Transform-Free Feature Coding via Entropy-Constrained Vector Quantization

Qiaoxi Chen[1], Changsheng Gao[2*], Li Li[1], Dong Liu[1]

[1]MOE Key Laboratory of Brain-Inspired Intelligent Perception and Cognition, University of Science and Technology of China, Hefei, Anhui 230026, China
[2]Nanyang Technological University, Singapore 639798
xxii@mail.ustc.edu.cn, changsheng.gao@ntu.edu.sg, {lil1, dongeliu}@ustc.edu.cn

Abstract

Feature coding has recently emerged as a key technique for efficient transmission of intermediate representations in distributed AI systems. Existing approaches largely follow a *transform-based* pipeline inherited from image and video coding, where the transform module is used to remove spatial structural redundancies in visual signals. However, our analysis indicates that such redundancies have already been largely removed during feature extraction, which reduces the necessity of the transform module. Building on this insight, we propose a new *transform-free* pipeline that directly encodes the extracted features via a vector quantization module and an entropy model. The proposed transform-free framework jointly learns the quantization codebook and entropy model, enabling end-to-end optimization tailored to the inherent feature characteristics. Furthermore, the proposed method inherently avoids the computational complexity of the transform module. Experiments on features from diverse architectures and tasks demonstrate that our method achieves superior rate-distortion performance compared to transform-based baselines, while significantly reducing the encoding and decoding complexity.

Code — https://github.com/xxii111/FCVQ

Introduction

The rapid advancement of large models has greatly expanded the scope of artificial intelligence, powering applications such as autonomous driving, remote healthcare, and smart surveillance (Zhang et al. 2024; Qu et al. 2025; Friha et al. 2024). However, the deployment of such models in real-world scenarios faces two critical challenges: computational cost and data. Centralized execution on the cloud raises privacy concerns due to raw data transmission, while on-device execution is often infeasible due to limited computational resources (Vepakomma et al. 2018). To address these issues, distributed deployment has emerged as a practical paradigm, where models are partitioned between cloud and edge devices (Lepikhin et al. 2020; Tian et al. 2022, 2024; Wang et al. 2024). In this setting, intermediate features become the primary medium of information exchange between the two sides. Efficiently compressing these features is therefore crucial for reducing communication overhead and enabling real-time inference, making feature coding a key technique for large model deployments.

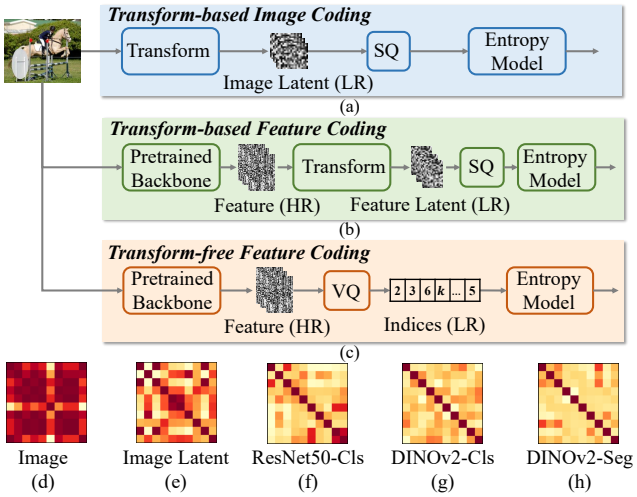


Figure 1: Comparison of three coding pipelines. The key difference lies in (c), where the transform stage is removed, reducing complexity while preserving task-relevant semantics. Subfigures (d)-(h) visualize the feature correlations of the input image, image latent, and features. The low correlation observed in (f)-(h) indicates that extracted features are already decorrelated, eliminating the need for an additional transform module.

Existing feature coding methods are largely inspired by image and video coding pipelines, typically adopting a *transform-quantization-entropy coding* framework (Chen et al. 2019; Choi, Bajić et al. 2021; Alvar and Bajić 2021; Liu et al. 2023b; Yamazaki et al. 2022; Suzuki et al. 2022; Cai, Xing, and Gao 2022; Kim et al. 2023; Yang et al. 2024; Gao et al. 2025a). The transform module is central to decorrelating raw visual signals, a necessity in image coding where spatial redundancies are high. However, deep features differ fundamentally from images: they have undergone multiple nonlinear operations in the backbone network, which already suppress structural redundancies and enrich

*Corresponding author.

semantic representations. Our systematic correlation analysis confirms this distinction, revealing that deep features exhibit significantly lower correlations than raw images or even image latents. This observation challenges the necessity of the transform step and motivates a fundamental question: Can we design a transform-free feature coding framework tailored to the properties of deep features? In addition, prior studies often overlook the computational overhead of the transform module, which can substantially affect system latency and energy consumption. These limitations highlight the need for an effective and more efficient feature coding paradigm.

In this work, we propose a transform-free feature coding framework based on entropy-constrained vector quantization (ECVQ) (Chou et al. 2002). Instead of performing an explicit transform, our method directly partitions intermediate features into low-dimensional chunks and maps them to a learned codebook. The quantized indices are then entropy-coded using a discrete probability model. Both the codebook and entropy model are jointly optimized under a rate–distortion objective, enabling end-to-end training that aligns compression efficiency with downstream task performance. This design simultaneously simplifies the coding pipeline and reduces computational cost, while leveraging the inherent decorrelation properties of deep features.

Distinct from previous studies that primarily focus on either CNNs or large models, our framework is validated on diverse architectures and tasks, including ResNet50 (He et al. 2016) and DINOv2 (Oquab et al. 2023) for both classification and semantic segmentation. This comprehensive evaluation demonstrates the generalization ability of our method across heterogeneous feature types. Experiments show that our approach achieves superior rate-distortion performance compared to transform-based baselines, particularly at low bitrates, while reducing computation complexity. Such efficiency makes the proposed method well-suited for real-time and privacy-preserving deployment in practical edge-cloud systems. In summary, our contributions are threefold:

- We identify the unique statistical properties of deep features compared to raw images, providing empirical evidence that challenges the necessity of the transform module in feature coding and introduce the transform-free feature coding framework.
- We design an entropy-constrained vector quantization scheme that jointly learns the codebook and entropy model for end-to-end rate-distortion optimization.
- Our framework achieves higher compression efficiency across CNN and Transformer features for both classification and segmentation, while maintaining significantly lower computational complexity.

Related work

In this section, we introduce the related work on feature coding and vector quantization.

Input	Input Size	Mean	Max	Min	Median
Image Spatial	3	0.76	1.00	0.00	0.88
Image Channel	10000	0.98	1.0	0.96	0.98
Image Latent Spatial	192	0.52	1.00	0.00	0.55
Image Latent Channel	320	0.07	0.67	0.00	0.05
ResNet50 Cls Spatial	2048	0.36	0.95	0.00	0.31
ResNet50 Cls Channel	49	0.25	0.97	0.00	0.22
DINOv2 Cls Hor	64	0.15	1.00	0.00	0.11
DINOv2 Cls Ver	64	0.21	0.93	0.00	0.19
DINOv2 Seg Hor	10	0.30	1.00	0.00	0.27
DINOv2 Seg Ver	10	0.39	1.00	0.00	0.37
DINOv2 Seg Hor	180	0.14	1.00	0.00	0.10
DINOv2 Seg Ver	180	0.15	0.99	0.00	0.12

Table 1: Correlation statistics on various input images and features. The features show much lower correlation compared to images.

Feature Coding

Feature coding is introduced to analyzing before compressing, which is one of the two tracks included within the Coding for Machines (Duan et al. 2020; Yang et al. 2024). Existing feature coding methods are typically derived from image coding methods, leveraging transforms to decorrelate feature representations (Chen et al. 2019; Choi, Bajić et al. 2021; Liu et al. 2023b; Gao et al. 2024a,b; Ma et al. 2024). For example, Chen et al. propose to quantize and repack CNN feature maps, which are considered as one frame for compression via video codecs (Chen et al. 2019). This method serves as the anchor in MPEG-FCM, where multi-layer features are first fused and then encoded (WG2 2023). Kim et al. introduce an end-to-end multi-scale feature coding framework that iteratively transforms and fuses multi-scale features into latent representations (Kim et al. 2023). Liu et al. propose a mutual feature coding strategy by compressing lower-scale features and predicting higher-scale ones (Liu et al. 2023b). In parallel, task-aware feature coding approaches aim for scalability by exploiting semantic relationships. Choi et al. proposed a latent-scalability method, using partial channels for detection and full channels for reconstruction (Choi, Bajić et al. 2021). Other scalable methods encode a base layer and residual streams to support multiple tasks (Wang et al. 2019; Liu et al. 2021; Yan et al. 2021; Chen et al. 2024). Recently, Gao et al. illustrate the importance of feature coding in large model deployments and introduce the large model feature coding methods (Gao et al. 2025d,c,b).

While effective, these transform-based methods inherit the complexity of image codecs and overlook the intrinsic properties of deep features, potentially limiting coding efficiency and flexibility.

Vector Quantization

Vector quantization (VQ) maps high-dimensional vectors to a discrete codebook by assigning each input vector to its nearest codeword (Gray 1984). It has been widely used in variou areas (Qi et al. 2025; Wu et al. 2025b,a; Jia

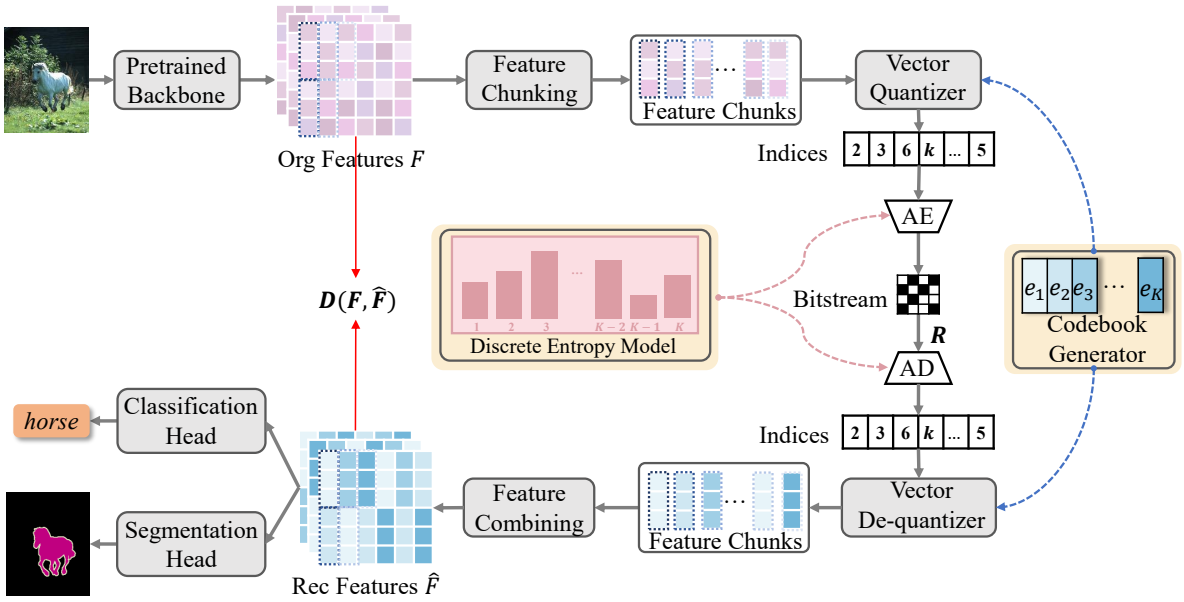


Figure 2: Overview of the proposed transform-free feature coding framework. The input image is processed by a pretrained backbone to generate the original feature F, which is uniformly partitioned into lower-dimensional chunks along a specified axis and subsequently fed into a vector quantizer to obtain discrete indices. These indices are entropy-coded into a bitstream using an arithmetic encoder. The decoding process mirrors the encoding pathway to reconstruct the feature. In this framework, optimization is applied solely to the discrete entropy model and the codebook generator under the rate-distortion objective, where bitrate R is estimated via the discrete entropy model and distortion D is measured by the MSE between the original and reconstructed features.

et al. 2024). In the coding area, the encoder transmits the index of the closest codeword, and the decoder reconstructs the vector via codebook lookup. However, VQ does not optimize rate-distortion performance, as codeword assignment ignores the bit-rate constraint. This shortcoming is addressed by Entropy-Constrained Vector Quantization, which introduces a Lagrangian formulation to optimize rate and distortion jointly (Chou et al. 2002). ECVQ modifies the codeword assignment procedure by incorporating a rate-distortion loss, enabling variable-rate encoding that is locally optimal. This framework aligns the quantization process more closely with information-theoretic principles, where both distortion and coding cost are explicitly balanced.

Recent advances in learning-based image compression have reintegrated VQ within the framework of rate-distortion optimization (Agustsson et al. 2017; Zhu et al. 2022; Feng et al. 2023; Qi et al. 2025; Jia et al. 2024). While existing methods focus on applying VQ in the transform stage or latent domains, its potential in the visual feature domain remains largely unexplored.

Analysis and Motivation

Revisiting Transform in Image and Feature Coding

Image coding has long adopted the *transform-quantization-entropy coding* pipeline, shown as Figure 1 (a) (Ballé et al. 2016, 2018; Cheng et al. 2020; Bross et al. 2021; He et al. 2022; Liu et al. 2023a). **The transform module plays a central role by removing spatial and channel redundancies in natural images**, thereby producing compact latents with reduced correlations. Such decorrelation simplifies subsequent entropy modeling, since entropy coders perform optimally when their inputs are nearly independent and identically distributed.

Inspired by image coding, most existing feature coding approaches, shown as Figure 1 (b), inherit the same design: features extracted by a pretrained backbone are passed through a transform before quantization and entropy coding. However, this design overlooks fundamental differences between images and features:

- Images are raw signals captured by cameras and naturally exhibit strong spatial correlations.
- Features, by contrast, are intermediate representations extracted by deep neural networks that have already undergone multiple nonlinear transformations aimed at redundancy suppression and semantic enrichment.

In other words, the pretrained backbone itself can be viewed as a transform, potentially reducing correlations inherently. This raises a key question: **Has the correlation already been sufficiently removed during feature extraction?**

Correlation Analysis

To answer this question, we conducted a systematic correlation analysis across different representations, including raw images, image latents, and features from multiple models

(ResNet50, DINOv2) and tasks (classification, segmentation). We evaluated correlations along different axes. Specifically, we compute the absolute value of the Pearson correlation coefficient along the spatial and channel axes for images, image latents, and ResNet-50 features (all represented in the $H \times W \times C$ format). For DINOv2 features, which are structured as $H \times W$, the correlation is evaluated along the horizontal and vertical axes. Figure 1 and Table 1 summarize the results. The key observations are:

- Images and image latents exhibit high correlations (e.g., 0.98 in Table 1), which justifies the transform stage in image coding.
- Deep features show dramatically reduced correlations. For example, ResNet50 features have mean spatial correlation of only 0.36.
- Feature correlations are even lower than image latents, indicating that neural networks inherently perform decorrelation as part of representation learning.

These findings suggest that the main benefit of the transform module (redundancy removal) is largely unnecessary for features. Thus, applying an additional transform to already decorrelated features may be redundant. This motivates us to propose the transform-free feature coding framework.

Transform-Free Feature Coding

Although removing the transform seems straightforward, one must recognize another implicit function of the transform in image coding: dimensionality reduction. By reducing the resolution, the transform generates lower-dimensional latents that are easier to quantize and encode.

To retain this benefit in a transform-free design, we introduce vector quantization as a core component. The proposed transform-free feature coding pipeline is shown in Figure 1 (c). The framework bypasses the transform stage entirely and directly applies vector quantization to deep features extracted from a pretrained backbone. The quantized indices are then modeled using an entropy model to produce compact bitstreams. This design leverages the inherent decorrelation of deep features while simplifying the coding pipeline and reducing computational overhead.

Proposed Method

In this section, we introduce the overall framework of the proposed transform-free feature coding method and its main components.

Overall Framework

We first present the overall framework, illustrated in Figure 2. Let $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$ denote the intermediate features extracted from a pretrained backbone, where C, H, and W represent the channel, height, and width dimensions, respectively. The objective is to encode $\mathbf{F}$ into a compact bitstream while preserving task-relevant semantics for downstream tasks such as classification and segmentation.

The framework first divides $\mathbf{F}$ into lower-dimensional *feature chunks* suitable for vector quantization. Each chunk is mapped to a codeword index via the vector quantizer, and the indices are encoded by an arithmetic coder guided by a learnable discrete entropy model. The resulting bitstream can be transmitted or stored. On the decoder side, the indices are decoded and de-quantized using the shared codebook, and the reconstructed chunks are combined to form the reconstructed feature $\hat{\mathbf{F}}$. The entire pipeline is trained end-to-end under a rate–distortion constraint.

Vector Quantization

Vector quantization in our framework consists of two stages: (1) feature chunking, which reorganizes the original features into low-dimensional chunks, and (2) codeword matching, which maps each chunk to a learned codeword.

Feature Chunking Directly quantizing the high-dimensional feature $\mathbf{F}$ is inefficient due to (1) training instability with high-dimensional codebooks, (2) increased computational cost in codeword search, and (3) unreliable distance measures under the curse of dimensionality. To address this, we split $\mathbf{F}$ into lower-dimensional chunks:

$$\mathbf{V} \in \mathbb{R}^{n \times d}, \quad n = \frac{C \cdot H \cdot W}{d}, \tag{1}$$

where d is the chunk size and n is the total number of chunks. Each chunk $\mathbf{v}_i \in \mathbb{R}^d$ corresponds to a contiguous segment of the original feature.

In practice, the chunk size d is adapted to different backbones and tasks. For example, for ResNet50 features, each channel is treated as a chunk; for DINOv2 classification features, we set $d = 64$; and for DINOv2 segmentation features, we adjust d according to target bitrates (details in Table 2).

Codeword Matching We define a learnable codebook

$$\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K\} \subset \mathbb{R}^d, \tag{2}$$

where K is the codebook size. For each chunk $\mathbf{v}_i$, the quantizer selects a codeword $\mathbf{e}_{k(i)}$ and represents it by its index $k(i)$.

The quantization distortion of a single chunk is measured by Euclidean distance:

$$D(\mathbf{v}_i) = \left\|\mathbf{v}_i - \mathbf{e}_{k(i)}\right\|_2^2. \tag{3}$$

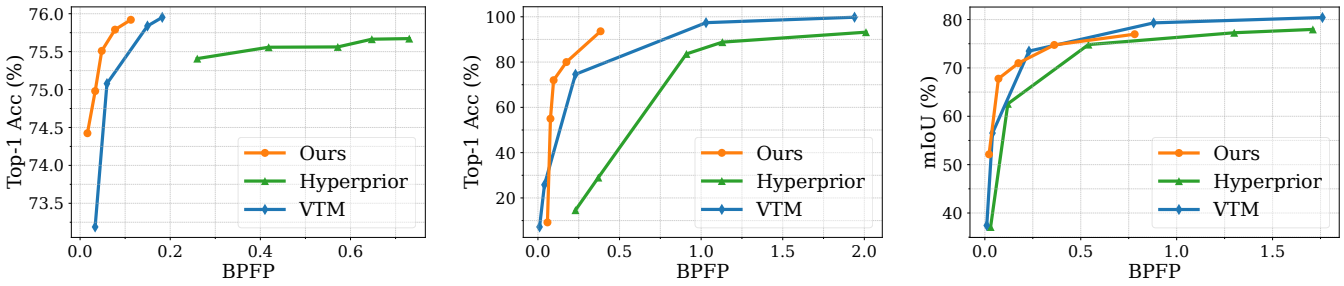
The overall reconstruction distortion between the original feature $\mathbf{F}$ and the reconstructed feature $\hat{\mathbf{F}}$ is then averaged over all chunks:

$$D(\mathbf{F}, \hat{\mathbf{F}}) = \frac{1}{n} \sum_{i=1}^{n} D(\mathbf{v}_i). \tag{4}$$

Discrete Entropy Modeling

To enable entropy coding of quantized indices, we introduce a discrete entropy model that learns the probability distribution of codeword usage. Specifically, each codeword index $k \in \{1, \ldots, K\}$ is associated with a learnable logit $a_k \in \mathbb{R}$. The probability of k is given by

$$P(k) = \frac{\exp(a_k)}{\sum_{j=1}^{K} \exp(a_j)}. \tag{5}$$



(a) (b) (c)

Figure 3: Rate-distortion performance comparisons between the proposed method and baseline methods. (a) ResNet50-Cls features. (b) DINOv2-Cls features. (c) DINOv2-Seg features.

This softmax-based formulation yields a valid categorical distribution and supports gradient-based training. The entropy model provides probability estimates for the arithmetic encoder and decoder, achieving near-optimal coding efficiency. During training, this model encourages biased codeword usage toward frequently selected indices, thereby reducing the expected bitrate.

Rate-Distortion Optimization

Standard VQ focuses solely on minimizing distortion, which is suboptimal for compression. We therefore adopt entropy-constrained vector quantization to jointly optimize bitrate and distortion. Specifically, the codeword index for each chunk is selected via

$$k(i) = \arg\min_j \left(\|\mathbf{v}_i - \mathbf{e}_j\|_2^2 - \frac{1}{\lambda} \log P(j) \right), \quad (6)$$

where $\lambda > 0$ is the rate–distortion trade-off parameter. This formulation favors codewords that are both close in Euclidean space and probable under the entropy model.

The entire framework is trained end-to-end by minimizing the following composite loss:

$$\mathcal{L} = \mathbb{E}_{\mathbf{v}} \left[-\log P(k) + \lambda \|\mathbf{v} - \mathbf{e}_k\|_2^2 \right], \quad (7)$$

where the first term approximates the coding rate and the second term measures quantization distortion. Optimizing this loss jointly updates the codebook and entropy model, yielding compact yet semantically faithful feature representations.

Experiments

In this section, we first introduce the experiment setup. Then, we compare our method with baselines in terms of rate-distortion performance and computational complexity.

Experiment Setup

Backbones, Tasks and Datasets We adopt two representative backbones to ensure feature diversity: the CNN-based ResNet50 (He et al. 2016) and the Transformer-based DINOv2 (Oquab et al. 2023). Two widely studied tasks are considered: image classification (Cls) and semantic segmentation (Seg). For ResNet50, classification features are extracted from the final Batch Normalization layer, which yields feature maps of size 2048×7×7. For DINOv2, we follow the settings in (Gao et al. 2025d), obtaining features of size 257×1536 for Cls and features of size 2×1370×1536 for Seg.

Backbone Task	Epoch	LR	Codebook Size	Chunk Size	Lambda
ResNet50 Cls	20	1e-3	2, 4, 8 32, 256	49, 49, 49, 49, 49	1, 1 1, 1, 1
DINOv2 Cls	800	1e-5	8, 16 32, 512, 2048	64, 64, 64, 64, 32	1, 1 1, 1, 1
DINOv2 Seg	800	1e-4	16, 64, 256, 512, 512	180, 50, 20, 10, 10	1, 1, 1, 1, 3

Table 2: Summary of training configurations and VQ settings.

Baseline Method We compare our approach with two widely used feature coding baselines: the handcrafted VTM-based codec and the learning-based Hyperprior codec. For DINOv2 features, we directly adopt the results reported in Gao et al. (Gao et al. 2025d). For ResNet50 features, we reproduce both baselines under the same settings to ensure fair comparisons.

Training Strategies The training configurations are summarized in Table 2. To cover different bitrate points, we vary the codebook sizes and chunk sizes accordingly. All training datasets are composed of features extracted from pretrained backbones using the same procedure. Specifically, we employ 320000, 20000, and 5000 samples for ResNet50-Cls, DINOv2-Cls, and DINOv2-Seg, respectively.

Rate-Distortion Performance

For rate-distortion evaluation, we adopt Top-1 accuracy and mean Intersection-over-Union (mIoU) to measure image classification and semantic segmentation tasks. The bitrate is measured by Bits Per Feature Point (BPFP). We present the rate-distortion performance in Figure 3 and Table 4.

Across all tasks and feature types, our method consistently achieves substantially better RD performance than the learning-based Hyperprior baseline. This advantage is particularly pronounced in scenarios with simpler tasks (e.g.,

Backbone Task	MACs (↓) Ours	MACs (↓) Hyper.	Params (↓) Ours	Params (↓) Hyper.	Enc. Time Ours (ms)	Enc. Time Hyper. (ms)	Enc. Time VTM (s)	Dec. Time Ours (ms)	Dec. Time Hyper. (ms)	Dec. Time VTM (ms)	Codebook Size	Codebook Util.(%)
ResNet50 Cls	0.4M	2.8G	0.1K	48.7M	1.5	16.1	2.1	0.6	16.7	70.6	2	100.0
	0.8M		0.2K		1.5	16.2	4.4	0.6	16.7	73.4	4	100.0
	1.6M		0.4K		1.5	16.0	7.1	0.7	16.2	75.6	8	100.0
	6.4M		1.6K		1.7	15.7	14.3	0.7	16.0	79.2	32	93.8
	51.4M		12.8K		1.7	15.6	16.6	0.7	15.9	79.2	256	80.9
DINOv2 Cls	7.9M	41.2G	0.5K	5.1M	4.0	59.8	8.6	5.8	69.1	140.0	8	100.0
	15.7M		1.0K		4.0	60.7	28.0	6.3	68.3	130.0	16	100.0
	31.5M		2.1K		4.3	56.1	124.4	6.4	63.6	150.0	32	100.0
	0.50G		33.3K		4.2	54.6	275.0	6.7	61.2	220.0	512	100.0
	1.8G		67.6K		5.7	64.1	267.2	9.1	75.9	230.0	2048	100.0
DINOv2 Seg	0.14G	354.6G	2.9K	5.1M	10.6	420.0	70.7	47.4	453.4	270.0	16	100.0
	0.55G		3.3K		28.8	405.6	407.3	82.0	448.9	320.0	64	100.0
	2.2G		5.4K		56.0	447.1	1399.7	99.0	446.0	520.0	256	100.0
	4.3G		5.6K		90.4	442.0	2039.3	160.4	488.3	940.0	512	97.3
	4.3G		5.6K		81.2	396.0	2301.3	170.5	460.6	1110.0	512	100.0

Table 3: Complexity comparisons between the proposed method and baseline methods and the codebook utilization. Our method achieves much lower computational, model, and time complexity than the baselines.

classification) or less complex backbones (e.g., ResNet50). In these cases, the feature representations contain limited semantic redundancy, rendering the transform stage in Hyperprior not only unnecessary but also detrimental–introducing additional complexity without improving compression efficiency. Moreover, compared with Hyperprior, our method reaches peak performance at significantly lower bitrates, demonstrating faster RD convergence and higher compression efficiency.

The comparison with the VTM baseline varies across feature types. For ResNet50 features, our method markedly outperforms VTM across all bitrate ranges, reflecting the suitability of VQ-based coding for compact, task-specific features. For DINOv2 features, our method still offers competitive or superior RD performance in low-to-mid bitrate regimes but shows a reduced margin at higher bitrates. Nevertheless, despite VTM's competitive performance, its computational cost is significantly higher.

Complexity Analyses

We evaluate the codec complexity in three aspects: computational complexity, model complexity, and time complexity. To ensure fair comparison, we run the Hyperprior baseline and our method on a single NVIDIA GeForce RTX 3090 GPU. For the VTM baseline, we run it on Intel® Xeon® E5-2690 v4 CPUs. Since the VTM baseline is handcrafted and cannot be run on GPUs, we only report its time complexity. All the complexity comparisons are presented in Table 3.

Computational Complexity The computational complexity is measured in multiply-accumulate operations (MACs). As shown in Table 3, our proposed method achieves a dramatic reduction in MACs. At low bitrates, our method is about 0.0001 times that of the Hyperprior model, demonstrating the effectiveness of the transform-free design. By removing the transform module, which is unnecessary for already decorrelated features, the encoder avoids substantial redundant computation.

Unlike the Hyperprior baseline, which exhibits nearly bitrate-invariant complexity, our method demonstrates bitrate-dependent scalability. This stems from our adaptive selection of codebook size and chunk size, where lower bitrates produce fewer quantized indices and thus lower entropy-coding cost. This adaptive behavior enables efficient resource utilization, particularly important for scenarios with dynamic bandwidth constraints.

As expected, the computational cost increases with the feature size because larger feature maps generate more indices for entropy coding. This trend highlights the importance of our lightweight entropy model in maintaining low complexity even for high-resolution inputs.

Model Complexity The model complexity, measured by the number of trainable parameters, is also substantially lower for our method compared to Hyperprior. The largest configuration of our method requires only tens of thousands of parameters, whereas Hyperprior requires millions. This compactness directly follows from eliminating the transform module and adopting a shared vector quantization codebook rather than separate learnable transforms. Similar to computational complexity, the parameter count of our method increases moderately with higher bitrates, as larger codebooks are employed to enhance quantization fidelity.

Time Complexity We evaluate time complexity in terms of both encoding and decoding time. Our method consistently achieves one order of magnitude lower time than Hyperprior across all bitrates, benefiting from its reduced parameters. In contrast, VTM exhibits significantly longer runtime due to its CPU-only implementation, with encoding time exceeding seconds even at moderate bitrates. This makes VTM unsuitable for real-time deployment. Additionally, VTM's encoding time grows with bitrate, reflecting the heavy cost of transform and block-based coding methods. Decoding times follow a similar trend: our method is consistently faster than Hyperprior and dramatically faster than VTM.

Method	Metric	ResNet50-Cls (Acc)					DINOv2-Cls (Acc)					DINOv2-Seg (mIoU)				
Ours	**BPFP**	0.02	0.03	0.05	0.08	0.11	0.06	0.08	0.1	0.17	0.38	0.02	0.11	0.28	0.36	0.78
	Performance	74.42	74.98	75.51	75.79	75.92	9.20	55.0	72.00	80.00	93.60	52.13	62.13	71.54	74.75	76.97
w/o Entropy Model	**BPFP**	0.02	0.03	0.07	0.09	0.15	0.13	0.15	0.16	0.18	0.19	0.39	0.49	0.59	0.69	0.79
	Performance	74.29	75.10	75.60	75.77	75.92	77.40	78.20	80.00	79.00	84.20	68.37	71.4	73.17	74.49	75.75

Table 4: Rate-distortion performance comparison between the proposed method and its variant with no entropy constraint applied.

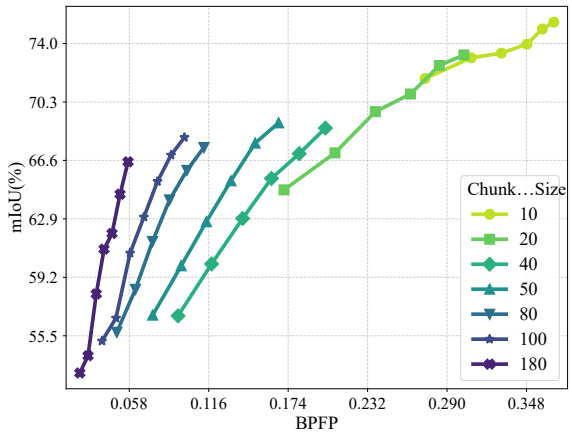


Figure 4: Rate-distortion performance comparison between the proposed method variants with various chunk sizes.

Effectiveness of VQ

The codebook utilization ratio serves as a direct indicator of the effectiveness of the VQ module. A high utilization ratio implies that the codebook entries are effectively exploited. As shown in the rightmost column of Table 3, our method achieves consistently high codebook utilization across different tasks, backbones, and bitrates.

The utilization ratio varies with the feature characteristics of different backbones. For ResNet50 features, which are trained in a supervised manner for classification, the information content is strongly class-discriminative and less diverse. Consequently, when the codebook size increases, the additional codewords are rarely used, leading to a noticeable drop in utilization (e.g., from 100% to 80.9%). In contrast, DINOv2 features, derived from self-supervised learning, encode richer and more diverse semantic information. As a result, even with larger codebook sizes (up to 2048 entries), the utilization remains nearly 100%, indicating that the increased codebook capacity continues to capture meaningful variations in the features.

Ablation Study

In this subsection, we investigate the impact of VQ and entropy model on the proposed method.

Ablation on VQ We evaluate the role of the VQ module by varying the chunk size, which determines the dimension of each quantized vector. As shown in Figure 4, the impact of chunk size on rate-distortion performance differs across bitrates. At low bitrates, increasing the chunk size leads to better performance: a larger chunk encapsulates more feature information into a single index, reducing the total number of indices and thus the overall bitrate, while the additional quantization distortion is relatively less significant. In contrast, large chunk sizes result in fewer indices, making it difficult to achieve high bitrates, and smaller chunks become more effective as the reduced quantization distortion dominates. This observation reveals a key trade-off: chunk size simultaneously controls the quantization distortion and index count, and its optimal value depends on the target bitrate. Therefore, selecting an appropriate chunk size according to the bitrate requirement is crucial for achieving the best rate-distortion performance.

Ablation on Entropy Model This ablation aims to assess the contribution of the entropy model and entropy constraint to our framework. As shown in Table 4, removing the entropy model has a negligible impact on classification performance with ResNet50 features, where the accuracy curves remain almost identical to our full method. This is because ResNet50 features, designed for classification tasks, exhibit low entropy and limited distributional complexity, making entropy modeling less beneficial. In contrast, for DINOv2-Cls and DINOv2-Seg features, integrating the entropy model provides two key benefits: (1) it enables more flexible rate control, covering a broader range of BPFP values; and (2) it yields more consistent performance at low bitrates. This demonstrates that the entropy model is particularly important for features with rich semantics and diverse distributions, such as those from self-supervised models.

Conclusion

In this paper, we revisited the necessity of the transform stage in feature coding and proposed a transform-free framework based on entropy-constrained vector quantization. By leveraging the inherent decorrelation of deep features, the method eliminates redundant computation while jointly learning the codebook and entropy model for optimal rate-distortion trade-offs. Experiments on both CNN and Transformer features across classification and segmentation tasks demonstrate superior compression efficiency and significantly lower complexity compared to transform-based baselines.

We identify several directions for future exploration. First, extending the framework to multi-task scenarios, where features from multiple tasks or modalities are encoded jointly, remains promising. Second, integrating advanced entropy models, such as context-based or hierarchical priors, may further enhance rate-distortion performance.

Acknowledgments

This work was supported in part by the National Key Research and Development Plan under Grant 2024YFF0505702. It was also supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

Agustsson, E.; Mentzer, F.; Tschannen, M.; Cavigelli, L.; Timofte, R.; Benini, L.; and Van Gool, L. 2017. Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations. *Advances in Neural Information Processing Systems*, 30.

Alvar, S. R.; and Bajić, I. V. 2021. Pareto-Optimal Bit Allocation for Collaborative Intelligence. *IEEE Transactions on Image Processing*, 30: 3348–3361.

Ballé, J.; Laparra, V.; Simoncelli, E. P.; et al. 2016. End-to-End Optimized Image Compression. *arXiv preprint arXiv:1611.01704*.

Ballé, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational Image Compression With a Scale Hyperprior. *arXiv preprint arXiv:1802.01436*.

Bross, B.; Wang, Y.-K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G. J.; and Ohm, J.-R. 2021. Overview of the Versatile Video Coding (VVC) Standard and Its Applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10): 3736–3764.

Cai, Y.; Xing, P.; and Gao, X. 2022. High Efficient 3D Convolution Feature Compression. *IEEE Transactions on Circuits and Systems for Video Technology*.

Chen, Q.; Gao, C.; Liu, D.; et al. 2024. End-to-End Learned Scalable Multilayer Feature Compression for Machine Vision Tasks. In *2024 IEEE International Conference on Image Processing (ICIP)*, 1781–1787. IEEE.

Chen, Z.; Fan, K.; Wang, S.; Duan, L.; Lin, W.; and Kot, A. C. 2019. Toward Intelligent Sensing: Intermediate Deep Feature Compression. *IEEE Transactions on Image Processing*, 29: 2230–2243.

Cheng, Z.; Sun, H.; Takeuchi, M.; and Katto, J. 2020. Learned Image Compression With Discretized Gaussian Mixture Likelihoods and Attention Modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7939–7948.

Choi, H.; Bajić, I. V.; et al. 2021. Latent-Space Scalability for Multi-Task Collaborative Intelligence. In *2021 IEEE International Conference on Image Processing (ICIP)*, 3562–3566. IEEE.

Chou, P. A.; Lookabaugh, T.; Gray, R. M.; et al. 2002. Entropy-Constrained Vector Quantization. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(1): 31–42.

Duan, L.; Liu, J.; Yang, W.; Huang, T.; and Gao, W. 2020. Video Coding for Machines: A Paradigm of Collaborative Compression and Intelligent Analytics. *IEEE Transactions on Image Processing*, 29: 8680–8695.

Feng, R.; Guo, Z.; Li, W.; and Chen, Z. 2023. NVTC: Nonlinear Vector Transform Coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6101–6110.

Friha, O.; Ferrag, M. A.; Kantarci, B.; Cakmak, B.; Ozgun, A.; and Ghoualmi-Zine, N. 2024. LLM-Based Edge Intelligence: A Comprehensive Survey on Architectures, Applications, Security and Trustworthiness. *IEEE Open Journal of the Communications Society*.

Gao, C.; Jiang, Y.; Li, L.; Liu, D.; and Wu, F. 2024a. DMOFC: Discrimination Metric-Optimized Feature Compression. In *2024 Picture Coding Symposium (PCS)*, 1–5.

Gao, C.; Jiang, Y.; Wu, S.; Ma, Y.; Li, L.; and Liu, D. 2024b. IMOFC: Identity-Level Metric Optimized Feature Compression for Identification Tasks. *IEEE Transactions on Circuits and Systems for Video Technology*.

Gao, C.; Li, Z.; Li, L.; Liu, D.; Wu, F.; and Lin, W. 2025a. Rethinking Joint Optimization in Feature Compression: Insights from Person Re-Identification. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6.

Gao, C.; Liu, S.; Wu, F.; and Lin, W. 2025b. Cross-architecture universal feature coding via distribution alignment. 1–5.

Gao, C.; Liu, Z.; Li, L.; Liu, D.; Sun, X.; and Lin, W. 2025c. DT-UFC: Universal Large Model Feature Coding via Peaky-to-Balanced Distribution Transformation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 5198–5207. Association for Computing Machinery.

Gao, C.; Ma, Y.; Chen, Q.; Xu, Y.; Liu, D.; and Lin, W. 2025d. Feature Coding in the Era of Large Models: Dataset, Test Conditions, and Benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1068–1077.

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

He, D.; Yang, Z.; Peng, W.; Ma, R.; Qin, H.; and Wang, Y. 2022. ELIC: Efficient Learned Image Compression With Unevenly Grouped Space-Channel Contextual Adaptive Coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5718–5727.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Jia, Z.; Li, J.; Li, B.; Li, H.; and Lu, Y. 2024. Generative Latent Coding for Ultra-Low Bitrate Image Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26088–26098.

Kim, Y.; Jeong, H.; Yu, J.; Kim, Y.; Lee, J.; Jeong, S. Y.; and Kim, H. Y. 2023. End-to-End Learnable Multi-Scale Feature Compression for VCM. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(5): 3156–3167.

Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; and Chen, Z. 2020. GShard: Scaling Giant Models With Conditional Computation and Automatic Sharding. *arXiv preprint arXiv:2006.16668*.

Liu, J.; Sun, H.; Katto, J.; et al. 2023a. Learned Image Compression With Mixed Transformer-CNN Architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14388–14397.

Liu, K.; Liu, D.; Li, L.; Yan, N.; and Li, H. 2021. Semantics-to-Signal Scalable Image Compression With Learned Revertible Representations. *International Journal of Computer Vision*, 129(9): 2605–2621.

Liu, T.; Xu, M.; Li, S.; Chen, C.; Yang, L.; and Lv, Z. 2023b. Learnt Mutual Feature Compression for Machine Vision. In *ICASSP 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Ma, Y.; Gao, C.; Chen, Q.; Li, L.; Liu, D.; and Sun, X. 2024. Feature Compression With 3D Sparse Convolution. In *2024 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 1–5.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. 2023. Dinov2: Learning Robust Visual Features Without Supervision. *arXiv preprint arXiv:2304.07193*.

Qi, L.; Jia, Z.; Li, J.; Li, B.; Li, H.; and Lu, Y. 2025. Generative Latent Coding for Ultra-Low Bitrate Image and Video Compression. *IEEE Transactions on Circuits and Systems for Video Technology*.

Qu, G.; Chen, Q.; Wei, W.; Lin, Z.; Chen, X.; and Huang, K. 2025. Mobile Edge Intelligence for Large Language Models: A Contemporary Survey. *IEEE Communications Surveys & Tutorials*.

Suzuki, S.; Takeda, S.; Takagi, M.; Tanida, R.; Kimata, H.; and Shouno, H. 2022. Deep Feature Compression Using Spatio-Temporal Arrangement Toward Collaborative Intelligent World. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6): 3934–3946.

Tian, Y.; Wan, Y.; Lyu, L.; Yao, D.; Jin, H.; and Sun, L. 2022. FedBERT: When Federated Learning Meets Pre-training. *ACM Transactions on Intelligent Systems and Technology*, 13(4): 1–26.

Tian, Y.; Zhang, Z.; Yang, Y.; Chen, Z.; Yang, Z.; Jin, R.; Quek, T. Q. S.; and Wong, K.-K. 2024. An Edge-Cloud Collaboration Framework for Generative AI Service Provision With Synergetic Big Cloud Model and Small Edge Models. *IEEE Network*, 38(5): 37–46.

Vepakomma, P.; Gupta, O.; Swedish, T.; and Raskar, R. 2018. Split Learning for Health: Distributed Deep Learning Without Sharing Raw Patient Data. *arXiv preprint arXiv:1812.00564*.

Wang, S.; Wang, S.; Zhang, X.; Wang, S.; Ma, S.; and Gao, W. 2019. Scalable Facial Image Compression With Deep Feature Reconstruction. In *2019 IEEE International Conference on Image Processing (ICIP)*, 2691–2695. IEEE.

Wang, Y.; Yang, C.; Lan, S.; Zhu, L.; and Zhang, Y. 2024. End-Edge-Cloud Collaborative Computing for Deep Learning: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 26(4): 2647–2683.

WG2. 2023. Call for Proposals on Feature Compression for Video Coding for Machines. *Standard ISO/IEC JTC 1/SC 29/WG 2, N282*.

Wu, X.; Hou, X.; Lai, Z.; Zhou, J.; Zhang, Y.-n.; Pedrycz, W.; and Shen, L. 2025a. A codebook-driven approach for low-light image enhancement. *Engineering Applications of Artificial Intelligence*, 156: 111115.

Wu, X.; Lai, Z.; Hou, X.; Zhou, J.; Zhang, Y.-n.; and Shen, L. 2025b. LightQANet: Quantized and Adaptive Feature Learning for Low-Light Image Enhancement. *arXiv preprint arXiv:2510.14753*.

Yamazaki, M.; Kora, Y.; Nakao, T.; Lei, X.; and Yokoo, K. 2022. Deep Feature Compression Using Rate-Distortion Optimization Guided Autoencoder. In *2022 IEEE International Conference on Image Processing (ICIP)*, 1216–1220. IEEE.

Yan, N.; Gao, C.; Liu, D.; Li, H.; Li, L.; and Wu, F. 2021. SSSIC: Semantics-to-Signal Scalable Image Coding With Learned Structural Representations. *IEEE Transactions on Image Processing*, 30: 8939–8954.

Yang, W.; Huang, H.; Hu, Y.; Duan, L.-Y.; and Liu, J. 2024. Video Coding for Machines: Compact Visual Representation Compression for Intelligent Collaborative Analytics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7): 5174–5191.

Zhang, M.; Shen, X.; Cao, J.; Cui, Z.; and Jiang, S. 2024. Edgeshard: Efficient LLM Inference Via Collaborative Edge Computing. *IEEE Internet of Things Journal*.

Zhu, X.; Song, J.; Gao, L.; Zheng, F.; and Shen, H. T. 2022. Unified Multivariate Gaussian Mixture for Efficient Neural Image Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17612–17621.