# Discrete Tokenization for Multimodal LLMs: A Comprehensive Survey

Jindong Li, Yali Fu, Jiahong Liu, Linxiao Cao, Wei Ji, Menglin Yang, Irwin King, Ming-Hsuan Yang

**Abstract**—The rapid advancement of large language models (LLMs) has intensified the need for effective mechanisms to transform continuous multimodal data into discrete representations suitable for language-based processing. Discrete tokenization, with vector quantization (VQ) as a central approach, offers both computational efficiency and compatibility with LLM architectures. Despite its growing importance, there is a lack of a comprehensive survey that systematically examines VQ techniques in the context of LLM-based systems. This work fills this gap by presenting the first structured taxonomy and analysis of discrete tokenization methods designed for LLMs. We categorize 8 representative VQ variants that span classical and modern paradigms and analyze their algorithmic principles, training dynamics, and integration challenges with LLM pipelines. Beyond algorithm-level investigation, we discuss existing research in terms of classical applications without LLMs, LLM-based single-modality systems, and LLM-based multimodal systems, highlighting how quantization strategies influence alignment, reasoning, and generation performance. In addition, we identify key challenges including codebook collapse, unstable gradient estimation, and modality-specific encoding constraints. Finally, we discuss emerging research directions such as dynamic and task-adaptive quantization, unified tokenization frameworks, and biologically inspired codebook learning. This survey bridges the gap between traditional vector quantization and modern LLM applications, serving as a foundational reference for the development of efficient and generalizable multimodal systems. A continuously updated version is available at: https://github.com/jindongli-Ai/LLM-Discrete-Tokenization-Survey.

**Index Terms**—Discrete Tokenization, Vector Quantization (VQ), Multiple Modalities, Large Language Models (LLMs).

## 1 Introduction

Recent advances in large language models (LLMs) [30, 50, 52, 141, 149, 150, 232] have significantly transformed the way machines understand and generate human language. These models have demonstrated exceptional capabilities in language comprehension and generation, driving their adoption across a wide range of applications. As research continues to evolve, there is growing interest in extending the capabilities of LLMs beyond text to encompass multimodal data, including images [62, 276], audio [33, 59], and video [81, 87], thus introducing new challenges in unifying heterogeneous modalities within a common framework.

Discrete tokenization based on vector quantization (VQ) has emerged as a key technique to address these challenges, offering significant advantages for multimodal integration in LLMs [19, 100]. As illustrated in Fig. 1, by transforming high-dimensional continuous inputs into compact discrete tokens, it enables non-text modalities to be processed in a format aligned with the inherently token-based structure of language models. This design not only improves computational efficiency through compression, but also retains semantic granularity essential for cross-modal reasoning. Due to these strengths, discrete tokenization has become a core component in many state-of-the-art multimodal LLM systems.

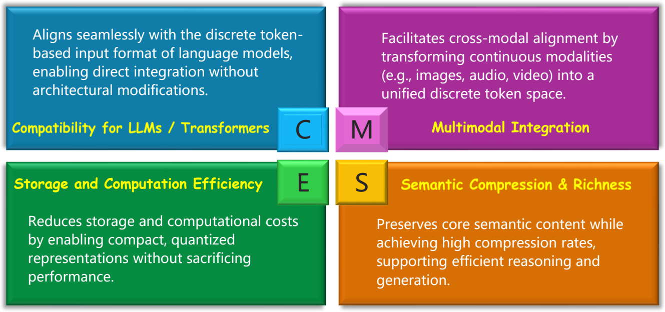
Aligns seamlessly with the discrete token-based input format of language models, enabling direct integration without architectural modifications.
Compatibility for LLMs / Transformers
C
Facilitates cross-modal alignment by transforming continuous modalities (e.g., images, audio, video) into a unified discrete token space.
M
Multimodal Integration
Storage and Computation Efficiency
E
Reduces storage and computational costs by enabling compact, quantized representations without sacrificing performance.
S
Semantic Compression & Richness
Preserves core semantic content while achieving high compression rates, supporting efficient reasoning and generation.

Fig. 1. Discrete tokenization enables seamless integration with language models and supports efficient, scalable, and semantically meaningful processing for multimodal LLMs.

Despite the growing relevance of discrete tokenization, existing surveys remain limited in both scope and technical depth. Several earlier reviews [9, 92, 142, 188, 221] cover topics before the emergence of LLMs and are no longer adequate in the context of today's rapidly evolving AI landscape. While recent works have offered broader overviews of multimodal learning systems [77], the treatment of quantization techniques remains insufficient. Other surveys are narrowly scoped and restricted to individual modalities or tasks. For instance, Lin et al. [102] provides a comprehensive account of quantization methods for graph-structured data, yet does not generalize beyond this domain. Similarly, [111]

- *Jindong Li, Linxiao Cao, and Menglin Yang are with Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. E-mail: jli839@connect.hkust-gz.edu.cn, lcao950@connect.hkust-gz.edu.cn, menglin.yang@outlook.com.*
- *Yali Fu is with Jilin University, Changchun, China. E-mail: fuyl23@mails.jlu.edu.cn.*
- *Jiahong Liu and Irwin King are with The Chinese University of Hong Kong, Hong Kong, China. E-mail: jiahong.liu21@gmail.com, king@cse.cuhk.edu.hk.*
- *Wei Ji is with the Nanjing University, China. Email:weiji0523. gmail.com*
- *Ming-Hsuan Yang is with the University of California at Merced, United States. Email: mhyang@ucmerced.edu*
- *Jindong Li and Yali Fu contribute equally as co-first authors. Menglin Yang is the corresponding author.*

*Manuscript received xxxx xx, 20xx; revised xxxx xx, 20xx.*

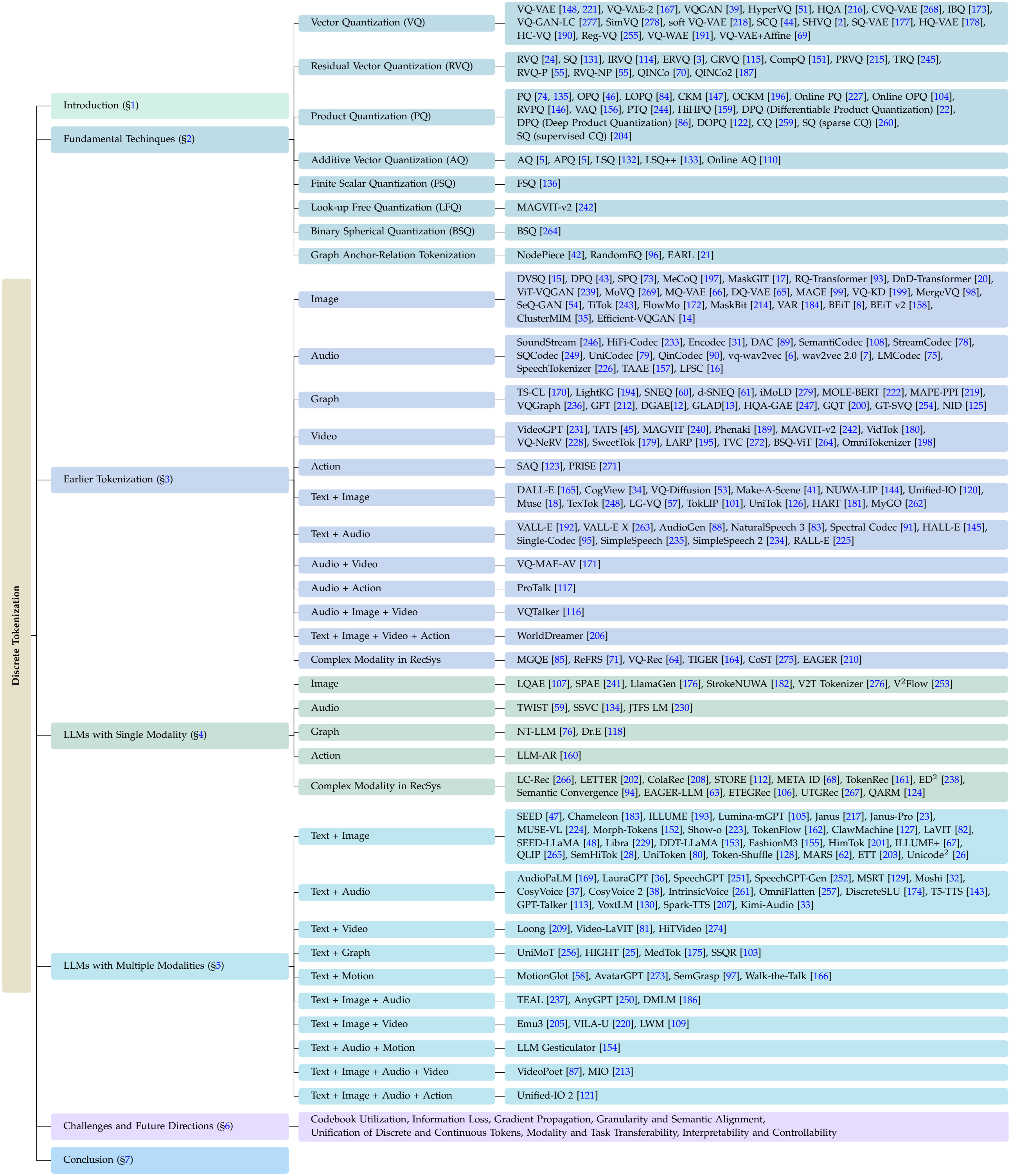
Discrete Tokenization
Introduction (§1)
Fundamental Techinques (§2)
Vector Quantization (VQ)
VQ-VAE [148, 221], VQ-VAE-2 [167], VQGAN [39], HyperVQ [51], HQA [216], CVQ-VAE [268], IBQ [173], VQ-GAN-LC [277], SimVQ [278], soft VQ-VAE [218], SCQ [44], SHVQ [2], SQ-VAE [177], HQ-VAE [178], HC-VQ [190], Reg-VQ [255], VQ-WAE [191], VQ-VAE+Affine [69]
Residual Vector Quantization (RVQ)
RVQ [24], SQ [131], IRVQ [114], ERVQ [3], GRVQ [115], CompQ [151], PRVQ [215], TRQ [245], RVQ-P [55], RVQ-NP [55], QINCo [70], QINCo2 [187]
Product Quantization (PQ)
PQ [74, 135], OPQ [46], LOPQ [84], CKM [147], OCKM [196], Online PQ [227], Online OPQ [104], RVPQ [146], VAQ [156], PTQ [244], HiHPQ [159], DPQ (Differentiable Product Quantization) [22], DPQ (Deep Product Quantization) [86], DOPQ [122], CQ [259], SQ (sparse CQ) [260], SQ (supervised CQ) [204]
Additive Vector Quantization (AQ)
AQ [5], APQ [5], LSQ [132], LSQ++ [133], Online AQ [110]
Finite Scalar Quantization (FSQ)
FSQ [136]
Look-up Free Quantization (LFQ)
MAGVIT-v2 [242]
Binary Spherical Quantization (BSQ)
BSQ [264]
Graph Anchor-Relation Tokenization
NodePiece [42], RandomEQ [96], EARL [21]
Earlier Tokenization (§3)
Image
DVSQ [15], DPQ [43], SPQ [73], MeCoQ [197], MaskGIT [17], RQ-Transformer [93], DnD-Transformer [20], ViT-VQGAN [239], MoVQ [269], MQ-VAE [66], DQ-VAE [65], MAGE [99], VQ-KD [199], MergeVQ [98], SeQ-GAN [54], TiTok [243], FlowMo [172], MaskBit [214], VAR [184], BEiT [8], BEiT v2 [158], ClusterMIM [35], Efficient-VQGAN [14]
Audio
SoundStream [246], HiFi-Codec [233], Encodec [31], DAC [89], SemantiCodec [108], StreamCodec [78], SQCodec [249], UniCodec [79], QinCodec [90], vq-wav2vec [6], wav2vec 2.0 [7], LMCodec [75], SpeechTokenizer [226], TAAE [157], LFSC [16]
Graph
TS-CL [170], LightKG [194], SNEQ [60], d-SNEQ [61], iMoLD [279], MOLE-BERT [222], MAPE-PPI [219], VQGraph [236], GFT [212], DGAE[12], GLAD[13], HQA-GAE [247], GQT [200], GT-SVQ [254], NID [125]
Video
VideoGPT [231], TATS [45], MAGVIT [240], Phenaki [189], MAGVIT-v2 [242], VidTok [180], VQ-NeRV [228], SweetTok [179], LARP [195], TVC [272], BSQ-ViT [264], OmniTokenizer [198]
Action
SAQ [123], PRISE [271]
Text + Image
DALL-E [165], CogView [34], VQ-Diffusion [53], Make-A-Scene [41], NUWA-LIP [144], Unified-IO [120], Muse [18], TexTok [248], LG-VQ [57], TokLIP [101], UniTok [126], HART [181], MyGO [262]
Text + Audio
VALL-E [192], VALL-E X [263], AudioGen [88], NaturalSpeech 3 [83], Spectral Codec [91], HALL-E [145], Single-Codec [95], SimpleSpeech [235], SimpleSpeech 2 [234], RALL-E [225]
Audio + Video
VQ-MAE-AV [171]
Audio + Action
ProTalk [117]
Audio + Image + Video
VQTalker [116]
Text + Image + Video + Action
WorldDreamer [206]
Complex Modality in RecSys
MGQE [85], ReFRS [71], VQ-Rec [64], TIGER [164], CoST [275], EAGER [210]
LLMs with Single Modality (§4)
Image
LQAE [107], SPAE [241], LlamaGen [176], StrokeNUWA [182], V2T Tokenizer [276], V$^2$Flow [253]
Audio
TWIST [59], SSVC [134], JTFS LM [230]
Graph
NT-LLM [76], Dr.E [118]
Action
LLM-AR [160]
Complex Modality in RecSys
LC-Rec [266], LETTER [202], ColaRec [208], STORE [112], META ID [68], TokenRec [161], ED$^2$ [238], Semantic Convergence [94], EAGER-LLM [63], ETEGRec [106], UTGRec [267], QARM [124]
LLMs with Multiple Modalities (§5)
Text + Image
SEED [47], Chameleon [183], ILLUME [193], Lumina-mGPT [105], Janus [217], Janus-Pro [23], MUSE-VL [224], Morph-Tokens [152], Show-o [223], TokenFlow [162], ClawMachine [127], LaVIT [82], SEED-LLaMA [48], Libra [229], DDT-LLaMA [153], FashionM3 [155], HimTok [201], ILLUME+ [67], QLIP [265], SemHiTok [28], UniToken [80], Token-Shuffle [128], MARS [62], ETT [203], Unicode$^2$ [26]
Text + Audio
AudioPaLM [169], LauraGPT [36], SpeechGPT [251], SpeechGPT-Gen [252], MSRT [129], Moshi [32], CosyVoice [37], CosyVoice 2 [38], IntrinsicVoice [261], OmniFlatten [257], DiscreteSLU [174], T5-TTS [143], GPT-Talker [113], VoxtLM [130], Spark-TTS [207], Kimi-Audio [33]
Text + Video
Loong [209], Video-LaVIT [81], HiTVideo [274]
Text + Graph
UniMoT [256], HIGHT [25], MedTok [175], SSQR [103]
Text + Motion
MotionGlot [58], AvatarGPT [273], SemGrasp [97], Walk-the-Talk [166]
Text + Image + Audio
TEAL [237], AnyGPT [250], DMLM [186]
Text + Image + Video
Emu3 [205], VILA-U [220], LWM [109]
Text + Audio + Motion
LLM Gesticulator [154]
Text + Image + Audio + Video
VideoPoet [87], MIO [213]
Text + Image + Audio + Action
Unified-IO 2 [121]
Challenges and Future Directions (§6)
Codebook Utilization, Information Loss, Gradient Propagation, Granularity and Semantic Alignment,
Unification of Discrete and Continuous Tokens, Modality and Task Transferability, Interpretability and Controllability
Conclusion (§7)

Fig. 2. Taxonomy of this survey with representative works. Specifically, it is organized from the perspective of modality.

centers exclusively on recommender systems, emphasizing efficiency and representation quality, while [56] focuses solely on discrete speech tokens for representation learning. This fragmentation and lack of cross-modal integration pose challenges for researchers aiming to design general-purpose, LLM-based multimodal systems.

In this work, VQ-based discrete tokenization is systematically discussed to better understand its role in addressing the limitations of current multimodal LLM systems. The analysis connects tokenization design choices to key integration requirements of LLMs, such as maintaining token alignment and ensuring effective gradient propagation through quantized representations. By analyzing applications across all major modalities within a unified analytical framework, this survey offers comparative insights that have been lacking in prior literature. Furthermore, it identifies and clarifies key challenges in current implementations, providing practical insights for enhancing quantization quality and system robustness. The overall structure of our survey is shown in Fig. 2. Our main contributions are summarized as follows:

- We establish a comprehensive taxonomy that organizes existing discrete tokenization methods based on their codebook learning paradigms and compatibility with LLM integration requirements.
- Representative applications in non-LLM settings are reviewed to reveal how their design principles can inform the construction of modality-specific tokenization strategies suitable for LLMs.
- A detailed modality-wise analysis is provided that compares discrete tokenization approaches across various data types within LLM systems.
- Key challenges in current techniques are identified and future research directions are outlined, including strategies to mitigate codebook collapse and to enable dynamic and adaptive quantization.

## 2 Preliminaries

In the context of LLM, discrete tokenization (quantization in non-LLM models) serves as the fundamental unit of representation, enabling efficient processing and generation of complex data across modalities. Tokens are derived through quantization techniques, which map continuous or high-dimensional data to a discrete, finite set of representations known as a codebook. A typical formulation of discrete quantization follows the pipeline as shown in Fig. 3.

**General Formulation.** *The discrete quantization pipeline begins with input data* $\mathbf{x}$ *(e.g., image, audio), which is processed by an encoder into a continuous latent representation* $\mathbf{z}$*. This continuous representation* $\mathbf{z}$ *is then discretized to a specific representation* $\mathbf{c}_q$ *in the codebook through a quantization process* $\mathcal{Q}$*. Finally, the discrete representation* $\mathbf{c}_q$ *is passed to a decoder, outputting* $\hat{\mathbf{x}}$ *to approximate* $\mathbf{x}$ *as much as possible.*

The process involves transforming continuous data into discrete tokens, which are encouraged to retain sufficient information and then used for downstream tasks such as generation or classification. The encoder and decoder typically consist of a deep neural network (e.g., convolutional or transformer-based layers) [118, 226, 239, 242], depending on the data modality.

To effectively implement the discrete quantization pipeline, three critical questions need to be addressed: **Q1:** How to train the entire pipeline? **Q2:** How to flow gradient through the discrete bottleneck? **Q3:** How to implement the quantization process $\mathcal{Q}$? These questions are key to enabling efficient, end-to-end training and implementation of discrete tokenization.

### *Q1: How to Train the Entire Pipeline?*

There are three primary methods for training the discrete quantization process: reconstruction-based, adversarial-based, and contrastive-based methods.

**Reconstruction-based Methods.** This paradigm usually refers to a variational autoencoder (VAE)-based framework, which learns discrete representations by optimizing the reconstruction quality of the original input data.

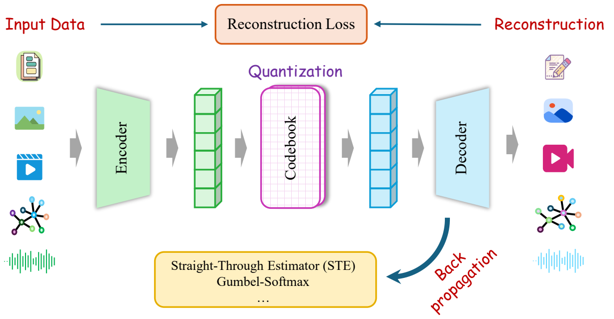

Fig. 3. General pipeline of discrete quantization based on VAE [148], involving three main stages: encoding, quantization, and decoding.

The classical and fundamental models, VQ-VAE [148] and hierarchical VQ-VAE (i.e., VQ-VAE-2 [167]) jointly optimize the whole quantization pipeline by minimizing a combined loss:

$$\mathcal{L}_{\text{vq}-\text{vae}} = \|\mathbf{x}-\hat{\mathbf{x}}\|_2^2 + \|\text{sg}[\mathbf{z}]-\mathbf{c}_q\|_2^2 + \beta\|\mathbf{z}-\text{sg}[\mathbf{c}_q]\|_2^2, \quad (1)$$

$$\text{sg}(x) = \begin{cases} x & \text{forward pass, identity function} \\ 0 & \text{backward pass, gradient is stopped} \end{cases}, \quad (2)$$

where the three terms correspond to reconstruction loss, codebook loss, and commitment loss (scaled by weight $\beta$), and $\text{sg}(\cdot)$ denotes the stop-gradient operator.

**Adversarial-based Methods.** VQGAN [39] extends the standard VQ-VAE framework by introducing adversarial training and a perceptual loss for learning a perceptually rich codebook.

The network is optimized by combining the VQ-VAE loss $\mathcal{L}_{\text{vq-vae}}$ in Eq. (1) with an adversarial loss $\mathcal{L}_{\text{gan}}$:

$$\mathcal{L}_{\text{vqgan}} = \mathcal{L}_{\text{vq-vae}} + \lambda\,\mathcal{L}_{\text{gan}}, \quad \lambda = \frac{\nabla_{\mathcal{D}_L}[\mathcal{L}_{\text{per}}]}{\nabla_{\mathcal{D}_L}[\mathcal{L}_{\text{gan}}]+\delta}, \quad (3)$$

$$\mathcal{L}_{\text{gan}} = \log\mathbb{D}(\mathbf{x}) + \log(1-\mathbb{D}(\hat{\mathbf{x}})), \quad (4)$$

where $\mathbb{D}$ is the patch-based discriminator, $\lambda$ is the weighting coefficient, $\mathcal{D}$ denotes the decoder, $\mathcal{L}_{\text{per}}$ is the perceptual loss, $\nabla_{\mathcal{D}_L}[\cdot]$ denotes the gradient of its input with respect to the last layer $L$ of the decoder, and $\delta$ is a small constant for numerical stability.

### *Q2: How to Flow Gradient Through Discrete Bottleneck?*

The argmax operation in quantization is non-differentiable (detailed in Section 2.1.2), which blocks the gradient flow during training. To address this, various strategies have been proposed.

**Straight-Through Estimator (STE).** STE [10] offers a heuristic method that enables gradient flow through a non-differentiable discrete bottleneck. It treats the quantization as an identity function during the backward pass, and directly copies the gradients from the decoder input to the encoder output. This leads to the following approximation:

$$\nabla_{\mathbf{z}}\mathcal{L} \approx \nabla_{\mathbf{c}_q}\mathcal{L}, \quad (5)$$

**Gumbel-Softmax.** The Gumbel-Softmax [6, 72] provides a differentiable approximation to categorical sampling by replacing non-differentiable discrete sampling of quantization

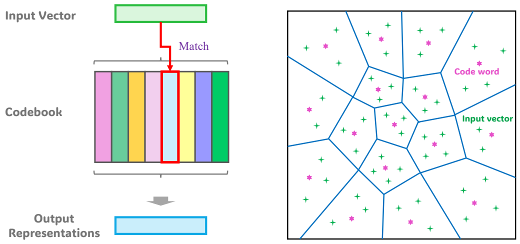

Fig. 4. Illustration of the vector quantization (VQ) mapping process: each input vector is matched to its nearest codeword in the finite codebook (left), corresponding to a partitioning of the continuous space into discrete regions (right).

with a differentiable continuous relaxation perturbed by Gumbel noise during training.

Specifically, given a categorical distribution with class probabilities $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, the discrete one-hot sample can be approximated by a differentiable softmax function:

$$\mathbf{y} = \text{softmax}\left(\frac{\log \boldsymbol{\pi} + \mathbf{g}}{\tau}\right), \tag{6}$$

where $\mathbf{g} = (g_1, \ldots, g_K)$ is a vector of i.i.d. samples drawn from the $Gumbel(0,1)$ distribution:

$$g_k = -\log\left(-\log(u_k)\right), \quad u_k \sim \text{Uniform}(0,1), \tag{7}$$

and $\tau > 0$ is the temperature parameter that controls the smoothness of the probability distribution.

During testing, as $\tau \to 0$, the distribution becomes closer to a one-hot vector by the non-differentiable argmax operation:

$$\lim_{\tau \to 0} \mathbf{y} = \text{one-hot}\left(\arg\max_k (\log \pi_k + g_k)\right), \tag{8}$$

where the exact categorical sample is recovered via the Gumbel-Max trick.

This relaxation allows gradients to flow through the discrete sampling process, enabling gradient-based optimization. During training, $\tau$ is typically annealed from a high value (for smoother distributions) to a low value (for near one-hot outputs), bridging the gap between continuous and discrete representations.

**Rotation Trick.** Fifty et al. [40] proposes a rotation trick for gradient propagation, aligning encoder outputs to their nearest codebook vectors via rotation, rescaling linear transformation and encoding relative magnitude and angle between encoder output and codebook vector in the gradient.

### *Q3: How to Implement the Quantization Process $\mathcal{Q}$?*

There are eight primary methods (section 2.1 - section 2.8) for implementing the quantization process $\mathcal{Q}$, each offering unique approaches to discretizing continuous data. The following subsections systematically review fundamental quantization methods from classical algorithms to modern innovations by highlighting their unique mechanisms.

## 2.1 Vector Quantization

In LLMs, Vector Quantization (VQ) [39, 148, 221] is a technique that discretizes continuous latent representations by mapping them to the closest entries in a finite codebook, as illustrated in Fig. 4. It plays a key role that bridges between continuous and discrete representations, enabling compact and interpretable modeling.

**Definition [Vector Quantization].** *Let $\mathcal{Z} \subseteq \mathbb{R}^D$ be the continuous input space, $\mathbf{z} \in \mathcal{Z}$ be a D-dimensional input vector, and $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\} \subseteq \mathbb{R}^D$ denote the codebook containing $K$ codewords (also called codevectors or codes). Vector Quantization defines a mapping function $q : \mathcal{Z} \to \mathcal{C}$, which assigns a continuous vector $\mathbf{z}$ to its nearest codeword $\mathbf{c}_{k^*}$, i.e., $q(\mathbf{z}) = \mathbf{c}_{k^*}$.*

### *2.1.1 Codebook Initialization*

Each codevector $\mathbf{c}_k$ in the codebook is usually a prototype vector. For the initialization of $K$ codevectors $\{\mathbf{c}_k\}_{k=1}^K$, a common practice is to sample from a Gaussian distribution $\mathcal{N}(0, I)$ or use uniform initialization in a small range (e.g., $\mathcal{U}(-0.1, 0.1)$) [239, 278]. In addition, the $K$-means clustering method can be applied to find the cluster centroids of the training embeddings for initialization [148, 200]. HyperVQ [51] defines geometrically constrained code vectors by performing hyperbolic multinomial logistic regression and selecting a representative point in the decision hyperplane.

### *2.1.2 Code Assignment: Embedding to Code Mapping*

As illustrated in Fig. 4, given the continuous latent embedding $\mathbf{z}$, vector quantization assigns it to the nearest code in the codebook by argmax operation:

$$k^\star = \arg\min_k \|\mathbf{z} - \mathbf{c}_k\|_2, \tag{9}$$

and the quantized output is:

$$\mathbf{c}_q = \mathbf{c}_{k^\star}. \tag{10}$$

The above argmax assignment is typically referred to as *deterministic quantization*, where identical input is always assigned to the same codeword. Additionally, some methods [6, 165, 195, 255] employ the Gumbel-Softmax operation to introduce stochasticity or noise during training, named *stochastic quantization*, where it assigns codewords based on probability distribution and can assign different codewords for identical input, helping to escape local optima.

### *2.1.3 Codebook Updating*

Updating the codebook during training is critical to ensure it remains representative and stable. In addition to codebook loss, one commonly adopted approach is EMA (exponential moving average) updating.

**Codebook Loss.** The codebook can be updated by codebook loss, second term in Eq. (1), which pulls codewords toward the encoder outputs. This loss encourages the codebook to better cover the distribution of encoded features and improves quantization quality.

**Exponential Moving Average (EMA) Updating.** EMA strategy updates the codebook by progressively reflecting the distribution of encoder outputs through running averages that track both the assignment counts and the cumulative encoder outputs for each codevector $\mathbf{c}_i$ [148, 167, 168].

For each training step $t$, the following statistics are updated:

$$N_i^{(t)} := \gamma N_i^{(t-1)} + (1-\gamma) n_i^{(t)}, \tag{11}$$

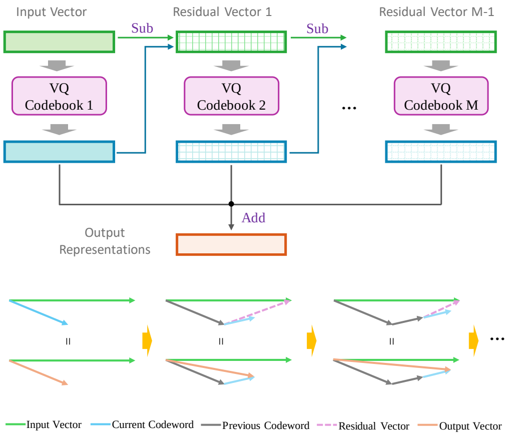

Fig. 5. Illustration of RVQ with multi-stage quantization. Each stage quantizes the residual vector from the previous stage (top), progressively approximating the input vector as shown in the geometric visualization of vector operations (bottom).

$$m_i^{(t)} := \gamma m_i^{(t-1)} + (1-\gamma)\sum_{j=1}^{n_i^{(t)}} E(x)_{i,j}^{(t)}, \tag{12}$$

where $E(x)$ denotes the encoder output vectors in the current mini-batch, $n_i^{(t)}$ is the number of vectors in $E(x)$ which will be quantized to the codevector $\mathbf{c}_i$, and $\gamma$ is a decay parameter (typically 0.99).

After accumulating these statistics, the codevector $\mathbf{c}_i$ is updated as:

$$\mathbf{c}_i^{(t)} := \frac{m_i^{(t)}}{N_i^{(t)}}. \tag{13}$$

This EMA approach ensures that the codebook evolves smoothly during training by integrating information across multiple mini-batches, leading to a more stable and representative set of codevectors. Furthermore, Roy et al. [168] introduces a soft Expectation Maximization (EM) algorithm, which assigns the input embedding to a probabilistic distribution over codevectors and updates the involved codevectors, instead of updating only the nearest codevector.

The above vanilla vector quantization serves as the basis for a wide range of quantization techniques, where it maps inputs to the nearest codeword to achieve compact and discrete representations. We also discuss the prevalent codebook collapse issue in vector quantization at the end of this section, especially representative solutions under the vanilla VQ mechanism, such as HQA [216], CVQ-VAE [268], VQ-WAE [191], SQ-VAE [177], and HQ-VAE [178].

## 2.2 Residual Vector Quantization

Residual Vector Quantization (RVQ) [9, 24] introduces a multi-stage quantization mechanism to gradually reduce the quantization error. As depicted in Fig. 5, instead of mapping the input vector to a single codeword, RVQ encodes the input through a sequence of residual quantization stages, where each stage encodes the quantization residual from the previous stage.

**Definition [Residual Vector Quantization].** *Let $\mathbf{z} \in \mathbb{R}^D$ be a $D$-dimensional input vector, and $\{\mathcal{C}^{(i)}\}_{i=1}^{M}$ be a set of $M$ codebooks, where each codebook $\mathcal{C}^{(i)} = \{\mathbf{c}_1^{(i)}, \ldots, \mathbf{c}_{K_i}^{(i)}\} \subseteq \mathbb{R}^D$ contains $K_i$ codewords. Residual Vector Quantization defines $M$ sequential quantization stages. For the stage $(i+1)$, RVQ quantizes the residual vector $\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \mathbf{c}_{k^*}^{(i)}$ to its nearest codeword $\mathbf{c}_{k^*}^{(i+1)}$ in the $(i+1)$-th codebook $\mathcal{C}^{(i+1)}$, in particular, the first residual $\mathbf{r}^{(1)} = \mathbf{z}$. The final quantized output $\mathbf{z}_q$ is obtained by summing the selected codewords: $\mathbf{z}_q = \sum_{i=1}^{M} \mathbf{c}_{k^*}^{(i)}$.*

### 2.2.1 Codebook Structure and Optimization

The effectiveness of RVQ heavily depends on the structure and optimization of its stage-wise codebooks. SQ [131] presents a hierarchical dependency structure among subcodebooks, utilizing greedy coarse-to-fine encoding, and employing hierarchical k-means and top-down refinement to initialize and update subcodebooks, thereby reducing quantization error. In [114], IRVQ combines subspace clustering with warm-started k-means to learn high-entropy codebooks and introduces a multi-path encoding strategy that mitigates greedy encoding errors. On the other hand, ERVQ [3] proposes a joint optimization to iteratively optimize all stage codebooks by the others, instead of training them sequentially. Liu et al. [115] propose the generalized RVQ (GRVQ) by introducing transition clustering to improve k-means and multipath encoding for lower quantization error. CompQ [151] introduces a competitive quantization to jointly train all codebooks via redefining ”winner codevector” and stochastic gradient descent.

### 2.2.2 Projected or Transformed Residual Quantization

RVQ can be enhanced by applying projections or transformations to the residual vectors, improving alignment and quantization accuracy across stages. PRVQ [215] enhances RVQ by incorporating PCA projections with dimensionality reduction before residual quantization, ensuring that the discarded projection information is retained and used. Transformed Residual Quantization (TRQ) [245] introduces cluster-wise transforms in RVQ by learning a local rotation matrix for each residual cluster and aligns residual vectors via the proposed iterative alignment (IA) to reduce quantization noise and improve subsequent quantization accuracy. Guo et al. [55] optimize RVQ by projection of data with an orthogonal matrix, proposing the non-parametric RVQ-NP and the parametric RVQ-P.

### 2.2.3 Implicit and Neural Codebook Generation

Implicit and neural methods construct RVQ codebooks in a data-adaptive manner during quantization. QINCo [70] replaces the fixed codebooks in RVQ with neural networks that generate step-specific codebooks conditioned on partial reconstructions, allowing the codebooks to adapt to residual distributions. QINCo2 [187] further improves QINCo by introducing the codeword pre-selection with beam search for improved vector encoding, a lookup-based decoder for efficient large-scale search, and an optimized training procedure and network architecture.

## 2.3 Product Quantization

Product Quantization (PQ) [74, 135] decomposes the original vector space into multiple lower-dimensional subspaces and quantizes each subspace independently, as illustrated

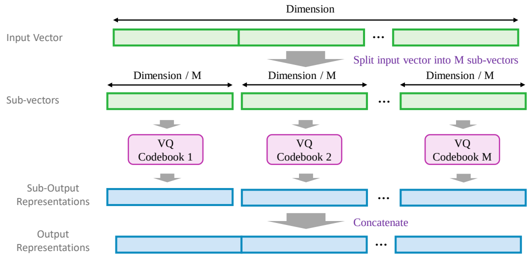

Fig. 6. Illustration of PQ. Each sub-vector of the high-dimensional vector is quantized independently in its own subspace.

in Fig. 6. This approach drastically reduces quantization error while maintaining compact representations, and is particularly effective in high-dimensional scenarios.

**Definition [Product Quantization].** *Let $\mathcal{Z} \subseteq \mathbb{R}^D$ be the input space, and let $\mathbf{z} \in \mathcal{Z}$ be a $D$-dimensional input vector. Product Quantization partitions $\mathbf{z}$ into $M$ disjoint sub-vectors: $\mathbf{z} = [\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}]$, where each $\mathbf{z}^{(m)} \in \mathbb{R}^{D/M}$. For each subspace, a separate sub-codebook $\mathcal{C}^{(m)} = \{\mathbf{c}_1^{(m)}, \ldots, \mathbf{c}_K^{(m)}\} \subseteq \mathbb{R}^{D/M}$ is trained. The overall codebook $\mathcal{C}$ is then defined as the Cartesian product $\mathcal{C} = \mathcal{C}^{(1)} \times \mathcal{C}^{(2)} \times \cdots \times \mathcal{C}^{(M)}$, resulting in $K^M$ possible composite codewords. The sub-vector $\mathbf{z}^{(m)}$ is quantized independently by mapping it to its nearest sub-codeword $\mathbf{c}_{k^*}^{(m)}$ in m-th subspace. The final quantized output $\mathbf{z}_q$ is given by concatenating those codewords:*

$$\mathbf{z}_q = [\mathbf{c}_{k^*}^{(1)}, \mathbf{c}_{k^*}^{(2)}, \ldots, \mathbf{c}_{k^*}^{(M)}]. \tag{14}$$

### 2.3.1 Space Decomposition Optimization

Optimizing subspace decomposition plays a central role in enhancing PQ performance by reducing quantization distortion. Optimized Product Quantization (OPQ) [46] considers the optimal space decomposition issue and transforms the input space by a rotation matrix $\mathbf{R}$, allowing optimal subspace partitioning. Locally Optimized Product Quantization (LOPQ) [84] employs a coarse quantizer to assign data to cells, then independently optimizes a rotation matrix and a product quantizer to encode residuals within each cell. CKM [147] optimally rotates the original space, enabling lower distortion. OCKM [196] further optimizes CKM by introducing multiple sub-codebooks in each subspace and the multi-codeword selection in each sub-codebook.

### 2.3.2 Codebook Structure and Update

The design and maintenance of sub-codebooks are crucial for ensuring efficient and accurate quantization in PQ. Online PQ [227] presents two budget constraints to update the partial codebooks incrementally. In addition, Online OPQ [104] extends to dynamically update the quantization codebooks and the rotation matrix via the Orthogonal Procrustes problem. Residual Vector Product Quantization (RVPQ) [146] introduces residual codebooks within each subspace and optimizes them jointly, enhancing the quantization structure. On the other hand, Variance-Aware Quantization (VAQ) [156] adapts codebook sizes to subspaces based on subspace importance via linear dimensionality reduction, and Product Tree Quantization (PTQ) [244] introduces the tree-structured codebooks and relaxes the subspace independence assumption of PQ, thereby reducing distortion. Recently, HiHPQ [159] proposes a hyperbolic product quantizer by a Cartesian product of hyperbolic subspaces and a soft hyperbolic codebook quantization based on Lorentzian distance.

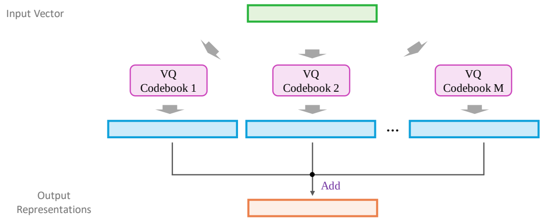

Fig. 7. Illustration of AQ. The input vector is quantized via multiple full-dimensional codebooks without dimension split.

### 2.3.3 End-to-end Learning-Based PQ

End-to-end learning-based PQ enables joint optimization of quantization and task-specific objectives through differentiable formulations. Differentiable Product Quantization (DPQ) [22] jointly learns discrete codes and task-specific objectives in an end-to-end differentiable manner via a differentiable softmax operation and a centroid-based approximation. Deep Product Quantization (DPQ) [86] introduces a supervised end-to-end learnable PQ framework that leverages the supervised signal to learn soft and hard representations through a direct-through estimator jointly. Differentiable Optimized Product Quantization (DOPQ) [122] optimizes the non-differentiable argmax operation based on direct loss minimization for end-to-end training.

### 2.3.4 Generalized Product Quantization

Composite Quantization (CQ) [259] can be viewed as a generalized formulation of PQ. Unlike PQ, CQ has no subspace decomposition and the orthogonality constraint, and introduces a constant interdictionary-element-product constraint between codebooks. When the codebooks are constrained to be mutually orthogonal and codewords are zero-padded outside the designated subspace, CQ degenerates to PQ.

**Definition [Composite Quantization].** *Let $\mathcal{Z} \subseteq \mathbb{R}^D$ be the input space and let $\mathbf{z} \in \mathcal{Z}$ be a $D$-dimensional input vector. Composite Quantization quantizes $\mathbf{z}$ in the original space as a summation of selected codewords $\mathbf{c}_{k^*}^{(m)}$ from $M$ global codebooks $\{\mathcal{C}^{(m)}\}_{m=1}^M$, where each codebook $\mathcal{C}^{(m)} = \{\mathbf{c}_1^{(m)}, \ldots, \mathbf{c}_K^{(m)}\} \subseteq \mathbb{R}^D$ contains $K$ codewords. In addition, codebooks satisfy a constant inter-dictionary-element-product constraint, i.e.,*

$$\langle C_i^\top, C_j \rangle = \xi, \quad \forall i \neq j \in \{1, 2, \ldots, M\}, \tag{15}$$

*where $\langle \cdot, \cdot \rangle$ denotes the inner product, $\xi$ is a constant, and CQ degenerates to PQ when $\xi = 0$. The quantized output $\mathbf{z}_q$ is $\mathbf{z}_q = \sum_{m=1}^M \mathbf{c}_{k^*}^{(m)}$.*

Zhang et al. [260] introduce the sparse CQ (i.e., SQ) to construct sparse codebooks via the constant inter-dictionary-element-product constraint and the sparsity regularization. Supervised Quantization (SQ) [204] is a supervised CQ method through quantization of the input in a linearly transformed discriminative subspace.

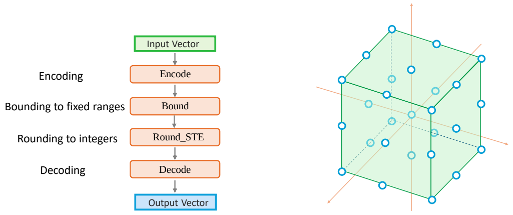

Fig. 8. Illustration of FSQ. Each dimension of $D$-dimensional input vector is bounded and rounded to $L$ corresponding integers (left). The formed codebook has a size $L^D$, like the hypercube visualization of codeword distribution for $D = 3$ and $L = 3$ (right).

## 2.4 Additive Vector Quantization

Additive Vector Quantization (AQ) [5] quantizes the input as a sum of codewords selected from multiple full-dimensional codebooks, as illustrated in Fig. 7.

**Definition [Additive Vector Quantization].** *Let $\mathcal{Z} \subseteq \mathbb{R}^D$ be the input space, and let $\mathbf{z} \in \mathcal{Z}$ be a D-dimensional input vector. Additive Vector Quantization quantizes $\mathbf{z}$ as a summation of $M$ codewords $\{\mathbf{c}_{k^*}^{(m)}\}_{m=1}^M$ selected from $M$ codebooks $\{\mathcal{C}^{(m)}\}_{m=1}^M$, where each codebook $\mathcal{C}^{(m)} = \{\mathbf{c}_1^{(m)}, \ldots, \mathbf{c}_K^{(m)}\} \subseteq \mathbb{R}^D$. The quantized output $\mathbf{z}_q$ is $\mathbf{z}_q = \sum_{m=1}^M \mathbf{c}_{k^*}^{(m)}$.*

Babenko and Lempitsky [5] introduce the additive product quantization (APQ) method that uses OPQ optimization to rotate the data and then applies AQ encoding to different parts of the rotated vector. The local search quantization (LSQ) [132] enhances AQ by incorporating iterated local search (ILS) to efficiently handle the NP-hard encoding problem, and enforces the sparsity of the codebooks. On the other hand, LSQ++ [133] improves LSQ by introducing a fast codebook update for a lower running time and stochastic relaxation techniques for greater recall. In addition, Online AQ [110] dynamically updates codebooks for streaming data, and introduces a randomized block beam search algorithm to assign discrete codes to incoming data efficiently, with a better regret bound than online PQ.

## 2.5 Finite Scalar Quantization

Finite Scalar Quantization (FSQ) [136] projects latent inputs to a few dimensions (typically less than 10) by the final encoder layer and quantizes each dimension independently to a small set of fixed scalar values by rounding to integers, using STE to propagate gradient through the non-differentiable rounding operation, as illustrated in Fig. 8. FSQ is a simple yet effective alternative to vector quantization in VQ-VAEs without any auxiliary losses and codebook collapse issue.

**Definition [Finite Scalar Quantization].** *Given a D-dimensional input vector $\mathbf{z} = [z_1, z_2, \ldots, z_D] \in \mathbb{R}^D$ (typically with $D < 10$), Finite Scalar Quantization quantizes each dimension $z_i$ into one of $L$ values $\{-\lfloor L/2 \rfloor, \ldots, \lfloor L/2 \rfloor\}$. Specially, for each dimension $z_i$, FSQ firstly applies a bounding function $f(\cdot)$ (e.g., $f(z_i) = \frac{L}{2} \cdot \tanh(z_i)$), and then rounds to integers, i.e.,*

$$q(z_i) = \mathrm{round}(f(z_i)) \in \{-\lfloor \frac{L}{2} \rfloor, \ldots, \lfloor \frac{L}{2} \rfloor\}. \tag{16}$$

*The final quantized output $\hat{\mathbf{z}}$ is $\hat{\mathbf{z}} = [q(z_1), \ldots, q(z_D)]$. For the vector $\mathbf{z} \in \mathbb{R}^D$, there are $L^D$ possible quantization outcomes, forming the implicit codebook with the size $L^D$.*

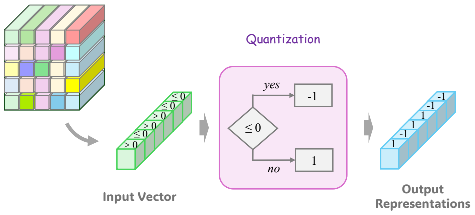

Fig. 9. Illustration of LFQ. Each dimension of an input vector is directly quantized into 1 or -1.

## 2.6 Look-up Free Quantization

Unlike the above VQ-based approaches such as vanilla VQ and RVQ, which need to look up $K$ D-dimensional codewords to find the nearest neighbor in the codebook for quantization, Lookup-Free Quantization (LFQ) [242] directly maps the input to a binary integer set without look-up, as illustrated in Fig. 9.

**Definition [Look-up Free Quantization].** *Given an D-dimensional input vector $\mathbf{z} = [z_1, z_2, \ldots, z_D] \in \mathbb{R}^D$, Lookup-Free Quantization constructs an implicit codebook $\mathcal{C}$ as the Cartesian product of D binary sets:*

$$\mathcal{C} = \times_{i=1}^D \mathcal{C}_i, \quad \text{where } \mathcal{C}_i = \{-1, +1\}, |\mathcal{C}| = 2^D. \tag{17}$$

*Each dimension $z_i$ is quantized independently into binary codebook $\mathcal{C}_i$ via the sign function sign$(\cdot)$:*

$$q(z_i) = \mathit{sign}(z_i) = -1 \cdot \mathbb{I}_{[z_i \leq 0]} + 1 \cdot \mathbb{I}_{[z_i > 0]}, \tag{18}$$

*where $\mathbb{I}_{[\cdot]}$ is the indicator function. The quantized binary code $q(\mathbf{z}) \in \{-1, +1\}^D$ defines a unique codeword in $\mathcal{C}$, and the corresponding token index is given by:*

$$\mathit{Index}(\mathbf{z}) = \sum_{i=1}^{D} 2^{i-1} \cdot \mathbb{I}_{[z_i > 0]}. \tag{19}$$

LFQ thus avoids explicit codebook lookup and enables efficient discrete tokenization with binary latent representations, growing the vocabulary size in a way.

## 2.7 Binary Spherical Quantization

Binary Spherical Quantization (BSQ) [264] employs a spherical projection-based quantization with binary encoding. An illustrative comparison between FSQ, LFQ, and BSQ in 2D is shown in Fig. 10.

Compared with LFQ, BSQ has bounded reconstruction error by constraining the codebook on the unit hypersphere, enabling faster convergence for large-scale visual and video modeling.

**Definition [Binary Spherical Quantization].** *Given an D-dimensional input vector $\mathbf{z} \in \mathbb{R}^D$, $\mathbf{z}$ is linearly projected into $\mathbf{v} = \mathit{Linear}(\mathbf{z}) \in \mathbb{R}^L$, where $L \ll D$, and then normalized on a unit sphere by $\ell_2$ normalization, i.e., $\mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2} \in \mathbb{S}^{L-1}$. Binary Spherical Quantization defines an implicit codebook $\mathcal{C} = \left\{-\frac{1}{\sqrt{L}}, \frac{1}{\sqrt{L}}\right\}^L \subseteq \mathbb{S}^{L-1}$, where each codeword $\mathbf{c} \in \mathbb{R}^L$ satisfies $\|\mathbf{c}\|_2 = 1$. BSQ quantzes $\mathbf{u}$ along each dimension via*

$$\mathbf{c}_k = \hat{\mathbf{u}} = \frac{1}{\sqrt{L}} \cdot \mathit{sign}(\mathbf{u}) \in \mathcal{C}, k = \sum_{i=1}^{L} \mathbb{I}_{[v_i > 0]} \cdot 2^{i-1}, \tag{20}$$

*where sign$(\cdot)$ is sign function with sign(0) $= 1$, and $\mathbb{I}_{[\cdot]}$ is the indicator function.*

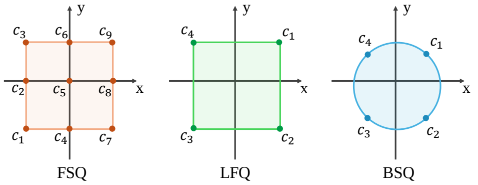

Fig. 10. Comparison of FSQ, LFQ, and BSQ in the 2D space. While the codewords of FSQ and LFQ partition the space into axis-aligned hypercubic cells, the codewords of BSQ are uniformly distributed on the unit hypersphere.

TABLE 1
Comparison of Codebook Collapse across Quantization Methods.

| Method | Collapse Mitigation | Reason / Comment |
|---|---|---|
| VQ (§2.1) | No | Learnable codebook easily collapses if not regularized |
| RVQ (§2.2) | No | Each stage has a learnable codebook; collapse still occurs |
| PQ (§2.3) | Partially | Multiple sub-codebooks reduce impact, but each can still collapse |
| AQ (§2.4) | Partially | Additive structure softens collapse impact, but doesn't prevent it |
| FSQ (§2.5) | Yes | Implicit fixed codebook; uses fixed scalar values |
| LFQ (§2.6) | Yes | Implicit fixed codebook; binary quantization |
| BSQ (§2.7) | Yes | Implicit fixed codebook; binary quantization on unit sphere |
| GART (§2.8) | Yes | Uses shared anchors + relation types, avoiding the codebook |

### 2.8 Graph Anchor-Relation Tokenization

Graph Anchor-Relation Tokenization (GART) exploits the anchor-based graph representation learning technique, and tokenizes nodes into a composition of selected anchor nodes and relational context to form compact and discrete representations [21, 42, 96]. This tokenization design drastically reduces the vocabulary size while retaining expressiveness, especially for knowledge graphs.

**Definition [Graph Anchor-Relation Tokenization].** *Let $\mathcal{V}$ denote the set of nodes in a graph and $\mathcal{R}$ the set of relation types between nodes where $\mathcal{V} \ll |\mathcal{R}|$. The codebook with size $M + |\mathcal{R}|$ is contructed by $M$ anchors and $|\mathcal{R}|$ relation types, where the anchor set $\mathcal{A} = \{a_1, a_2, \ldots, a_M\}$ is pre-selected from $\mathcal{V}$ according to certain strategies and $M \ll |\mathcal{V}|$. Given a target node $v \in \mathcal{V}$, it is matched to a composition of k-nearest anchors from $\mathcal{A}$ and its d connected relations from $\mathcal{R}$, denoted as $\mathcal{W} = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}, r_{j_1}, r_{j_2}, \ldots, r_{j_d}\}$ where $a_{i_k} \in \mathcal{A}$ and $r_{j_d} \in \mathcal{R}$. Finally, node $v$ is tokenized into the codevector $E(\mathcal{W})$ through an encoder function $E(\cdot)$.*

In practice, the encoder can adopt MLP, Transformer, or GNNs [21, 42, 96]. For anchor selection strategies, random selection works well, with Personalized PageRank (PPR) and degree-based strategies as alternatives. And some methods also encode the distances between target nodes and anchors [42] or multi-hop neighbors [21] to retain topological information and semantic context.

### 2.9 Discussion

Despite its strengths, discrete quantization faces a critical challenge—*codebook collapse*, which limits codebook diversity and the expressiveness of the quantized representations.

**Codebook Collapse.** *Codebook collapse refers to a situation where only a small subset of codewords are actively utilized during training, while the majority remain unused and don't get updated, those called "dead codewords", resulting in low codebook utilization.*

Beyond multi-codebooks, hierarchical structures and codebook-free designs in Table 1, numerous methods have been developed to mitigate codebook collapse: ***(i) Code Reset.*** HQA [216] reinitializes unused codes near frequently used ones during training to mitigate under-utilization. CVQ-VAE [268] further introduces an online clustering strategy, dynamically reinitializing the unused codes based on running average statistics. ***(ii) Linear Reparameterization.*** Recent methods apply linear transformation over code vectors to optimize codebooks for the collapse issue. For example, Huh et al. [69] proposes an affine reparameterization of codes by shared mean and standard deviation, assigning affine parameters to enable gradients to flow through unused codes. VQGAN-LC [277] and SimVQ [278] employ a learnable linear layer and reparameterize the codes to ensure all codes remain active. ***(iii) Soft Quantization.*** Unlike nearest-neighbor hard assignments in standard VQ methods, soft quantization quantizes inputs as a weighted combination of codewords to improve codebook utilization. IBQ [173] applies STE on the one-hot categorical distribution between the encoded feature and codebook, letting all codes be selected equally. soft VQ-VAE [218] and SCQ [44] quantize inputs as a convex combination problem of codewords. SHVQ [2] and SQ-VAE [177] also introduce the anneal mechanism, gradually approaching hard assignment from soft quantization during training. Furthermore, HQ-VAE [178] incorporates a hierarchical structure into SQ-VAE to mitigate the collapse issue. ***(iv) Regularization.*** Several works introduce different regularization terms to improve codebook utilization. HC-VQ [190] proposes an entropy regularization based on persistent homology, indicating higher entropy of VQ latent space is associated with higher codebook utilization. On the other hand, Reg-VQ [255] introduces a prior distribution regularization where all codevectors are used, preventing collapse of the predicted token distribution. VQ-WAE [191] combines a KL-regularization with WS distance approximated entropic regularized dual form, to match the codebook with latent data distribution. Some methods [173, 242, 264] additionally add an entropy penalty to encourage codebook utilization.

## 3 Earlier Tokenization

Before the advent of LLMs, discrete tokenization, mainly via vector quantization, has been widely used for efficient data compression and representation learning. This section reviews applications in image, audio, video, graph, and recommendation systems [19, 157, 181, 247, 275], which laid the groundwork for modern multimodal systems by demonstrating the effectiveness of quantized representations across diverse data types, and continue to provide transferable insights and readily adaptable techniques for LLM-based multimodal modeling.

### 3.1 Image

Discrete tokenization has been widely used in image retrieval [73, 197], generation [214, 243], and representation learning [8, 35]. Quantized visual tokens enabled compact and expressive image modeling.

***(i) Image Retrieval.*** DVSQ [15] jointly learns visual-semantic embeddings and quantizers for efficient image retrieval with compact binary codes. Deep Progressive Quantization (DPQ) [43] learns codes of varying lengths by progressively approximating the feature space for large-scale image retrieval. In [73], SPQ achieves self-supervised product quantization via cross-quantized contrastive learning for unsupervised image retrieval. MeCoQ [197] introduces contrastive unsupervised quantization with code memory for reduced drift and regularization to prevent degeneration.

***(ii) Image Synthesis.*** MaskGIT [17] tokenizes images via VQ-GAN and uses a bidirectional Transformer decoder to predict masked tokens for synthesis. On the other hand, RQ-Transformer [93] and DnD-Transformer [20] quantize feature maps of images by RQ-VAE based on RVQ for 2D autoregressive generation. In addition, ViT-VQGAN [239] and Efficient-VQGAN [14] replace the CNN with a vision Transformer for improved reconstruction, with ViT-VQGAN also introducing factor and $\ell_2$-normalized codes for better codebook usage. Both MQ-VAE [66] and DQ-VAE [65] consider the codebook redundancy issue caused by ignoring different perceptual importance of image regions. TiTok [243] and FlowMo [172] are 1D tokenizer, tokenizing images into 1D latent codes. In particular, FlowMo innovatively employs a transformer-based diffusion autoencoder. MaskBit [214] enables the image generation without embedding via LFQ, generating bit tokens directly without learning new embeddings. To unify image generation and representation learning, additional methods have also made efforts [54, 98, 99, 184, 199, 269], more details can be found in Appendix A.

***(iii) Image Classification.*** BEiT [8] introduces BERT-style masked image modeling for vision Transformers by predicting discrete tokens from masked patches. Building on BEiT, BEiT v2 [158] proposes VQ-KD to train a semantic visual tokenizer, pushing MIM beyond pixel-level targets. Further exploring tokenizer design, ClusterMIM [35] introduces a label-free clustering tokenizer for MIM and the TCAS metric to evaluate its quality.

## 3.2 Audio

Recent developments in audio modeling leverage discrete tokenization for efficient compression and self-supervised representation learning [6, 7], primarily through neural codecs [78, 108, 249] and quantized speech tokens.

***(i) Self-Supervised Speech Representation via Discrete Units.*** VQ-wav2vec [6] and Wav2vec [7] introduce BERT-style self-supervised contrastive paradigm for modeling speech representations from raw audio.

***(ii) High-Fidelity Audio Compression with Discrete Tokens.*** SoundStream [246] uses RVQ with structured dropout, trained by the VQ-GAN formulation for unified codec at variable bitrates. HiFi-Codec [233] introduces group-residual vector quantization with only four codebooks, Encodec [31] develops a multiscale spectrogram adversary and loss balancer, and DAC [89] adds periodic inductive biases. LMCodec [75] introduces a fully causal transformer with conditional entropy coding for low-bitrate speech codec.

***(iii) Semantic-Aware and General-Purpose Tokenization.*** SemantiCodec [108] consists of semantic and acoustic encoders, dual-layer vector quantization and a diffusion based decoder, supporting diverse audio types. Along same lines, SpeechTokenizer [226] unifies semantic and acoustic tokens for speech language modeling, hierarchically disentangling speech information across RVQ layers.

***(iv) Real-Time and Lightweight Audio Codecs.*** StreamCodec [78] is a streamable causal audio codec for real time communication with residual scalar-vector quantization to enhance codebook utilization. Similarly, SQCodec [249] designs single-quantizer architecture based on TConv module and FSQ for lightweight audio codec. A broader set of representative works and further details [16, 79, 90, 157] can be found in Appendix A.

## 3.3 Graph

Graphs are ubiquitous in domains such as knowledge graphs [170] and molecular systems [279]. Their non-Euclidean structure and the requirement for permutation invariance pose fundamental challenges for scalable and effective modeling [12, 13, 42, 125, 279]. To address these issues, discrete tokenization techniques have emerged as a compact and interpretable alternative for graph representation, enabling scalable modeling and structure-aware representation learning.

***(i) Graph Representation Compression.*** TS-CL [170] and LightKG [194] leverage discrete codes to compress knowledge graph embeddings for efficient storage and inference. Similarly, SNEQ [60] and d-SNEQ [61] learn low-dimensional network embeddings under semi-supervised settings by self-attention-based and autoencoder-based PQ, respectively. NID [125] learns compact discrete node codes by compressing GNN layers, enabling interpretable graph tokenization.

***(ii) Molecular Representation Learning.*** iMoLD [279] learns distribution-invariant molecular representations via a first-encode-then-separate paradigm and task-agnostic self-supervised objective. Similarly, MOLE-BERT [222] introduces a context-aware tokenizer with group VQ-VAE and a joint pretraining framework combining masked atom modeling and contrastive learning. For efficient PPI modeling, MAPE-PPI [219] encodes protein microenvironments into discrete codes and masks the codebook.

***(iii) Graph Generation.*** DGAE[12] and GLAD[13] both improve permutation-invariant graph generation in discrete latent spaces, where DGAE introduces a graph-to-set autoencoder with an autoregressive 2D-Transformer, while GLAD introduces a diffusion model with diffusion bridges. Additionally, Appendix A further discusses representative methods with varied design choices and objectives [200, 212, 236, 247, 254] on other directions like graph transformers.

## 3.4 Video

Discrete tokenization is also an essential component in video modeling, enabling compact representations of spatio-temporal information. Recent work explores its use in video synthesis, compression, and unified representation learning across diverse temporal scales. VideoGPT [231] leverages VQ-VAE with 3D convolutions to obtain spatio-temporally aware discrete latent representations of videos. To better support flexible long video generation, TATS [45] uses a time-agnostic VQGAN to tokenize videos into temporally agnostic codes. MAGVIT [240] introduces a 3D tokenizer that quantizes videos into spatiotemporal tokens for unified video synthesis. As a subsequent extension, MAGVIT-v2 [242] shows that strong visual tokenizers enable autoregressive LMs to outperform diffusion models in image and video generation. Phenaki [189] proposes a discrete video tokenizer with causal temporal attention to compress variable-length videos into compact token sequences. OmniTokenizer [198] introduces a spatial-temporal transformer and progressive training for joint image and video tokenization. Building on unified visual modeling, TVC [272] combines discrete and continuous token compression for ultra-low bitrate video reconstruction with high fidelity. In a similar vein, BSQ-ViT [264] is a unified image-video tokenizer using a transformer with block-wise causal masking and BSQ for variable-length inputs. Several additional efforts have been made from different perspectives [179, 180, 195, 228], and further details are in Appendix A.

### 3.5 Action

Discrete tokenization has also been explored in action modeling, particularly for encoding continuous control signals into compact action tokens. Early efforts focus on quantization for reinforcement learning and efficient temporal abstraction. SAQ [123] introduces a VQ-VAE [148]-based offline RL framework that discretizes actions by state, enabling more stable policy learning. To enhance temporal abstraction in control, PRISE [271] employs VQ for action discretization and designs byte pair encoding (BPE) to extract skill tokens for efficient sequence modeling.

### 3.6 Multiple Modalities

**(a) Text + Image.** In text-image tasks, discrete tokenization serves as a bridge between visual and linguistic modalities. It enables unified token spaces for text-to-image generation and multimodal representation learning.

***(i) Unified Discrete Token Spaces for Generation.*** DALL-E [165] and CogView [34] both model text and image tokens jointly via a transformer for text-to-image generation, where image tokens are obtained through dVAE and VQ-VAE [148], respectively. Incorporating VQ-VAE [148] with DDPM in the token space, VQ-Diffusion [53] enables efficient, high-fidelity text-to-image generation. For more controllable generation, Make-A-Scene [41] uses discrete prompts and layouts to guide aligned scene construction. Unified-IO [120] unifies vision and language tasks by converting all inputs and outputs—whether images, masks, or text—into discrete sequences for unified sequence modeling. Muse [18] models masked VQ tokens conditioned on text and reconstructs them iteratively in base and super-resolution stages. Focusing on tokenizer design, UniTok [126] introduces a unified tokenizer with multi-codebook quantization for high-fidelity generation and semantic understanding. Recently, HART [181] combines discrete VQ and residual continuous tokens for efficient high-resolution image generation.

***(ii) Language-Guided Tokenization and Alignment.*** NUWA-LIP [144] improves language-guided image inpainting via a defect-free VQGAN [39], fusing semantic and visual cues. Similarly, TexTok [248] introduces a text-conditioned tokenizer that improves both continuous and discrete tokenization quality. LG-VQ [57] and TokLIP [101] both align visual tokens with textual semantics to enhance multimodal understanding, through language-guided codebook learning [57] and CLIP-aligned token encoders [101], respectively. Extending to knowledge graphs, MyGO [262] tokenizes multimodal data into fine-grained tokens and boosts entity representations via contrastive learning.

**(b) Text + Audio.** Discrete tokenization has been explored for aligning textual and acoustic modalities, particularly in text-to-speech synthesis. These methods leverage quantized speech representations to enable controllable, high-fidelity generation and efficient language-to-audio modeling.

***(i) Tokenization-Driven Generative Modeling.*** Specifically, AudioGen [88] proposes a text and audio mixing augmentations for text-to-audio generation, improving compositionality. Similarly, NaturalSpeech 3 [83] introduces a factorized diffusion model for TTS on disentangled subspaces via a factorized neural speech codec (FACodec). To enable high-quality TTS, Spectral Codec [91] and Single-Codec [95] tokenize mel-spectrograms using FSQ and single-codebook VQ-VAE, respectively. The SimpleSpeech series [234, 235] focuses on efficient TTS with scalar quantization and diffusion. Concretely, SimpleSpeech [235] proposes scalar-quantized speech codec (SQ-Codec) and transformer-based diffusion, while SimpleSpeech 2 [234] further introduces Time MoE and flow-based diffusion.

***(ii) Codec-Based Language Modeling.*** VALL-E [192] and VALL-E X [263] models text to speech synthesis (TTS) as conditional language modeling on discrete codec tokens in monolingual and cross-lingual settings, respectively, enabling in-context learning capabilities in zero-shot scenarios. To enhance robustness, RALL-E [225] introduces chain-of-thought (CoT) prompting to improve the realiability of TTS generation. Further, HALL-E [145] introduces a post-training approach which hierarchically reorganizes discrete tokens through knowledge distillation, reducing frame rate for minute-long TTS.

**(c) Audio + Video.** Discrete tokenization has also been applied to joint audio-video modeling, enabling unified representations for multimodal tasks. Initial efforts demonstrate its potential in audiovisual understanding. VQ-MAE-AV [171] introduces a vector-quantized masked autoencoder for audiovisual speech emotion recognition, which learns discrete audio-visual speech representations via self-supervised multimodal fusion.

**(d) Audio + Action.** In audio-action tasks, discrete tokenization enables mapping speech to compact motion representations. A representative approach is outlined below. ProTalk [117] introduces a PQ-based non-autoregressive framework for generating diverse and coordinated full-body co-speech motions, integrating structured quantization and motion refinement for realism.

**(e) Audio + Image + Video.** In multimodal synthesis involving audio, image, and video, discrete tokenization enables compact control over facial motion. For instance,

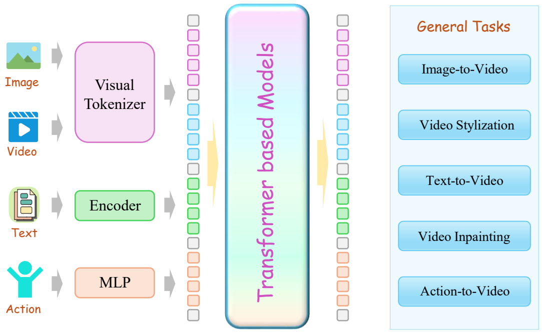

Fig. 11. Non-LLM-based multimodal pipeline that encodes each modality via specialized modules before fusion through transformer-based models for different tasks [206].

VQTalker [116] employs group residual scalar quantization for facial motion tokenization, enabling high-fidelity multilingual talking head synthesis at low bitrates.

**(f) Text + Image + Video + Action.** Token-based representations have been extended to unify text, image, video, and action modalities, supporting the modeling of complex spatio-temporal dynamics within a shared discrete space. As illustrated in Fig. 11, such systems typically adopt modality-specific tokenizers followed by a transformer-based fusion module. WorldDreamer [206] models various video generation as masked visual token prediction by proposed spatial temporal patchwise transformer on discrete visual tokens across general world physics and motions.

**(g) Recommendation Systems.** Discrete tokenization in recommendation systems supports compact modeling of behaviors and semantics, enabling more efficient, transferable, and generative recommendation frameworks. Specifically, MGQE [85] extends DPQ [22] with variable capacities to handle power-law distribution, learning compact embeddings for recommendation. ReFRS [71] and VQ-Rec [64] both employ vector-quantized representations for sequential recommendation. ReFRS [71] emphasizes privacy-preserving federated learning via VQ-VAE-based temporal embeddings and semantic clustering, while VQ-RecVQ-Rec [64] focuses on transferability through contrastive pretraining and permutation-based OPQ alignment. TIGER [164] and CoST [275] both target generative recommendation via semantic tokenization. Concretely, TIGER [164] adopts RQ-VAE [93] to represent items as semantic IDs and autoregressively predicts the next item, while CoST [275] improves token quality through contrastive quantization. In [210], EAGER integrates behavior and semantic tokens via contrastive learning and semantic-guided transfer in a two-stream framework.

## 4 LLMs with Single Modality

LLMs have demonstrated remarkable capabilities in generation, understanding, and generalization across various tasks, making them an attractive backbone for modeling other non-text modalities. To benefit from these powerful capabilities of LLMs, recent studies [59, 107, 210] have explored how to encode single non-text modalities into LLM-readable tokens via discrete tokenization, e.g., mapping data features into LLMs' vocabulary space without explicit text inputs [241]. This section reviews how such discrete tokens serve as a bridge that allows LLMs to complete downstream tasks like node classification [118] and recommendation [238]. The left side of Fig. 12 illustrates the evolution of such applications across different single modalities and years. Among them, image and recommendation tasks dominate in volume. In terms of model choices, LLaMA [52, 139, 185] based LLMs are most frequently adopted, followed by T5 [50, 163] variants, Qwen [4, 232], PaLM 2 [49], GPT-series [149] models and so on. The key information and open source of these applications are summarized in Table 2 in Appendix.

**(a) Image.** Discrete tokenization enables LLMs to process visual inputs by converting image features into semantic tokens, supporting visual alignment, generation, and understanding. Both LQAE [107] and SPAE [241] leverage pretrained LLM vocabularies to discretize visual signals for efficient visual generation. LQAE [107] introduces a VQ-VAE [148]-style tokenizer that maps images to the token space of frozen LLMs, enabling few-shot multimodal tasks via direct token-level interaction, while SPAE [241] extends this idea with a semantic pyramid token structure that generates variable-length lexical tokens, enabling multimodal in-context learning. In addition, LlamaGen [176] applies the vanilla autoregressive model Llama to image generation, achieving high-quality image tokenization with a downsample ratio of 16. Similarly, StrokeNUWA [182] proposes a stroke token as a better visual representation, which is semantically rich, LLM-compatible, and highly compressed, enabling efficient vector graphic synthesis through LLMs. Both V2T Tokenizer [276] and V$^2$Flow [253] adopt LLM vocabularies as visual codebooks, enabling seamless integration with frozen LLMs. Concretely, V2T Tokenizer [276] introduces a global-local tokenization scheme to support visual understanding and denoising, while V$^2$Flow [253] incorporates a vocabulary resampler and rectified-flow decoder for high-quality autoregressive generation.

**(b) Audio.** In the audio domain, discrete tokenization has been explored to improve speech generation and recognition by mapping acoustic signals to LLM-compatible token sequences. TWIST [59] introduces a warm start from the pretrained LLM to initialize speech language models for speech generation. To improve stability and control, SSVC [134] disentangles speaker identity and linguistic content via self-supervised learning and residual vector quantization. In addition, JTFS LM [230] systematically compares discrete and continuous speech representations in LLM-based automatic speech recognition, showing that supervised discrete tokens offer robust performance and better alignment.

**(c) Graph.** In graph applications, discrete tokenization helps encode structural information into LLM-compatible tokens for integration and reasoning. Specifically, NT-LLM [76] employs graph anchors for node tokenization, selecting anchors via a greedy algorithm, and encoding nodes for LLM input based on anchor-based distance. Similarly, Dr.E [118] employs a dual-residual VQ-VAE to discretize graphs into tokens aligned with LLM vocabulary, enabling token-level integration of graph-structured data into LLMs through multi-view structural enhancement.

**(d) Action.** For action understanding, discrete tokenization

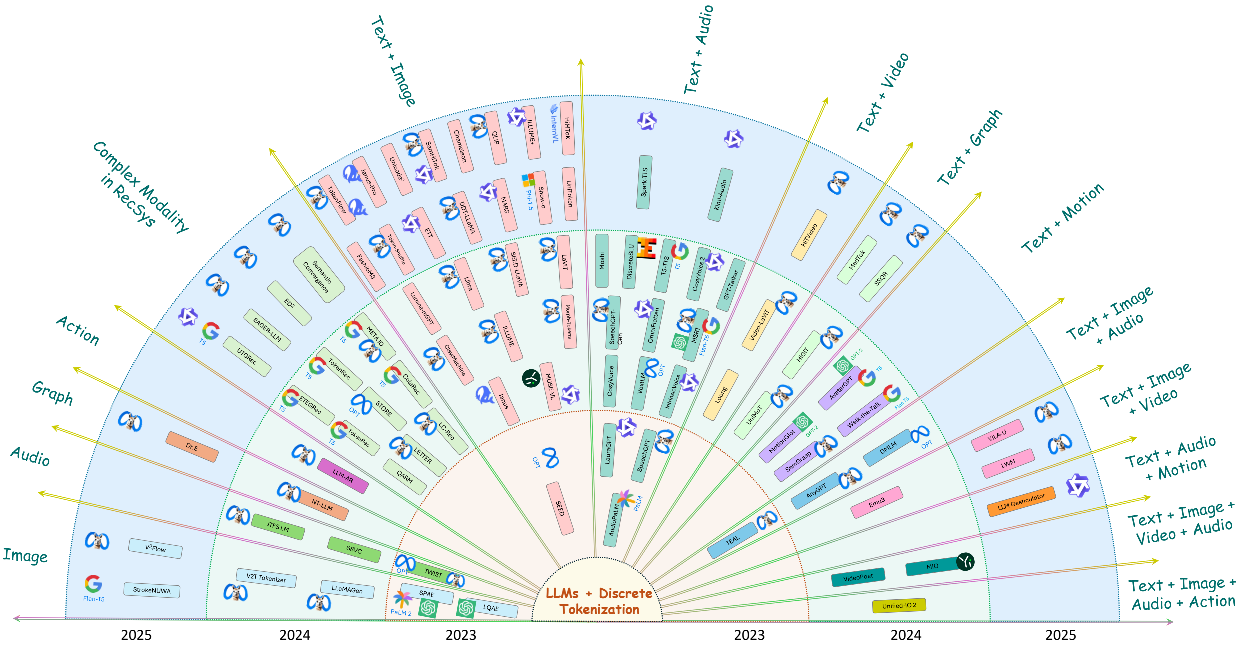

Fig. 12. Discrete tokenization with (M)LLMs emerges in 2023 and gains widespread adoption in 2024, especially in LLaMA [52, 139, 185]-based models. The trend continues to accelerate, showing strong momentum in research and applications.

has been used to convert motion sequences into structured token inputs for LLMs. LLM-AR [160] treats LLMs as action recognizers by projecting skeleton sequences into "action sentences" through a linguistic projection process, where the hyperbolic codebook is designed for the tree-like human skeleton representations.

**(e) Recommendation Systems.** In recommendation systems, discrete tokenization bridges collaborative and semantic signals, enabling LLMs to handle user-item interactions effectively.

***(i) Alignment-Based Semantic-Collaborative Tokenization.*** To unify item representations, several methods [202, 208, 266] align semantic content with collaborative signals for better tokenization and recommendation quality. Specifically, LC-Rec [266] introduces a tree-structured residual quantizer with uniform semantic mapping and leverages fine-tuning tasks to integrate collaborative semantics; LETTER [202] employs three regularizations to fuse hierarchical semantics, collaborative signals, and code diversity into learnable item tokens; and ColaRec [208] unifies content and interaction signals through an auxiliary indexing task and a contrastive loss for aligned token learning. To align input tokens with the representation space of LLMs, several methods [68, 94, 112] introduce auxiliary tokens or alignment modules to enhance compatibility and reasoning capacity. STORE [112] unifies semantic tokenization and generative recommendation using a single LLM; META ID [68] introduces out-of-vocabulary tokens via meta-path sampling to align user-item interaction information with LLMs; and Semantic Convergence [94] comprises an alignment tokenization module to synchronize item tokens with input semantic space of LLMs, and an alignment task module to fine-tune LLMs. In addition, QARM [124] discretizes aligned multi-modal representations into trainable code IDs for downstream tasks.

***(ii) Learnable ID Tokenization for Generative Recommendation.*** TokenRec [161] introduces a masked vector-quantized tokenizer to discretize user and item IDs into LLM-compatible tokens, capturing high-order collaborative knowledge for LLMs, and enables efficiency by only updating GNN. To enhance LLMs' comprehension towards tokens, ED$^2$ [238] introduces a dual dynamic index mechanism, unifying index generation and recommendation, and it designs a multi-grained token regulator. Further, EAGER-LLM [63] integrates endogenous and exogenous behavioral and semantic signals by dual-source knowledge-rich item indices and multiscale alignment reconstruction tasks. Recent methods [106, 267] unify tokenization and generation to enhance alignment and transferability. Specifically, ETEGRec [106] adopts a dual encoder-decoder architecture to jointly optimize item tokenization and autoregressive recommendation, while UTGRec [267] learns a universal item tokenizer across domains using a multimodal LLM and tree-structured codebooks for transferable generation.

## 5 LLMs with Multiple Modalities

While LLMs evolve into general-purpose agents, discrete tokenization makes it possible for LLMs to operate in multimodal contexts where modality-specific tokenizers can convert continuous signals to unified token sequences for LLM-based modeling. This section reviews multi-modality applications which demand more sophisticated alignment and integration for semantic consistency compared to single-modality applications. As shown in Fig. 12, early exploration began in 2023, followed by rapid expansion in 2024 across increasingly complex modality combinations. Among them, *Text + Image* has seen the most active development, followed by the integration of *Text + Audio*. Numerous LLM backbones have been developed, including LLaMA series [52, 139, 185], T5 [50, 163], Qwen [4, 232], DeepSeek [11, 30], Mistral [141], Vicuna [119], PaLM [29], and InternVL [27, 150]. The key information and source of these applications are summarized in Table 3 in Appendix.

**(a) Text + Image.** Text and image constitute the most common and extensively explored modality pair in multimodal learning. A key to empowering language models with visual capabilities is to integrate image inputs using the native modeling paradigm of LLMs. Recent studies approach this by discretizing visual signals for unified modeling.

***(i) Visual Tokenization for Multimodal Alignment.*** For alignment with left-to-right autoregressive modeling in LLMs, SEED [47] and SEED-LLaMA [48] generate 1D causally visual tokens by vector quantization, not conventional 2D representations. LaVIT [82] argues images should be tokenized into discrete tokens to enable LLMs to process images and text indiscriminately, and develops a dynamic variable-length tokenizer for images. Besides, many studies have focused on aligning visual tokens with language semantics through dedicated tokenizer design, like the discrete tokenizer with semantic constraints in MUSE-VL [224]. Designed as an early-fusion architecture, Chameleon [183] can generate interleaved textual and image contents by training mixed-modal discrete tokens. Also, ClawMachine [127] directly embeds discrete visual tokens into text for referential tasks, unifying visual referring and grounding without extra syntax. Recently, QLIP [265] introduces a BSQ [264]-based visual tokenizer aligned with text by contrastive and reconstruction learning. Beyond modality alignment and unification, some studies [23, 80, 162, 217] have also considered the gap of information granularities between generation and understanding. For instance, Janus series [23, 217] explores decoupled encoding pathways for understanding and generation, where Janus-Pro [23] further scales Janus [217] to a bigger model and data size. In [80], UniToken also combines VQ-based discrete tokens with continuous features via unified visual encoding. In addition, TokenFlow [162] decouples semantic and pixel representations through a dual-codebook design and aligns them by shared mapping, unifying understanding and generation.

***(ii) Generative Pretraining and Tokenizer Tuning.*** To improve the synergy between discrete visual tokens and LLMs, recent efforts focus on generative pretraining [105] and tokenizer-level optimization [203]. Lumina-mGPT [105] advances a multimodal generalist through unambiguous image representation with flexible supervised finetuning strategies. In addition, ETT [203] jointly trains the vision tokenizer and LLM by feeding codebook embeddings and applying token-level caption supervision.

***(iii) Diffusion-Enhanced Vision Decoding.*** Show-o [223] and MARS [62] both adopt autoregressive generation frameworks. Show-o [223] uses a single transformer with autoregressive language modeling and discrete diffusion-based image generation, while MARS [62] integrates frozen LLMs with trainable visual experts via SemVIE for fine-grained text-to-image generation. The ILLUME series [67, 193] combines semantic tokenization with diffusion decoding. Specifically, ILLUME [193] introduces a vision tokenizer to enable LLM-based understanding, generation, and self-enhancement, while ILLUME+ [67] extends it with a dual-branch tokenizer (DualViTok) and a diffusion decoder for high-fidelity image synthesis and editing. In addition, DDT-LLaMA [153] introduces discrete diffusion timestep tokens with a recursive structure to enhance visual representation in multimodal generation. In parallel, Token-Shuffle [128] designs a plug-and-play spatial token reordering strategy that enhances high-resolution autoregressive generation.

***(iv) Advanced Tokenizer Architectures and Integration.*** Morph-Tokens [152] and Libra [229] both decouple visual processing from MLLMs. Morph-Tokens [152] separates abstract prompts and visual tokens for task-specific comprehension and generation, while Libra [229] routes inputs through expert modules and cross-modal bridges for discrete autoregressive modeling. FashionM3 [155] finetunes the Show-O [223] model on discrete visual tokens derived from MAGVIT-v2 [242] to support fashion-specific multimodal recommendation and image generation. HimTok [201] equips an LLM with hierarchical discrete mask tokens based on TiTok tokenizer [243], enabling coarse-to-fine segmentation without relying on external decoders. Both SemHiTok [28] and Unicode$^2$ [26] adopt hierarchical codebook designs to improve visual tokenization. SemHiTok [28] employs semantic guidance to structure the hierarchy for better language alignment, while Unicode$^2$ [26] constructs a cascaded 500K-entry codebook to enhance stability.

**(b) Text + Audio.** In text-audio applications, discrete tokenization enables LLMs to jointly model speech and language for speech recognition, synthesis, and dialogue.

***(i) Discrete Speech Tokenization for Understanding and Generation.*** For instance, DiscreteSLU [174] explores applications of spoken language understanding in LLMs by discrete speech units and a speech adapter. MSRT [129] introduces a mixed-scale re-tokenization layer, enabling better alignment of multi-granularity speech information with language model inputs for speech recognition. The CosyVoice series [37, 38] improves TTS scalability and expressivity by incorporating multilingual supervision and streaming generation. CosyVoice [37] leverages supervised speech tokens for multilingual zero-shot synthesis, while CosyVoice 2 [38] incorporates streaming techniques for emotional and expressive control. T5-TTS [143] exploits attention priors and CTC-based alignment loss with a T5 [163] architecture and spectral codec [91] tokenizer for monotonic alignment between modalities, improving the robustness of TTS. Similarly, GPT-Talker [113] introduces semantic and style tokens derived from multimodal dialogue contexts for expressive speech. For attribute controllability of zero-shot TTS, Spark-TTS [207] introduces attribute labels and fine-grained attributes and generates tokens by the CoT.

***(ii) Real-Time and Dialog-Oriented Speech Modeling.*** The SpeechGPT series [251, 252] supports real-time dialogue through multi-stage training and semantic-perceptual disentanglement. SpeechGPT [251] adopts a three-stage strategy for cross-modal transfer, while SpeechGPT-Gen [252] introduces chain-of-information generation for efficient and expressive speech synthesis. And for low latency and computational overhead, In [261], IntrinsicVoice innovatively reduces the lengths of speech token sequences, and thereby lessens the differences between modalities. To support full-duplex dialogue [32, 257], Moshi [32] generates semantic and acoustic tokens in a streaming and hierarchical manner, while OmniFlatten [257] chunks and flattens speech and text tokens into a single sequence, followed by multi-stage post-training for half- and full-duplex abilities.

***(iii) Unified Speech-Language Foundation Models.*** AudioPaLM [169] and VoxtLM [130] extend LLMs with discrete audio tokens and unified vocabularies to support multitask speech-language modeling, including ASR, TTS, and speech-text continuation. Several models, such as LauraGPT [36] and Kimi-Audio [33], integrate discrete audio tokens with continuous representations for audio understanding, generation, recognition, and conversation.

**(c) Text + Video.** In text-video applications, discrete tokenization bridges language and visual dynamics, enabling LLMs to generate or understand videos through unified or hierarchical token sequences. Loong [209] unifies text and video tokens into a single autoregressive sequence and introduces progressive short-to-long training with reweighted loss and token re-encoding mechanisms, generating coherent minute-level videos. To support efficient multimodal understanding and generation, Video-LaVIT [81] presents a unified video-language pre-training framework that decouples visual and motion information through discrete tokenization. Building on hierarchical modeling, HiTVideo [274] encodes videos into multi-layer discrete tokens to balance compression and reconstruction, and enabling efficient text-to-video generation.

**(d) Text + Graph.** Recent methods have extended discrete tokenization to specialized domains through domain-specific adaptations. For molecular modeling, UniMoT [256] introduces a unified molecule-text language model that leverages vector quantization to discretize molecular representations, enabling joint sequence modeling and cross-modal generation in a shared token space. In addition, HIGHT [25] presents hierarchical graph tokenization with node-, motif-, and graph-level tokens to capture multi-scale structural semantics for graph-language alignment. For electronic health record tasks, MedTok [175] proposes a discrete tokenization framework for medical codes by integrating textual descriptions with graph-based relational contexts, supporting multimodal representation learning. To seamlessly integrate with LLMs for knowledge-aware reasoning, SSQR [103] develops a self-supervised quantization approach to encode knowledge graphs into discrete tokens.

**(e) Text + Motion.** Text-motion applications leverage discrete tokenization to map linguistic instructions to structured motion representations, supporting generation and control across diverse embodiments and tasks. MotionGlot [58] introduces a unified Transformer decoder that generates discrete motion tokens for diverse embodiments (e.g., humans, quadrupeds) using embodiment-specific VQ-VAEs and instruction-tuned text prompts. In [273], AvatarGPT uses VQ-VAE-based motion tokenization and integrates motion tokens into an LLM to unify understanding, planning, and generation tasks through instruction tuning. SemGrasp [97] decomposes grasp generation into a three-level (i.e., orientation, manner, and refinement) token prediction task, using hierarchical VQ-VAE-based discretization aligned with language and point cloud inputs. Recently, Walk-the-Talk [166] employs VQ-VAE to discretize pedestrian motion and leverages LLMs to generate diverse and realistic behaviors from descriptions of natural languages.

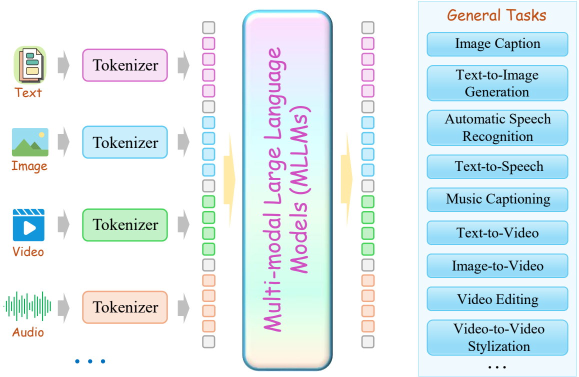

Fig. 13. MLLMs process multi-modal inputs by tokenizing text, image, video, and audio into a unified space for diverse tasks [87, 213].

**(f) Text + Image + Audio.** Some approaches have explored the mixed-modeling abilities of frozen language models by modality-specific discrete tokenizers, demonstrating the effectiveness of discrete representations in unified multimodal processing. For instance, TEAL [237] enables frozen LLMs to process multi-modal data by leveraging VQ-GAN and Whisper-based tokenizers. AnyGPT [250] extends this idea by employing SpeechTokenizer [226], Encodec [31] and SEED [47] tokenizers for speech, music, and vision, respectively, achieving unified discrete sequence modeling. Furthermore, DMLM [186] innovatively normalizes the sequence lengths across modalities and designs mixed supervised and unsupervised training for speech-centric tasks.

**(g) Text + Image + Video.** Most visual models still rely on diffusion-based approaches and adopt separate modules for understanding and generation, resulting in suboptimal alignment between perception and generation [205, 220]. Emu3 [205] attempts to eliminate this need for diffusion or compositional architectures by processing all modalities uniformly under the next-token prediction paradigm in a discrete space. Based on the same paradigm, VILA-U [220] also aligns visual tokens with textual inputs by contrastive learning. And LWM [109] extends the scalability of such models to long video modeling by introducing Blockwise RingAttention, supporting sequences exceeding one million tokens with efficient memory and compute optimization.

**(h) Text + Audio + Motion.** Cross-modal tasks can be formulated as sequence-to-sequence translation by discrete tokenization, enabling flexible any-to-any generation and allowing models to leverage the strengths of autoregressive sequence modeling. Building on this idea, LLM Gesticulator [154] generates rhythmically aligned and editable full-body co-speech gestures from audio signals with the aid of residual vector quantization, demonstrating scalability and controllability in gesture synthesis.

**(i) Text + Image + Audio + Video.** Building upon the foundation of discrete tokenization across modalities, some models extend support to audio and video, forming fully multimodal generative systems. As illustrated in Fig. 13, such models unify diverse modality streams via shared token spaces to support both cross-modal understanding and generation. Kondratyuk et al. [87] propose VideoPoet, introducing additional audio modality for video generation, encoding visual and audio signals into the discrete space by MAGVIT-v2 [242] and SoundStream [246] tokenizers. To support multimodal interleaved sequence generation on discrete tokens, MIO [213] introduces alignment pre-training, interleaved pre-training and speech-enhanced pre-training followed by supervised fine-tuning for multimodal foundation models.

**(j) Text + Image + Audio + Action.** Beyond incorporating diverse modalities into a shared space, achieving stable training across multiple modalities, especially stable training from scratch, has also attracted increasing attention. Unified-IO 2 [121] applies 2D rotary embeddings, QK normalization and scaled cosine attention mechanisms for stability, scaling Unified-IO [120] to audio and action modalities under a multimodal mixture of denoisers objective.

## 6 Challenges and Future Directions

Despite recent progress, discrete tokenization still faces challenges that hinder its effectiveness and generalization. In this section, we discuss key issues and promising future directions.

**(a) Codebook Utilization.** Under-utilized codebooks result in inefficient representations, limiting the expressiveness of discrete tokens. Although techniques like reparameterization tricks [69, 278] and diversity regularization [191, 242, 255] help improve token usage, they often compromise stability. Future research could focus on approaches that balance token diversity and coverage with stability, ensuring better codebook utilization without sacrificing performance. For instance, it is worth exploring curriculum-based code activation schedules to promote balanced code usage, and hybrid codebook designs that integrate multiple structural priors (e.g., semantic or spherical organization) to enhance flexibility while maintaining robustness.

**(b) Information Loss.** Discrete quantization inevitably causes information loss [20, 36, 44, 98], especially when multiple distinct continuous embeddings are mapped to the same code. In such cases, semantically different entities become indistinguishable, degrading the quality of the downstream representations. This issue is especially prominent in low-codebook scenarios or when codebooks collapse [216, 277]. Although this limitation is inherent to discretization, future research can explore task- and modality-aware strategies to mitigate its impact. For instance, image generation may tolerate loss in low-saliency regions, whereas classification or retrieval tasks require higher code precision. Adaptive coding schemes that allocate capacity based on downstream objectives offer a promising direction.

**(c) Gradient Propagation.** Discrete quantization breaks the differentiability of neural networks, making it difficult to propagate gradients through discrete latent variables. To enable end-to-end training, common approximations such as the Straight-Through Estimator (STE) [10, 148] and Gumbel-Softmax [6, 72] are widely adopted. However, these methods can introduce estimation bias, gradient variance, and convergence instability, especially in complex downstream tasks. An alternative is to design principled, stable gradient approximations for discrete token spaces. Promising approaches include score-based estimators, hybrid relaxations, and RL-inspired methods aligned with token selection. Task-specific gradient flows and regularization may further improve robustness and generalization.

**(d) Granularity and Semantic Alignment.** Balancing token granularity is crucial—coarse tokens may miss details, while overly fine-grained ones inflate sequence length and cost [65, 93, 207]. Existing methods also struggle to align with semantic boundaries, especially in continuous modalities such as image or audio, where meaningful units are often ambiguous or task-specific [28, 66, 162]. To address these issues, promising directions include adaptive and hierarchical quantization that modulates granularity based on content complexity and semantics. Techniques like dynamic masking, multi-scale encoding, and attention-guided segmentation may better align tokens with structure, leading to more efficient and interpretable representations.

**(e) Unification of Discrete and Continuous Tokens.** Discrete and continuous representations each have distinct advantages: discrete tokens offer compactness, modularity, and interpretability, while continuous embeddings preserve fine-grained information and facilitate gradient-based optimization [100]. However, most existing works often separate these two types of representations, limiting their synergy. Only a few recent studies have begun to explore their integration [82, 211, 270]. Developing hybrid architectures that unify discrete and continuous tokens during training and inference represents a promising direction. This includes using continuous features to inform discrete selection, or structuring continuous generation with discrete priors. Joint optimization and representation alignment may further enhance interoperability between the two spaces.

Beyond the five main challenges, two supplementary directions are included in Appendix to provide additional insights for future research.

## 7 Conclusion

This survey presents an overview of discrete tokenization techniques for integrating multimodal data with LLMs. We introduce a unified taxonomy of VQ methods, explore their adaptation across modalities, and highlight integration challenges. By synthesizing classical and modern insights, we identify key limitations and propose future research directions. This work aims to advance efficient and interpretable multimodal learning in foundation models and provide practical guidance for multimodal data integration into LLMs.

## References

[1] 01.AI. Yi-1.5 Series: Open LLMs by 01.AI. https://huggingface.co/01-ai, 2024. 23

[2] E. Agustsson, F. Mentzer, and et al. Soft-to-hard vector quantization for end-to-end learning compressible representations. *NeurIPS*, 2017. 2, 8

[3] L. Ai, J. Q. Yu, and et al. Optimized residual vector quantization for efficient approximate nearest neighbor search. *Multimedia Systems*, 2017. 2, 5

[4] Alibaba Cloud. Qwen: Open Large Language Models by Alibaba Cloud. https://huggingface.co/Qwen, 2024. 11, 12, 22, 23

[5] A. Babenko and V. Lempitsky. Additive quantization for extreme vector compression. In *CVPR*, 2014. 2, 7

[6] A. Baevski, S. Schneider, and et al. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv*, 2019. 2, 3, 4, 9, 15

[7] A. Baevski, Y. Zhou, and et al. wav2vec 2.0: A framework for self-supervised learning of speech representations. *NeurIPS*, 2020. 2, 9

[8] H. Bao, L. Dong, and et al. Beit: Bert pre-training of image transformers. *arXiv*, 2021. 2, 8, 9

[9] C. F. Barnes, S. A. Rizvi, and et al. Advances in residual vector quantization: A review. *IEEE Transactions on Image Processing*, 1996. 1, 5

[10] Y. Bengio, N. Léonard, and et al. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv*, 2013. 3, 15

[11] X. Bi, D. Chen, and et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv*, 2024. 13

[12] Y. Boget, M. Gregorova, and et al. Discrete graph auto-encoder. *TMLR*, 2024. 2, 9

[13] Y. Boget, F. Lavda, and et al. Glad: Improving latent graph generative modeling with simple quantization. In *AAAI*, 2025. 2, 9

[14] S. Cao, Y. Yin, and et al. Efficient-vqgan: Towards high-resolution image generation with efficient vision transformers. In *ICCV*, 2023. 2, 9

[15] Y. Cao, M. Long, and et al. Deep visual-semantic quantization for efficient image retrieval. In *CVPR*, 2017. 2, 9

[16] E. Casanova, R. Langman, and et al. Low frame-rate speech codec: a codec designed for fast high-quality speech llm training and inference. In *ICASSP*, 2025. 2, 9, 21

[17] H. Chang, H. Zhang, and et al. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 2, 9

[18] H. Chang, H. Zhang, and et al. Muse: Text-to-image generation via masked generative transformers. In *ICML*, 2023. 2, 10, 22

[19] L. Chen, Z. Wang, and et al. Next token prediction towards multimodal intelligence: A comprehensive survey. *arXiv*, 2024. 1, 8

[20] L. Chen, S. Tan, and et al. A spark of vision-language intelligence: 2-dimensional autoregressive transformer for efficient finegrained image generation. In *ICLR*, 2025. 2, 9, 15

[21] M. Chen, W. Zhang, and et al. Entity-agnostic representation learning for parameter-efficient knowledge graph embedding. In *AAAI*, 2023. 2, 8

[22] T. Chen, L. Li, and et al. Differentiable product quantization for end-to-end embedding compression. In *ICML*, 2020. 2, 6, 11

[23] X. Chen, Z. Wu, and et al. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv*, 2025. 2, 13, 23

[24] Y. Chen, T. Guan, and et al. Approximate nearest neighbor search by residual vector quantization. *Sensors*, 2010. 2, 5

[25] Y. Chen, Q. Yao, and et al. Improving graph-language alignment with hierarchical graph tokenization. In *ICML Workshop*, 2024. 2, 14, 23

[26] Y. Chen, H. Zhong, and et al. Unicode$^2$: Cascaded large-scale codebooks for unified multimodal understanding and generation. *arXiv*, 2025. 2, 13, 23

[27] Z. Chen, J. Wu, and et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024. 13

[28] Z. Chen, C. Wang, and et al. Semhitok: A unified image tokenizer via semantic-guided hierarchical codebook for multimodal understanding and generation. *arXiv*, 2025. 2, 13, 15, 23

[29] A. Chowdhery, S. Narang, and et al. Palm: Scaling language modeling with pathways. *JMLR*, 2023. 13

[30] DeepSeek-AI. DeepSeek Models on HuggingFace. https://huggingface.co/DeepSeek-AI, 2024. 1, 13, 23

[31] A. Défossez, J. Copet, and et al. High fidelity neural audio compression. *TMLR*, 2023. 2, 9, 14

[32] A. Défossez, L. Mazaré, and et al. Moshi: a speech-text foundation model for real-time dialogue. *arXiv*, 2024. 2, 13, 23

[33] D. Ding, Z. Ju, and et al. Kimi-audio technical report. *arXiv*, 2025. 1, 2, 14, 23

[34] M. Ding, Z. Yang, and et al. Cogview: Mastering text-to-image generation via transformers. *NeurIPS*, 2021. 2, 10

[35] T. Du, Y. Wang, and et al. On the role of discrete tokenization in visual representation learning. *arXiv*, 2024. 2, 8, 9

[36] Z. Du, J. Wang, and et al. Lauragpt: Listen, attend, understand, and regenerate audio with gpt. *arXiv*, 2023. 2, 14, 15, 23

[37] Z. Du, Q. Chen, and et al. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv*, 2024. 2, 13, 23

[38] Z. Du, Y. Wang, and et al. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv*, 2024. 2, 13, 23

[39] P. Esser, R. Rombach, and et al. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2, 3, 4, 10, 21

[40] C. Fifty, R. G. Junkins, and et al. Restructuring vector quantization with the rotation trick. In *ICLR*, 2025. 4

[41] O. Gafni, A. Polyak, and et al. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 2022. 2, 10

[42] M. Galkin, E. Denis, and et al. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. *arXiv*, 2021. 2, 8, 9

[43] L. Gao, X. Zhu, and et al. Beyond product quantization: Deep progressive quantization for image retrieval. *arXiv*, 2019. 2, 9

[44] T. Gautam, R. Pryzant, and et al. Soft convex quantization: revisiting vector quantization with convex optimization. In *L4DC*, 2024. 2, 8, 15

[45] S. Ge, T. Hayes, and et al. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *ECCV*, 2022. 2, 10

[46] T. Ge, K. He, and et al. Optimized product quantization for approximate nearest neighbor search. In *CVPR*, 2013. 2, 6

[47] Y. Ge, Y. Ge, and et al. Planting a seed of vision in large language model. *arXiv*, 2023. 2, 13, 14, 23

[48] Y. Ge, S. Zhao, and et al. Making llama see and draw with seed tokenizer. In *ICLR*, 2024. 2, 13, 23

[49] Google DeepMind. PaLM 2: Pathways Language Model. https://ai.google/discover/palm2/, 2023. 11, 22, 23

[50] Google Research. T5 and Variants (Hugging Face). https://huggingface.co/google/, 2022. 1, 11, 12, 22, 23

[51] N. Goswami, Y. Mukuta, and et al. Hypervq: Mlr-based vector quantization in hyperbolic space. *arXiv*, 2024. 2, 4

[52] A. Grattafiori, A. Dubey, and et al. The llama 3 herd of models. *arXiv*, 2024. 1, 11, 12

[53] S. Gu, D. Chen, and et al. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2022. 2, 10

[54] Y. Gu, X. Wang, and et al. Rethinking the objectives of vector-quantized tokenizers for image synthesis. In *CVPR*, 2024. 2, 9, 21

[55] D. Guo, C. Li, and et al. Parametric and nonparametric residual vector quantization optimizations for ann search. *Neurocomputing*, 2016. 2, 5

[56] Y. Guo, Z. Li, and et al. Recent advances in discrete speech tokens: A review. *arXiv*, 2025. 2

[57] L. Guotao, B. Zhang, and et al. Lg-vq: Language-guided codebook learning. *NeurIPS*, 2024. 2, 10

[58] S. Harithas, S. Sridhar, and et al. Motionglot: A multi-embodied motion generation model. *arXiv*, 2024. 2, 14, 23

[59] M. Hassid, T. Remez, and et al. Textually pretrained speech language models. *NeurIPS*, 2023. 1, 2, 11, 22

[60] T. He, L. Gao, and et al. Sneq: Semi-supervised attributed network embedding with attention-based quantisation. In *AAAI*, 2020. 2, 9

[61] T. He, L. Gao, and et al. Semisupervised network embedding with differentiable deep quantization. *TNNLS*, 2021. 2, 9

[62] W. He, S. Fu, and et al. Mars: Mixture of auto-regressive models for fine-grained text-to-image synthesis. In *AAAI*, 2025. 1, 2, 13, 23

[63] M. Hong, Y. Xia, and et al. Eager-llm: Enhancing large language models as recommenders through exogenous behavior-semantic integration. In *WWW*, 2025. 2, 12, 22

[64] Y. Hou, Z. He, and et al. Learning vector-quantized item representation for transferable sequential recommenders. In *WWW*, 2023. 2, 11

[65] M. Huang, Z. Mao, and et al. Towards accurate image coding: Improved autoregressive image generation with dynamic vector quantization. In *CVPR*, 2023. 2, 9, 15

[66] M. Huang, Z. Mao, and et al. Not all image regions matter: Masked vector quantization for autoregressive image generation. In *CVPR*, 2023. 2, 9, 15

[67] R. Huang, C. Wang, and et al. Illume+: Illuminating unified mllm with dual visual tokenization and diffusion refinement. *arXiv*, 2025. 2, 13, 23

[68] T. J. Huang, J. Q. Yang, and et al. Improving llms for recommendation with out-of-vocabulary tokens. *arXiv*, 2024. 2, 12, 22

[69] M. Huh, B. Cheung, and et al. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *ICML*, 2023. 2, 8, 15

[70] I. A. M. Huijben, M. Douze, and et al. Residual quantization with implicit neural codebooks. *arXiv*, 2024. 2, 5

[71] M. Imran, H. Yin, and et al. Refrs: Resource-efficient federated recommender system for dynamic and diversified user preferences. *TOIS*, 2023. 2, 11

[72] E. Jang, S. Gu, and et al. Categorical reparameterization with gumbel-softmax. *arXiv:1611.01144*, 2016. 3, 15

[73] Y. K. Jang and N. I. Cho. Self-supervised product quantization for deep unsupervised image retrieval. In *ICCV*, 2021. 2, 8, 9

[74] H. Jegou, M. Douze, and et al. Product quantization for nearest neighbor search. *TPAMI*, 2010. 2, 5, 21

[75] T. Jenrungrot, M. Chinen, and et al. Lmcodec: A low bitrate speech codec with causal transformer models. In *ICASSP*, 2023. 2, 9

[76] Y. Ji, C. Liu, and et al. Nt-llm: A novel node tokenizer for integrating graph structure into large language models. *arXiv*, 2024. 2, 11, 22

[77] J. Jia, J. Gao, and et al. From principles to applications: A comprehensive survey of discrete tokenizers in generation, comprehension, recommendation, and information retrieval. *arXiv*, 2025. 1

[78] X. H. Jiang, Y. Ai, and et al. A streamable neural audio codec with residual scalar-vector quantization for real-time communication. *SPL*, 2025. 2, 9

[79] Y. Jiang, Q. Chen, and et al. Unicodec: Unified audio codec with single domain-adaptive codebook. *arXiv*, 2025. 2, 9, 21

[80] Y. Jiao, H. Qiu, and et al. Unitoken: Harmonizing multimodal understanding and generation through unified visual encoding. *arXiv*, 2025. 2, 13, 23

[81] Y. Jin, Z. Sun, and et al. Video-lavit: Unified video-language pretraining with decoupled visual-motional tokenization. In *ICML*, 2024. 1, 2, 14, 23

[82] Y. Jin, K. Xu, and et al. Unified language-vision pretraining in llm with dynamic discrete visual tokenization. In *ICLR*, 2024. 2, 13, 15, 23

[83] Z. Ju, Y. Wang, and et al. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *ICML*, 2024. 2, 10

[84] Y. Kalantidis and Y. Avrithis. Locally optimized product quantization for approximate nearest neighbor search. In *CVPR*, 2014. 2, 6

[85] W. C. Kang, D. Z. Cheng, and et al. Learning multi-granular quantized embeddings for large-vocab categorical features in recommender systems. In *WWW*, 2020. 2, 11

[86] B. Klein and L. Wolf. End-to-end supervised product quantization for image search and retrieval. In *CVPR*, 2019. 2, 6

[87] D. Kondratyuk, L. Yu, and et al. Videopoet: A large language model for zero-shot video generation. In *ICML*, 2024. 1, 2, 14, 22, 23

[88] F. Kreuk, G. Synnaeve, and et al. Audiogen: Textually guided audio generation. In *ICLR*, 2023. 2, 10

[89] R. Kumar, P. Seetharaman, and et al. High-fidelity audio compression with improved rvqgan. *NeurIPS*, 2023. 2, 9

[90] Z. Lahrichi, G. Hadjeres, and et al. Qincodec: Neural audio compression with implicit neural codebooks. *arXiv*, 2025. 2, 9, 21

[91] R. Langman, A. Jukić, and et al. Spectral codecs: Spectrogram-based audio codecs for high quality speech synthesis. *arXiv*, 2024. 2, 10, 13

[92] C. F. Lee, C. C. Chang, and et al. A survey of data hiding based on vector quantization. In *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, 2020. 1

[93] D. Lee, C. Kim, and et al. Autoregressive image generation using residual quantization. In *CVPR*, 2022. 2, 9, 11, 15

[94] G. Li, X. Zhang, and et al. Semantic convergence: Harmonizing recommender systems via two-stage alignment and behavioral semantic tokenization. In *AAAI*, 2025. 2, 12, 22

[95] H. Li, L. Xue, and et al. Single-codec: Single-codebook speech codec towards high-performance speech generation. In *Interspeech*, 2024. 2, 10

[96] J. Li, Q. Wang, and et al. Random entity quantization for parameter-efficient compositional knowledge graph representation. In *EMNLP*, 2023. 2, 8

[97] K. Li, J. Wang, and et al. Semgrasp: Semantic grasp generation via language aligned discretization. In *ECCV*, 2024. 2, 14, 23

[98] S. Li, L. Zhang, and et al. Mergevq: A unified framework for visual generation and representation with disentangled token merging and quantization. *arXiv*, 2025. 2, 9, 15, 21

[99] T. Li, H. Chang, and et al. Mage: Masked generative encoder to unify representation learning and image synthesis. In *CVPR*, 2023. 2, 9, 21

[100] Z. Li, J. Zhang, and et al. Continuous or discrete, that is the question: A survey on large multi-modal models from the perspective of input-output space extension. *Preprints*, 2024. 1, 15

[101] H. Lin, T. Wang, and et al. Toklip: Marry visual tokens to clip for multimodal comprehension and generation. *arXiv*, 2025. 2, 10

[102] Q. Lin, Z. Peng, and et al. A survey of quantized graph representation learning: Connecting graph structures with large language models. *arXiv*, 2025. 1

[103] Q. Lin, T. Zhao, and et al. Self-supervised quantized representation for seamlessly integrating knowledge graphs with large language models. *arXiv*, 2025. 2, 14, 23

[104] C. Liu, D. Lian, and et al. Online optimized product quantization. In *ICDM*, 2020. 2, 6

[105] D. Liu, S. Zhao, and et al. Lumina-mgpt: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining. *arXiv*, 2024. 2, 13, 23

[106] E. Liu, B. Zheng, and et al. End-to-end learnable item tokenization for generative recommendation. *arXiv*, 2024. 2, 12, 22

[107] H. Liu, W. Yan, and et al. Language quantized autoencoders: Towards unsupervised text-image alignment. *NeurIPS*, 2023. 2, 11, 22

[108] H. Liu, X. Xu, and et al. Semanticodec: An ultra low bitrate semantic audio codec for general sound. *J-STSP*, 2024. 2, 9

[109] H. Liu, W. Yan, and et al. World model on million-length video and language with blockwise ringattention. *arXiv*, 2024. 2, 14, 22, 23

[110] Q. Liu, J. Zhang, and et al. Online additive quantization. In *KDD*, 2021. 2, 7

[111] Q. Liu, X. Dong, and et al. Vector quantization for recommender systems: a review and outlook. *arXiv*, 2024. 1

[112] Q. Liu, J. Zhu, and et al. Store: Streamlining semantic tokenization and generative recommendation with a single llm. *arXiv*, 2024. 2, 12, 22

[113] R. Liu, Y. Hu, and et al. Generative expressive conversational speech synthesis. In *MM*, 2024. 2, 13, 23

[114] S. Liu, H. Lu, and et al. Improved residual vector quantization for high-dimensional approximate nearest neighbor search. *arXiv*, 2015. 2, 5

[115] S. Liu, J. Shao, and et al. Generalized residual vector quantization and aggregating tree for large scale search. *IEEE Transactions on Multimedia*, 2017. 2, 5

[116] T. Liu, Z. Ma, and et al. Vqtalker: Towards multilingual talking avatars through facial motion tokenization. In *AAAI*, 2025. 2, 11

[117] Y. Liu, Q. Cao, and et al. Towards variable and coordinated holistic co-speech motion generation. In *CVPR*, 2024. 2, 10

[118] Z. Liu, L. Wu, and et al. Multi-view empowered structural graph wordification for language models. In *AAAI*, 2025. 2, 3, 11, 22

[119] LMSYS. Vicuna and Variants (Hugging Face). https://huggingface.co/lmsys, 2023. 13, 23

[120] J. Lu, C. Clark, and et al. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv*, 2022. 2, 10, 15

[121] J. Lu, C. Clark, and et al. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *CVPR*, 2024. 2, 15, 22, 23

[122] Z. Lu, D. Lian, and et al. Differentiable optimized product quantization and beyond. In *WWW*, 2023. 2, 6

[123] J. Luo, P. Dong, and et al. Action-quantized offline reinforcement learning for robotic skill learning. In *CoRL*, 2023. 2, 10

[124] X. Luo, J. Cao, and et al. Qarm: Quantitative alignment multimodal recommendation at kuaishou. *arXiv*, 2024. 2, 12, 22

[125] Y. Luo, H. Li, and et al. Node identifiers: Compact, discrete representations for efficient graph learning. *arXiv*, 2024. 2, 9, 22

[126] C. Ma, Y. Jiang, and et al. Unitok: A unified tokenizer for visual generation and understanding. *arXiv*, 2025. 2, 10

[127] T. Ma, L. Xie, and et al. Clawmachine: Fetching visual tokens as an entity for referring and grounding. *arXiv*, 2024. 2, 13, 23

[128] X. Ma, P. Sun, and et al. Token-shuffle: Towards high-resolution image generation with autoregressive models. *arXiv*, 2025. 2, 13, 23

[129] Y. Ma, C. Zhang, and et al. Tuning large language model for speech recognition with mixed-scale re-tokenization. *IEEE Signal*

*Processing Letters*, 2024. 2, 13, 23

[130] S. Maiti, Y. Peng, and et al. Voxtlm: Unified decoder-only models for consolidating speech recognition, synthesis and speech, text continuation tasks. In *ICASSP*, 2024. 2, 14, 23

[131] J. Martinez, H. H. Hoos, and et al. Stacked quantizers for compositional vector compression. *arXiv*, 2014. 2, 5

[132] J. Martinez, J. Clement, and et al. Revisiting additive quantization. In *ECCV*, 2016. 2, 7

[133] J. Martinez, S. Zakhmi, and et al. Lsq++: Lower running time and higher recall in multi-codebook quantization. In *ECCV*, 2018. 2, 7

[134] Á. Martín-Cortinas, D. Sáez-Trigueros, and et al. Enhancing the stability of llm-based speech generation systems through self-supervised representations. *arXiv*, 2024. 2, 11, 22

[135] Y. Matsui, Y. Uchida, and et al. A survey of product quantization. *ITE Transactions on Media Technology and Applications*, 2018. 2, 5

[136] F. Mentzer, D. Minnen, and et al. Finite scalar quantization: Vq-vae made simple. In *ICLR*, 2024. 2, 7, 21

[137] Meta AI. OPT and Variants (Hugging Face). https://huggingface.co/facebook, 2022. 22, 23

[138] Meta AI. Chameleon: Mixed-Modal Early-Fusion Foundation Models. https://huggingface.co/facebook, 2024. 23

[139] Meta AI. LLaMA: Open Foundation Models by Meta AI. https://ai.meta.com/llama/, 2024. 11, 12, 22, 23

[140] Microsoft Research. Phi and Variants (Hugging Face). https://huggingface.co/microsoft, 2023. 23

[141] Mistral AI. Mistral models on hugging face. https://huggingface.co/mistralai, 2023. 1, 13, 23

[142] N. M. Nasrabadi, R. A. King, and et al. Image coding using vector quantization: A review. *IEEE Transactions on Communications*, 1988. 1

[143] P. Neekhara, S. Hussain, and et al. Improving robustness of llm-based speech synthesis by learning monotonic alignment. In *Interspeech*, 2024. 2, 13, 23

[144] M. Ni, X. Li, and et al. Nuwa-lip: language-guided image inpainting with defect-free vqgan. In *CVPR*, 2023. 2, 10, 22

[145] Y. Nishimura, T. Hirose, and et al. Hall-e: hierarchical neural codec language model for minute-long zero-shot text-to-speech synthesis. *arXiv*, 2024. 2, 10

[146] L. Niu, Z. Xu, and et al. Residual vector product quantization for approximate nearest neighbor search. *ESWA*, 2023. 2, 6

[147] M. Norouzi and D. J. Fleet. Cartesian k-means. In *CVPR*, 2013. 2, 6

[148] A. V. D. Oord, O. Vinyals, and et al. Neural discrete representation learning. *NeurIPS*, 2017. 2, 3, 4, 10, 11, 15, 21

[149] OpenAI. GPT Series: Language Models by OpenAI. https://openai.com/gpt, 2024. 1, 11, 22, 23

[150] OpenGVLab. OpenGVLab/InternVL on hugging face. https://huggingface.co/OpenGVLab, 2024. 1, 13, 23

[151] E. C. Ozan, S. Kiranyaz, and et al. Competitive quantization for approximate nearest neighbor search. *TKDE*, 2016. 2, 5

[152] K. Pan, S. Tang, and et al. Auto-encoding morph-tokens for multimodal llm. In *ICML*, 2024. 2, 13, 23

[153] K. Pan, W. Lin, and et al. Generative multimodal pretraining with discrete diffusion timestep tokens. *arXiv*, 2025. 2, 13, 23

[154] H. Pang, T. Ding, and et al. Llm gesticulator: leveraging large language models for scalable and controllable co-speech gesture synthesis. In *ICCGV*, 2025. 2, 14, 22, 23

[155] K. Pang, X. Zou, and et al. Fashionm3: Multimodal, multitask, and multiround fashion assistant based on unified vision-language model. *arXiv*, 2025. 2, 13, 23

[156] J. Paparrizos, I. Edian, and et al. Fast adaptive similarity search through variance-aware quantization. In *ICDE*, 2022. 2, 6

[157] J. D. Parker, A. Smirnov, and et al. Scaling transformers for low-bitrate high-quality speech coding. *arXiv*, 2024. 2, 8, 9, 21

[158] Z. Peng, L. Dong, and et al. Beit v2: masked image modeling with vector-quantized visual tokenizers. *arXiv*, 2022. 2, 9

[159] Z. Qiu, J. Liu, and et al. Hihpq: Hierarchical hyperbolic product quantization for unsupervised image retrieval. In *AAAI*, 2024. 2, 6

[160] H. Qu, Y. Cai, and et al. Llms are good action recognizers. In *CVPR*, 2024. 2, 12, 22

[161] H. Qu, W. Fan, and et al. Tokenrec: learning to tokenize id for llm-based generative recommendation. *arXiv*, 2024. 2, 12, 22

[162] L. Qu, H. Zhang, and et al. Tokenflow: Unified image tokenizer for multimodal understanding and generation. *arXiv*, 2024. 2, 13, 15, 23

[163] C. Raffel, N. Shazeer, and et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020. 11, 12, 13

[164] S. Rajput, N. Mehta, and et al. Recommender systems with generative retrieval. *NeurIPS*, 2023. 2, 11

[165] A. Ramesh, M. Pavlov, and et al. Zero-shot text-to-image generation. In *ICML*, 2021. 2, 4, 10

[166] M. Ramesh, F. B. Flohr, and et al. Walk-the-talk: Llm driven pedestrian motion generation. In *IV*, 2024. 2, 14, 23

[167] A. Razavi, A. V. D. Oord, and et al. Generating diverse high-fidelity images with vq-vae-2. *NeurIPS*, 2019. 2, 3, 4

[168] A. Roy, A. Vaswani, and et al. Theory and experiments on vector quantized autoencoders. *arXiv*, 2018. 4, 5

[169] P. K. Rubenstein, C. Asawaroengchai, and et al. Audiopalm: A large language model that can speak and listen. *arXiv*, 2023. 2, 14, 23

[170] M. Sachan. Knowledge graph embedding compression. In *ACL*, 2020. 2, 9

[171] S. Sadok, S. Leglaive, and et al. A vector quantized masked autoencoder for audiovisual speech emotion recognition. *CVIU*, 2025. 2, 10

[172] K. Sargent, K. Hsu, and et al. Flow to the mode: Mode-seeking diffusion autoencoders for state-of-the-art image tokenization. *arXiv*, 2025. 2, 9

[173] F. Shi, Z. Luo, and et al. Taming scalable visual tokenizer for autoregressive image generation. *arXiv*, 2024. 2, 8

[174] S. Shon, K. Kim, and et al. Discreteslu: A large language model with self-supervised discrete speech units for spoken language understanding. In *Interspeech*, 2024. 2, 13, 23

[175] X. Su, S. Messica, and et al. Multimodal medical code tokenizer. *arXiv*, 2025. 2, 14, 23

[176] P. Sun, Y. Jiang, and et al. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv*, 2024. 2, 11, 22

[177] Y. Takida, T. Shibuya, and et al. Sq-vae: Variational bayes on discrete representation with self-annealed stochastic quantization. *arXiv*, 2022. 2, 5, 8

[178] Y. Takida, Y. Ikemiya, and et al. Hq-vae: Hierarchical discrete representation learning with variational bayes. *TMLR*, 2023. 2, 5, 8

[179] Z. Tan, B. Xue, and et al. Sweettokenizer: Semantic-aware spatial-temporal tokenizer for compact visual discretization. *arXiv*, 2024. 2, 10, 21

[180] A. Tang, T. He, and et al. Vidtok: A versatile and open-source video tokenizer. *arXiv*, 2024. 2, 10, 21

[181] H. Tang, Y. Wu, and et al. Hart: Efficient visual generation with hybrid autoregressive transformer. In *ICLR*, 2025. 2, 8, 10

[182] Z. Tang, C. Wu, and et al. Strokenuwa—tokenizing strokes for vector graphic synthesis. In *ICML*, 2024. 2, 11, 22

[183] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv*, 2024. 2, 13, 23

[184] K. Tian, Y. Jiang, and et al. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *NeurIPS*, 2024. 2, 9, 21

[185] H. Touvron, L. Martin, and et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2023. 11, 12

[186] V. A. Trinh, R. Southwell, and et al. Discrete multimodal transformers with a pretrained large language model for mixed-supervision speech processing. *arXiv*, 2024. 2, 14, 22, 23

[187] T. Vallaeys, M. Muckley, and et al. Qinco2: Vector compression and search with improved implicit neural codebooks. *arXiv*, 2025. 2, 5, 21

[188] A. Vasuki and P. T. Vanathi. A review of vector quantization techniques. *IEEE Potentials*, 2006. 1

[189] R. Villegas, M. Babaeizadeh, and et al. Phenaki: Variable length video generation from open domain textual descriptions. In *ICLR*, 2023. 2, 10

[190] I. Volkov. Homology-constrained vector quantization entropy regularizer. *arXiv*, 2022. 2, 8

[191] T.-L. Vuong, T. Le, and et al. Vector quantized wasserstein auto-encoder. In *ICML*, 2023. 2, 5, 8, 15

[192] C. Wang, S. Chen, and et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv*, 2023. 2, 10

[193] C. Wang, G. Lu, and et al. Illume: Illuminating your llms to see, draw, and self-enhance. *arXiv*, 2024. 2, 13, 23

[194] H. Wang, Y. Wang, and et al. A lightweight knowledge graph embedding framework for efficient inference and storage. In *CIKM*, 2021. 2, 9

[195] H. Wang, S. Suri, and et al. Larp: Tokenizing videos with a learned autoregressive generative prior. *arXiv*, 2024. 2, 4, 10, 21

[196] J. Wang, J. Wang, and et al. Optimized cartesian k-means. *TKDE*, 2014. 2, 6

[197] J. Wang, Z. Zeng, and et al. Contrastive quantization with code memory for unsupervised image retrieval. In *AAAI*, 2022. 2, 8, 9

[198] J. Wang, Y. Jiang, and et al. Omnitokenizer: A joint image-video tokenizer for visual generation. *NeurIPS*, 2024. 2, 10

[199] L. Wang, Y. Zhao, and et al. Image understanding makes for a good tokenizer for image generation. *NeurIPS*, 2024. 2, 9, 21

[200] L. Wang, K. Hassani, and et al. Learning graph quantized tokenizers. In *ICLR*, 2025. 2, 4, 9, 21

[201] T. Wang, C. Cheng, and et al. Himtok: Learning hierarchical mask tokens for image segmentation with large multimodal model. *arXiv*, 2025. 2, 13, 23

[202] W. Wang, H. Bao, and et al. Learnable item tokenization for generative recommendation. In *CIKM*, 2024. 2, 12, 22

[203] W. Wang, F. Zhang, and et al. End-to-end vision tokenizer tuning. *arXiv*, 2025. 2, 13, 23

[204] X. Wang, T. Zhang, and et al. Supervised quantization for similarity search. In *CVPR*, 2016. 2, 6

[205] X. Wang, X. Zhang, and et al. Emu3: Next-token prediction is all you need. *arXiv*, 2024. 2, 14, 22, 23

[206] X. Wang, Z. Zhu, and et al. Worlddreamer: Towards general world models for video generation via predicting masked tokens. *arXiv*, 2024. 2, 11

[207] X. Wang, M. Jiang, and et al. Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. *arXiv*, 2025. 2, 13, 15, 23

[208] Y. Wang, Z. Ren, and et al. Content-based collaborative generation for recommender systems. In *CIKM*, 2024. 2, 12, 22

[209] Y. Wang, T. Xiong, and et al. Loong: Generating minute-level long videos with autoregressive language models. *arXiv*, 2024. 2, 14, 23

[210] Y. Wang, J. Xun, and et al. Eager: Two-stream generative recommender with behavior-semantic collaboration. In *KDD*, 2024. 2, 11

[211] Y. Wang, Z. Lin, and et al. Tokenbridge: Bridging continuous and discrete tokens for autoregressive visual generation. *arXiv*, 2025. 15

[212] Z. Wang, Z. Zhang, and et al. Gft: Graph foundation model with transferable tree vocabulary. *NeurIPS*, 2024. 2, 9, 21

[213] Z. Wang, K. Zhu, and et al. Mio: A foundation model on multimodal tokens. *arXiv*, 2024. 2, 14, 15, 23

[214] M. Weber, L. Yu, and et al. Maskbit: Embedding-free image generation via bit tokens. *TMLR*, 2024. 2, 8, 9

[215] B. Wei, T. Guan, and et al. Projected residual vector quantization for ann search. *IEEE Multimedia*, 2014. 2, 5

[216] W. Williams, S. Ringer, and et al. Hierarchical quantized autoencoders. *NeurIPS*, 2020. 2, 5, 8, 15

[217] C. Wu, X. Chen, and et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv*, 2024. 2, 13, 23

[218] H. Wu and M. Flierl. Vector quantization-based regularization for autoencoders. In *AAAI*, 2020. 2, 8

[219] L. Wu, Y. Tian, and et al. Mape-ppi: Towards effective and efficient protein-protein interaction prediction via microenvironment-aware protein embedding. In *ICLR*, 2024. 2, 9

[220] Y. Wu, Z. Zhang, and et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv*, 2024. 2, 14, 22, 23

[221] Z. B. Wu and J. Q. Yu. Vector quantization: a review. *Frontiers of Information Technology & Electronic Engineering*, 2019. 1, 2, 4

[222] J. Xia, C. Zhao, and et al. Mole-bert: Rethinking pre-training graph neural networks for molecules. In *ICLR*, 2023. 2, 9

[223] J. Xie, W. Mao, and et al. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv*, 2024. 2, 13, 23

[224] R. Xie, C. Du, and et al. Muse-vl: Modeling unified vlm through semantic discrete encoding. *arXiv*, 2024. 2, 13, 23

[225] D. Xin, X. Tan, and et al. Rall-e: Robust codec language modeling with chain-of-thought prompting for text-to-speech synthesis. *arXiv*, 2024. 2, 10

[226] Z. Xin, Z. Dong, and et al. Speechtokenizer: Unified speech tokenizer for speech language models. In *ICLR*, 2024. 2, 3, 9, 14

[227] D. Xu, I. W. Tsang, and et al. Online product quantization. *TKDE*, 2018. 2, 6

[228] Y. Xu, X. Feng, and et al. Vq-nerv: A vector quantized neural representation for videos. *arXiv*, 2024. 2, 10, 21

[229] Y. Xu, X. Yang, and et al. Libra: Building decoupled vision system on large language models. In *ICML*, 2024. 2, 13, 23

[230] Y. Xu, S.-X. Zhang, and et al. Comparing discrete and continuous space llms for speech recognition. *arXiv*, 2024. 2, 11, 22

[231] W. Yan, Y. Zhang, and et al. Videogpt: Video generation using vq-vae and transformers. *arXiv*, 2021. 2, 9

[232] A. Yang, A. Li, and et al. Qwen3 technical report. *arXiv*, 2025. 1, 11, 12

[233] D. Yang, S. Liu, and et al. Hifi-codec: Group-residual vector quantization for high fidelity audio codec. *arXiv*, 2023. 2, 9

[234] D. Yang, R. Huang, and et al. Simplespeech 2: Towards simple and efficient text-to-speech with flow-based scalar latent transformer diffusion models. *arXiv*, 2024. 2, 10

[235] D. Yang, D. Wang, and et al. Simplespeech: Towards simple and efficient text-to-speech with scalar latent transformer diffusion models. In *Interspeech*, 2024. 2, 10

[236] L. Yang, Y. Tian, and et al. Vqgraph: Rethinking graph representation space for bridging gnns and mlps. In *ICLR*, 2024. 2, 9, 21

[237] Z. Yang, Y. Zhang, and et al. Teal: Tokenize and embed all for multi-modal large language models. *arXiv*, 2023. 2, 14, 22, 23

[238] J. Yin, Z. Zeng, and et al. Unleash llms potential for sequential recommendation by coordinating dual dynamic index mechanism. In *WWW*, 2025. 2, 11, 12, 22

[239] J. Yu, X. Li, and et al. Vector-quantized image modeling with improved vqgan. In *ICLR*, 2022. 2, 3, 4, 9

[240] L. Yu, Y. Cheng, and et al. Magvit: Masked generative video transformer. In *CVPR*, 2023. 2, 10

[241] L. Yu, Y. Cheng, and et al. Spae: Semantic pyramid autoencoder for multimodal generation with frozen llms. *NeurIPS*, 2023. 2, 11, 22

[242] L. Yu, J. Lezama, and et al. Language model beats diffusion-tokenizer is key to visual generation. In *ICLR*, 2024. 2, 3, 7, 8, 10, 13, 15, 21

[243] Q. Yu, M. Weber, and et al. An image is worth 32 tokens for reconstruction and generation. *NeurIPS*, 2024. 2, 8, 9, 13

[244] J. Yuan and X. Liu. Product tree quantization for approximate nearest neighbor search. In *ICIP*, 2015. 2, 6

[245] J. Yuan and X. Liu. Transformed residual quantization for approximate nearest neighbor search. *arXiv*, 2015. 2, 5

[246] N. Zeghidour, A. Luebs, and et al. Soundstream: An end-to-end neural audio codec. *TASLP*, 2021. 2, 9, 15

[247] L. Zeng, J. Yu, and et al. Hierarchical vector quantized graph autoencoder with annealing-based code selection. In *WWW*, 2025. 2, 8, 9, 21

[248] K. Zha, L. Yu, and et al. Language-guided image tokenization for generation. *arXiv*, 2024. 2, 10

[249] L. Zhai, H. Ding, and et al. One quantizer is enough: Toward a lightweight audio codec. *arXiv*, 2025. 2, 9

[250] J. Zhan, J. Dai, and et al. Anygpt: Unified multimodal llm with discrete sequence modeling. In *ACL*, 2024. 2, 14, 22, 23

[251] D. Zhang, S. Li, and et al. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. In *EMNLP*, 2023. 2, 13, 23

[252] D. Zhang, X. Zhang, and et al. Speechgpt-gen: Scaling chain-of-information speech generation. *arXiv*, 2024. 2, 13, 23

[253] G. Zhang, T. Zhang, and et al. V2flow: Unifying visual tokenization and large language model vocabularies for autoregressive image generation. *arXiv*, 2025. 2, 11, 22

[254] H. Zhang, J. Li, and et al. Gt-svq: A linear-time graph transformer for node classification using spiking vector quantization. *arXiv*, 2025. 2, 9, 21

[255] J. Zhang, F. Zhan, and et al. Regularized vector quantization for tokenized image synthesis. In *CVPR*, 2023. 2, 4, 8, 15

[256] J. Zhang, Y. Bian, and et al. Unimot: Unified molecule-text language model with discrete token representation. *arXiv*, 2024. 2, 14, 23

[257] Q. Zhang, L. Cheng, and et al. Omniflatten: An end-to-end gpt model for seamless voice conversation. *arXiv*, 2024. 2, 13, 14, 23

[258] S. Zhang, S. Roller, and et al. Opt: Open pre-trained transformer language models. *arXiv*, 2022. 23

[259] T. Zhang, C. Du, and et al. Composite quantization for approximate nearest neighbor search. In *ICML*, 2014. 2, 6

[260] T. Zhang, G. J. Qi, and et al. Sparse composite quantization. In

*CVPR*, 2015. 2, 6

[261] X. Zhang, X. Lyu, and et al. Intrinsicvoice: Empowering llms with intrinsic real-time voice interaction abilities. *arXiv*, 2024. 2, 13, 23

[262] Y. Zhang, Z. Chen, and et al. Mygo: Discrete modality information as fine-grained tokens for multi-modal knowledge graph completion. *arXiv*, 2024. 2, 10

[263] Z. Zhang, L. Zhou, and et al. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *arXiv*, 2023. 2, 10

[264] Y. Zhao, Y. Xiong, and et al. Image and video tokenization with binary spherical quantization. *arXiv*, 2024. 2, 7, 8, 10, 13

[265] Y. Zhao, F. Xue, and et al. Qlip: Text-aligned visual tokenization unifies auto-regressive multimodal understanding and generation. *arXiv*, 2025. 2, 13, 23

[266] B. Zheng, Y. Hou, and et al. Adapting large language models by integrating collaborative semantics for recommendation. In *ICDE*, 2024. 2, 12, 22

[267] B. Zheng, H. Lu, and et al. Universal item tokenization for transferable generative recommendation. *arXiv*, 2025. 2, 12, 22

[268] C. Zheng and A. Vedaldi. Online clustered codebook. In *ICCV*, 2023. 2, 5, 8

[269] C. Zheng, T. L. Vuong, and et al. Movq: Modulating quantized vectors for high-fidelity image generation. *NeurIPS*, 2022. 2, 9, 21

[270] P. Zheng, J. Wang, and et al. Rethinking discrete tokens: Treating them as conditions for continuous autoregressive image synthesis. *arXiv*, 2025. 15

[271] R. Zheng, C. Cheng, and et al. Prise: Llm-style sequence compression for learning temporal action abstractions in control. In *ICML*, 2024. 2, 10

[272] L. Zhou, C. Ruan, and et al. Tvc: Tokenized video compression with ultra-low bitrate. *arXiv*, 2025. 2, 10

[273] Z. Zhou, Y. Wan, and et al. Avatargpt: All-in-one framework for motion understanding planning generation and beyond. In *CVPR*, 2024. 2, 14, 23

[274] Z. Zhou, Y. Yang, and et al. Hitvideo: Hierarchical tokenizers for enhancing text-to-video generation with autoregressive large language models. *arXiv*, 2025. 2, 14, 23

[275] J. Zhu, M. Jin, and et al. Cost: Contrastive quantization based semantic tokenization for generative recommendation. In *RecSys*, 2024. 2, 8, 11

[276] L. Zhu, F. Wei, and et al. Beyond text: Frozen large language models in visual signal comprehension. In *CVPR*, 2024. 1, 2, 11, 22

[277] L. Zhu, F. Wei, and et al. Scaling the codebook size of vqgan to 100,000 with a utilization rate of 99%. *arXiv*, 2024. 2, 8, 15

[278] Y. Zhu, B. Li, and et al. Addressing representation collapse in vector quantized models with one linear layer. *arXiv*, 2024. 2, 4, 8, 15

[279] X. Zhuang, Q. Zhang, and et al. Learning invariant molecular representation in latent discrete space. *NeurIPS*, 2023. 2, 9

# Appendix

## A. Supplement for Classic Applications without LLMs

This section serves as a complementary resource to Section 3. Due to space constraints in the main text, several representative papers on classic applications without large language models are included in this appendix to maintain the completeness of the survey and ensure a comprehensive coverage of the topic.

### A.1 Image

As an extension to Section 3.1, this section summarizes several representative works on image tokenization. MoVQ [269] integrates conditional normalization and multichannel quantization into VQGAN [39] for spatially variant image information. To unify image generation and representation learning, MAGE [99] uses variable masking ratios, VQ-KD [199] distills knowledge from pretrained image understanding encoders (e.g., CLIP), and VAR [184] proposes a GPT-style AR model using next-scale prediction with multiscale VA-VAE quantization. Additionally, SeQ-GAN [54] balances semantic compression and detail with perceptual loss and fine-tuning of the decoder. Recently, MergeVQ [98] introduces token merging in VQ-based generative models, leading to semantically richer tokenizer and boosting image generation quality.

### A.2 Audio

To complement the main discussion in Section 3.2, this appendix provides a more detailed overview of several representative methods in audio tokenization, highlighting diverse strategies in codec design, quantization mechanisms, and model architecture choices. UniCodec [79] presents a partitioned domain-adaptive codebook and MoE strategy for unified audio codec with single-codebook. Similarly, QinCodec [90] leverages offline quantization with QINCo2 [187], enabling the use of any off-the-shelf quantizer without optimization constraints. Meanwhile, TAAE [157] and LFSC [16] adopt FSQ [136] for low-bitrate and low frame-rate speech codec, respectively.

### A.3 Graph

We include here several additional graph tokenization methods that further illustrate the diversity of discrete modeling strategies beyond those discussed in Section 3.3. VQ-Graph [236] introduces a structure-aware tokenizer that encodes local substructures into discrete codes for effective GNN-to-MLP distillation. Along similar lines, GFT [212] treats computation trees as a discrete tree vocabulary via tree reconstruction, unifying tasks into tree classification for graph foundation model. In contrast, HQA-GAE [247] applies VQ-VAE [148] to graphs with a hierarchical codebook and annealing-based selection to address underutilization and sparsity. Complementarily, GQT [200] leverages multitask self-supervised learning and RVQ to generate hierarchical graph tokens for efficient, generalizable tokenization. Aiming for efficiency, GT-SVQ [254] builds a linear-time graph transformer using spiking vector quantization, where spike count embeddings act as codewords to guide attention.

### A.4 Video

This section provides additional representative works that extend video tokenization techniques beyond those discussed in Section 3.4. LARP [195] introduces a holistic video tokenizer by stochastic quantization, enhancing generation performance with AR prior model. Focusing on improved quantization strategies, VidTok [180] is an open-source video tokenizer that uses FSQ [136] to improve discrete representation and mitigate codebook collapse in VQ methods. To better capture hierarchical video structure, VQ-NeRV [228] adopts a U-shaped architecture with VQ-based blocks to discretize shallow and inter-frame residual features. Complementarily, SweetTok [179] introduces a semantic-aware tokenizer that captures spatial-temporal cues to produce compact, informative video tokens.

## B. Supplement for LLMs with One Modality Applications

Table 2 serves as a complementary resource to Section 4, presenting a structured summary of LLM-based applications that leverage discrete tokenization on a single modality. Each entry is categorized by the quantized modality (e.g., image, audio, graph, action, or complex modality in recommendation), the employed quantization technique (e.g., VQ [148], PQ [74], LFQ [242]), the backbone LLM, and the availability of open-source implementations. This table provides concrete instances of the methods discussed in the main text, offering a practical reference for how vector quantization techniques have been integrated into single-modality LLM pipelines.

## C. Supplement for LLMs based Multiple Modalities Applications

Table 3 serves as a complementary resource to Section 5, presenting a structured summary of LLM-based applications that leverage discrete tokenization across multiple modalities. Each entry is categorized by the quantized modality (e.g., image, audio, video, motion), the employed quantization technique (e.g., VQ [148], k-means, LFQ [242]), the backbone LLM, and the availability of open-source implementations. This table provides concrete instances of the methods discussed in the main text, offering a practical reference for how vector quantization techniques have been integrated into multimodal LLM pipelines.

## D. Supplement for Challenges and Future Directions

This section serves as a complementary resource to Section 6. It highlights two important challenges that merit further discussion and attention. Due to space limitations in the main text, we provide a more detailed analysis here to ensure a more thorough and nuanced treatment of these issues.

**(f) Modality and Task Transferability.** Numerous tokenizers are handcrafted or domain-specific, limiting their applicability across tasks. A promising direction is to develop

TABLE 2
LLM-based Applications with Discrete Tokenization on Single Modality.

| Model | Quantized Modality | VQ Technique | LLM | Code |
|---|---|---|---|---|
| SPAE [241] | Image | VQ | PaLM 2 [49], GPT 3.5 [149] | - |
| LQAE [107] | Image | VQ | GPT-3 [149], InstructGPT [149] | https://github.com/haoliuhl/language-quantized-autoencoders |
| V2T Tokenizer [276] | Image | VQ | LLaMA 2-7B [139], LLaMA 2-13B [139], LLaMA 2-70B [139] | https://github.com/zh460045050/V2L-Tokenizer |
| LlamaGen [176] | Image | VQ | LLaMA [139] architecture | https://github.com/FoundationVision/LlamaGen |
| V$^2$Flow [253] | Image | VQ | LLaMA 2-7B [139] | https://github.com/zhangguiwei610/V2Flow |
| StrokeNUWA [182] | Image | RVQ | Flan-T5 3B [50] | - |
| TWIST [59] | Audio | k-means | OPT-125M,350M,1.3B [137], LLaMA-7B,13B [139] | https://pages.cs.huji.ac.il/adiyoss-lab/twist/ |
| SSVC [134] | Audio | RVQ | from scratch | - |
| JTFS LM [230] | Audio | k-means | LLaMA 2-7B [139] | https://github.com/xuyaoxun/ASRCompare |
| NT-LLM [76] | Graph | GART | LLaMA 3-8B [139] | - |
| Dr.E [118] | Graph | RVQ | LLaMA 2-7B [139] | - |
| LLM-AR [160] | Action | VQ | LLaMA-13B [139] | - |
| LC-Rec [266] | Complex modality in RecSys | RVQ | LLaMA-7B [139] | https://github.com/RUCAIBox/LC-Rec/ |
| LETTER [202] | Complex modality in RecSys | RVQ | LLaMA-7B [139] | https://github.com/HonghuiBao2000/LETTER |
| ColaRec [208] | Complex modality in RecSys | k-means | T5-small [50] | https://github.com/Junewang0614/ColaRec |
| STORE [112] | Complex modality in RecSys | k-means | OPT-base [137] | - |
| QARM [124] | Complex modality in RecSys | VQ, RVQ | Not reported | - |
| META ID [68] | Complex modality in RecSys | k-means | T5-small [50], LLaMA-7B [139] | - |
| TokenRec [161] | Complex modality in RecSys | VQ | T5-small [50] | - |
| ETEGRec [106] | Complex modality in RecSys | RVQ | T5 [139] | - |
| Semantic Convergence [94] | Complex modality in RecSys | RVQ | LLaMA-7B [139] | - |
| ED$^2$ [238] | Complex modality in RecSys | VQ | LLaMA 2 [139] | https://github.com/Esperanto-mega/ED2 |
| EAGER-LLM [63] | Complex modality in RecSys | k-means | LLaMA-7B [139] | https://github.com/Indolent-Kawhi/EAGER-LLM |
| UTGRec [267] | Complex modality in RecSys | RVQ | Qwen-VL-2B [4], T5 [50] | - |

generalizable tokenization methods that work across domains. This could be achieved through cross-modal pretraining or unified discrete spaces [87, 121, 250], enabling tokenization to function seamlessly across multiple data types. While several works have demonstrated success in aligning two modalities such as text and image, efforts to support three [109, 154, 186, 205, 220, 237, 250] or more [87, 121] modalities in a unified discrete representation remain rare, presenting a compelling direction for future research on multimodal tokenization.

**(g) Interpretability and Controllability.** Learned tokens often lack transparency, making them difficult to interpret and control. This is a significant challenge for applications requiring human-understandable representations. Prior work using discrete latent spaces has shown that while tokens can capture meaningful patterns, they often do not correspond to interpretable or manipulable concepts [18, 125, 144]. Future directions include aligning discrete tokens with human-centric semantics to enhance transparency and usability. This may involve concept-grounded codebooks, token-level editing, or interpretable priors during training. Improving interpretability can also help with debugging, personalization, and safety in downstream tasks.

TABLE 3
LLM-based Applications with Discrete Tokenization on Multi-Modality.

| Model | Quantized Modality | VQ Technique | LLM | Open Source |
|---|---|---|---|---|
| SEED [47] | Image | VQ | OPT-2.7B [137, 258] | https://github.com/AILab-CVC/SEED |
| Chameleon [183] | Image | VQ | 7B from scratch | https://github.com/facebookresearch/chameleon |
| Lumina-mGPT [105] | Image | VQ | Chameleon-7B, 30B [138, 183] | https://github.com/Alpha-VLLM/Lumina-mGPT |
| ILLUME [193] | Image | VQ | Vicuna-7B [119] | - |
| Janus [217] | Image | VQ | DeepSeek-LLM (1.3B) [30] | https://github.com/deepseek-ai/janus |
| Janus-Pro [23] | Image | VQ | DeepSeek-LLM (1.5B, 7B) [30] | https://github.com/deepseek-ai/janus |
| MUSE-VL [224] | Image | VQ | Qwen2.5-7B [4], Qwen2.5-32B [4], Yi-1.5-9B [1], Yi-1.5-34B [1] | - |
| Morph-Tokens [152] | Image | VQ | Vicuna [119] | https://github.com/DCDmllm/MorphTokens |
| LaVIT [82] | Image | VQ | LLaMA-7B [139] | https://github.com/jy0205/LaVIT |
| SEED-LLaMA [48] | Image | VQ | Vicuna-7B [119], LLaMA 2-13B-Chat [139] | https://github.com/AILab-CVC/SEED |
| Libra [229] | Image | LFQ | LLaMA 2-7B-Chat [139] | https://github.com/YifanXu74/Libra |
| Show-o [223] | Image | LFQ | Phi1.5-1.3B [140] | https://github.com/showlab/Show-o |
| TokenFlow [162] | Image | VQ | Vicuna-v1.5-13B [119], Qwen2.5-14B [4], LLaMA 2-7B [139] | https://byteflow-ai.github.io/TokenFlow/ |
| ClawMachine [127] | Image | VQ | LaVIT-7B [82] | https://github.com/martian422/ClawMachine |
| DDT-LLaMA [153] | Image | VQ | LLaMA 3-8B [139] | https://ddt-llama.github.io/ |
| FashionM3 [155] | Image | LFQ | fine-tuning Show-o [223] | - |
| HiMTok [201] | Image | VQ | InternVL-2.5-8B [150] | https://github.com/yayafengzi/LMM-HiMTok |
| ILLUME+ [67] | Image | VQ | Qwen2.5-3B [4] | https://illume-unified-mllm.github.io/ |
| QLIP [265] | Image | BSQ | LLaMA 3 [139] | - |
| SemHiTok [28] | Image | VQ | Vicuna-v1.5-7B [119], Qwen2.5-3B [4] | - |
| UniToken [80] | Image | VQ | Chameleon 7B [138, 183] | https://github.com/SxJyJay/UniToken |
| Token-Shuffle [128] | Image | VQ | LLaMA-2.7B [139] | - |
| MARS [62] | Image | VQ | Qwen-7B [4] | https://github.com/fusiming3/MARS |
| ETT [203] | Image | VQ | Qwen2.5-1.5B [4] | - |
| Unicode$^2$ [26] | Image | k-means | Qwen2.5-7B-Instruct [4] | - |
| AudioPaLM [169] | Audio | k-means | PaLM-2 8B [49] | https://google-research.github.io/seanet/audiopalm/examples/ |
| LauraGPT [36] | Audio | RVQ | Qwen-1.8B [4] | https://lauragpt.github.io/ |
| SpeechGPT [251] | Audio | k-means | LLaMA-13B [139] | https://github.com/0nutation/SpeechGPT<br>https://0nutation.github.io/SpeechGPT.github.io/ |
| SpeechGPT-Gen [252] | Audio | RVQ | LLaMA 2-7B-Chat [139] | https://github.com/0nutation/SpeechGPT |
| CosyVoice [37] | Audio | VQ | from scratch | https://github.com/FunAudioLLM/CosyVoice<br>https://fun-audio-llm.github.io/ |
| CosyVoice 2 [38] | Audio | FSQ | Qwen2.5-0.5B [4] | https://funaudiollm.github.io/cosyvoice2 |
| IntrinsicVoice [261] | Audio | k-means | Qwen2-7B-Instruct [4] | https://instrinsicvoice.github.io/ |
| Moshi [32] | Audio | RVQ | 7B from scratch | https://github.com/kyutai-labs/moshi |
| OmniFlatten [257] | Audio | VQ | Qwen2-0.5B [4] | https://omniflatten.github.io/ |
| MSRT [129] | Audio | k-means | FLAN-T5 [50], GPT2-Medium [149], LLaMA 2-7B [139] | - |
| T5-TTS [143] | Audio | FSQ | T5 [50] | - |
| DiscreteSLU [174] | Audio | k-means | Mistral-7B [141] | - |
| GPT-Talker [113] | Audio | k-means, VQ | from scratch | https://github.com/AI-S2-Lab/GPT-Talker |
| VoxtLM [130] | Audio | k-means | OPT [137] | https://soumimaiti.github.io/icassp24_voxtlm/ |
| Spark-TTS [207] | Audio | VQ, FSQ | Qwen2.5-0.5B-Instruct [4] | https://github.com/SparkAudio/Spark-TTS |
| Kimi-Audio [33] | Audio | VQ | Qwen2.5-7B [4] | https://github.com/MoonshotAI/Kimi-Audio |
| Loong [209] | Video | VQ | 700M, 3B, 7B from scratch | https://epiphqny.github.io/Loong-video |
| Video-LaVIT [81] | Video | VQ | LLaMA 2-7B [139] | https://video-lavit.github.io/ |
| HiTVideo [274] | Video | LFQ | LLaMA-3B [139] | https://ziqinzhou66.github.io/project/HiTVideo/ |
| UniMoT [256] | Graph | VQ | LLaMA 2-7B [139] | https://uni-mot.github.io/ |
| HIGHT [25] | Graph | VQ | Vicuna-v1.3–7B [119] | https://higraphllm.github.io/ |
| MedTok [175] | Graph | VQ | LLaMA 3.1–8B [139] | - |
| SSQR [103] | Graph | VQ | LLaMA 2-7B [139], LLaMA 3.1-8B [139] | - |
| MotionGlot [58] | Motion | VQ | GPT-2 small [149] | https://ivl.cs.brown.edu/research/motionglot.html |
| AvatarGPT [273] | Motion | VQ | GPT-2 Large [149], T5-Large [50] | https://zixiangzhou916.github.io/AvatarGPT/ |
| Semgrasp [97] | Motion | VQ | Vicuna-7B [119] | https://kailinli.github.io/SemGrasp/ |
| Walk-the-Talk [166] | Motion | VQ | Flan-T5-Base [50] | https://iv.ee.hm.edu/publications/w-the-t/ |
| TEAL [237] | Image<br>Audio | VQ<br>k-means | LLaMA [139] | - |
| DMLM [186] | Image<br>Audio | VQ<br>RVQ | OPT [137, 258] | - |
| AnyGPT [250] | Image<br>Audio | VQ<br>RVQ | LLaMA 2-7B [139] | https://junzhan2000.github.io/AnyGPT.github.io/ |
| Emu3 [205] | Image<br>Video | VQ<br>VQ | 8B from scratch | https://emu.baai.ac.cn/about |
| VILA-U [220] | Image<br>Video | RVQ<br>RVQ | LLaMA 2-7B [139] | https://github.com/mit-han-lab/vila-u |
| LWM [109] | Image<br>Video | VQ<br>VQ | LLaMA 2-7B [139] | https://largeworldmodel.github.io/lwm/ |
| LLM Gesticulator [154] | Audio<br>Motion | RVQ<br>RVQ | Qwen1.5-0.5B [4], Qwen1.5-1.8B [4], Qwen1.5-4B [4], Qwen1.5-7B [4] | - |
| VideoPoet [87] | Image<br>Video<br>Audio | LFQ<br>LFQ<br>RVQ | 1B, 8B from scratch | https://sites.research.google/videopoet/ |
| MIO [213] | Image<br>Video<br>Audio | VQ<br>VQ<br>RVQ | Yi-6B-Base [1] | https://github.com/MIO-Team/MIO |
| Unified-IO 2 [121] | Image<br>Audio | VQ<br>VQ | 1.1B, 3.2B, 6.8B from scratch | https://github.com/allenai/unified-io-2<br>https://unified-io-2.allenai.org/ |