A similar observation holds for Fig. 6 in music. Our model produces cleaner timbre with improved granularity and overall music quality. In comparison, SemantiCodec exhibits coarse and unstable sound textures, while MuCodec introduces evident high- and low-frequency noise. These qualitative comparisons further highlight the advantages of the proposed IVQ framework in both visual and auditory reconstruction. Fig. 7 illustrates a sample of video-to-music generation using

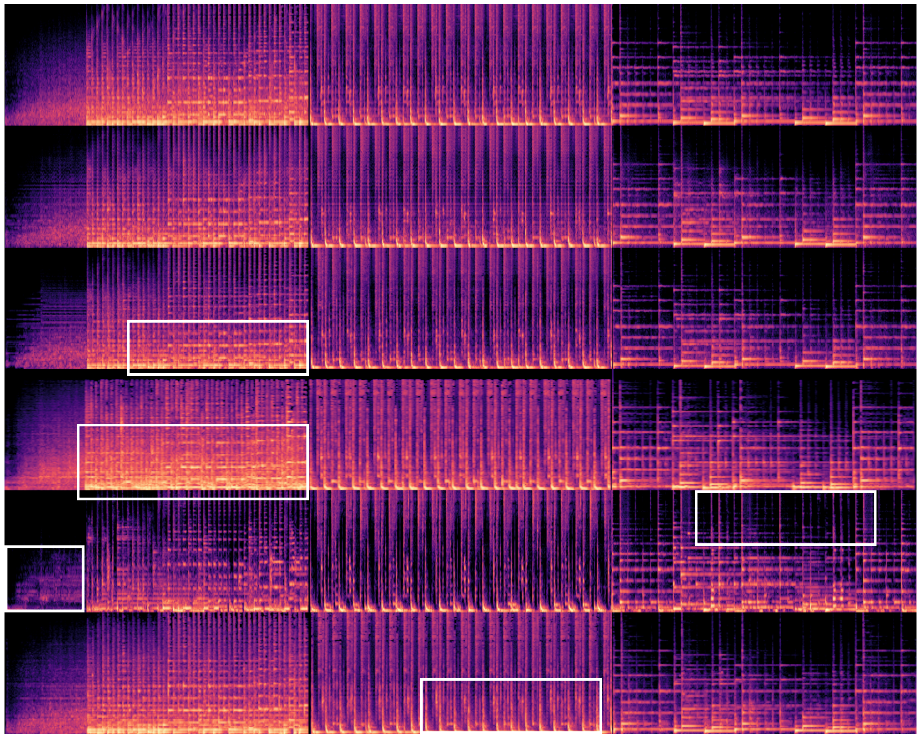

Figure 6. Music reconstruction comparison.

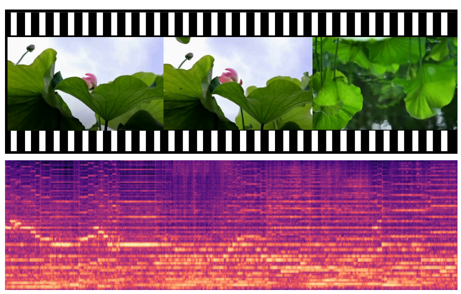

Figure 7. Sample of video-to-music generation.

IVQV2M. There is a piece of clear melody which is highly correspond with the input video.

We additionally supplement the visualization of the codebooks and embeddings presented in Fig. 3. All baseline models employ extremely large codebooks in which many codes heavily overlap while others remain unused, resulting in a substantial number of dead codes. Although these models can reconstruct embeddings with high fidelity, this comes at a significant representational and computational cost. In contrast, IVQ uses a much more compact and structured codebook. While a small portion of embeddings lie far from code vectors—leading to quantization loss—we argue that it is more faithful to the original intent of VQ. VQ is not meant to fill the entire latent space with a dense set of codes; rather, it is fundamentally a compression technique that should represent the space using a limited number of discrete units. Consequently, the commitment loss during quantization is an expected and reasonable trade-off, inherent to the VQ principle.

D. Limitations

Despite the significant contributions of our method in compressing codebooks and mitigating codebook collapse, several limitations remain. First, the *Yin–Yang* variant with an ultra-small codebook is still insufficient for high–fidelity image reconstruction, leading to notable failures, especially in fine-grained regions such as text. Even with the full IVQ design, certain subtle details may still be lost—an issue shared by many existing VQ-based models. In audio reconstruction, some inevitable noise artifacts continue to affect perceived sound quality.

Second, although our approach offers a lightweight cross-modal bridging mechanism for video-to-music generation, it remains limited in achieving fine-grained audiovisual alignment and in handling complex, narrative-driven videos, where deeper temporal and semantic modeling may be required.

Finally, due to limited computational and time resources, we were unable to evaluate IVQ on larger models. Thus, its applicability to very large-scale architectures remains an open question. These limitations reflect not only the boundaries of our current work but also broader challenges in the field, and they outline clear directions for future research.