Implicit Feature Decoupling with Depthwise Quantization

Iordanis Fostiropoulos
University of Southern California
Los Angeles, CA
fostirop@usc.edu

Barry Boehm
University of Southern California
Los Angeles, CA
boehm@usc.edu

Abstract

Quantization has been applied to multiple domains in Deep Neural Networks (DNNs). We propose Depthwise Quantization (DQ) where quantization is applied to a decomposed sub-tensor along the feature axis of weak statistical dependence. The feature decomposition leads to an exponential increase in representation capacity with a linear increase in memory and parameter cost. In addition, DQ can be directly applied to existing encoder-decoder frameworks without modification of the DNN architecture. We use DQ in the context of Hierarchical Auto-Encoders and train end-to-end on an image feature representation. We provide an analysis of the cross-correlation between spatial and channel features and propose a decomposition of the image feature representation along the channel axis. The improved performance of the depthwise operator is due to the increased representation capacity from implicit feature decoupling. We evaluate DQ on the likelihood estimation task, where it outperforms the previous state-of-the-art on CIFAR-10, ImageNet-32 and ImageNet-64. We progressively train with increasing image size a single hierarchical model that uses 69% fewer parameters and has faster convergence than the previous work.

1. Introduction

Quantization is an effective lossy compression process that maps a continuous signal to a set of discrete values, also called *codes*. Quantization is extended to vector feature spaces with learning paradigms such as *Vector Quantization* (VQ) and with a training objective identical to k-means. *Product Quantization* (PQ) decomposes the feature vector and assumes a weak statistical dependence between feature sub-vectors. *Additive Quantization* (AQ) decomposes the feature vector into a sum of quantized vectors as opposed to the concatenated output in PQ.

Quantization is used in conjunction with Deep Neural Networks (DNNs) for tasks such as classification [46], incremental learning [46], zero-shot learning [28], generation [33],

Figure 1. The original images (left) are reconstructed by DQ (middle) and VQ (right) with identical models and training setup. The perceptual quality of DQ outperform VQ.

compression [29] and data retrieval [4]. The discrete quantized feature representations can be used post-hoc [11,33,38] or as a learning objective (i.e. classification) [28, 46]. Our work is motivated by the increasing number of quantization applications to high dimensional feature tensors. We view the quantizer as a density estimator and evaluate it on the task of likelihood estimation for the visual domain.

Likelihood estimation models seek to minimize the *divergence* between the data distribution and the model prior. Explicit likelihood estimation models, including Vector Quanti-
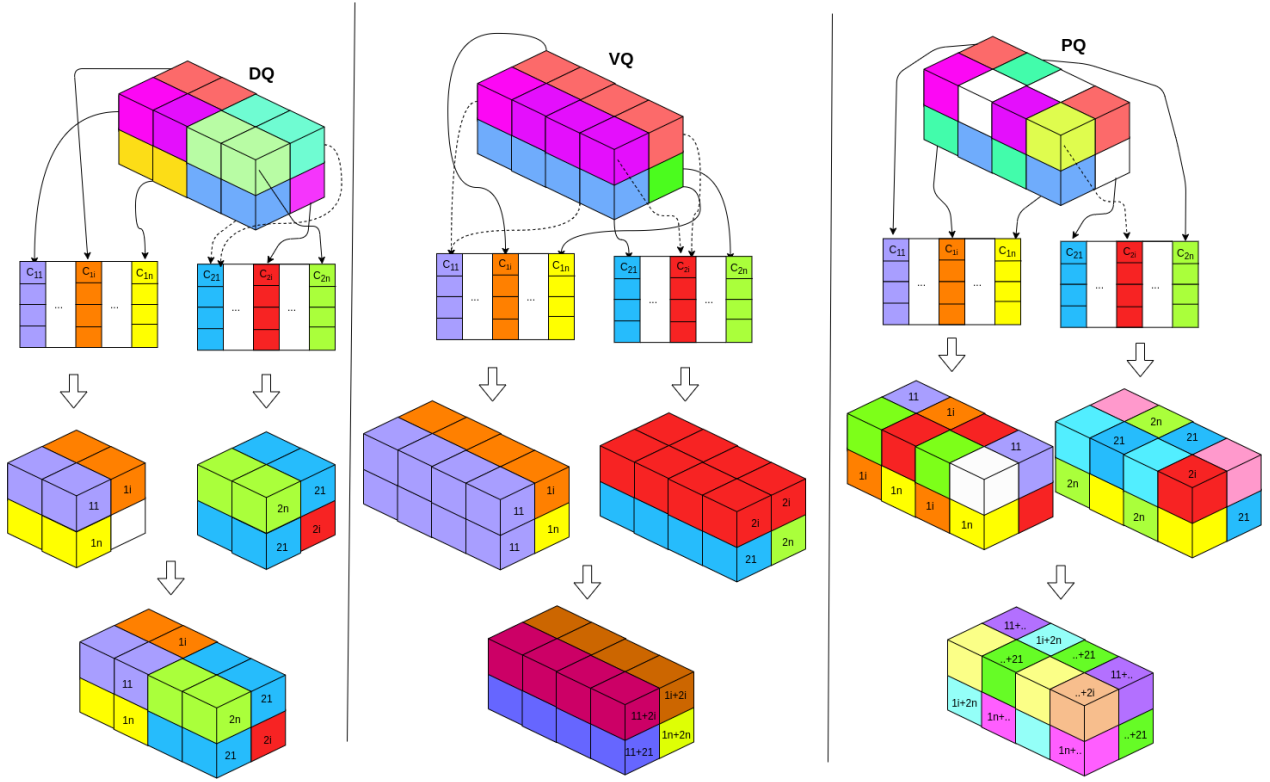
Figure 2. DQ (left) apply C_1 on the first slice of the sub-tensor and for all sub-vectors and concatenate the quantized vectors. VQ [33] (middle) and PQ (right) quantize the same vector with different codebook and combine the two sub-vectors by addition or concatenation.

zation (VQ), Variational Auto-encoders (VAEs), and Auto-Regressive (AR) models directly minimize a divergence. In this work, we focus on explicit likelihood estimation.

AR models do well in likelihood estimation and are applied in multiple domains such as language, vision, and audio. AR models have a recursive dependency on the input during training and inference. Therefore, AR models are computationally inefficient for domains with long sequences, such as pixels of an image. Even with caching [34] during sampling, AR models are still less efficient than VAEs.

The priors of VAEs provide a compressed feature representation that can be used as a surrogate training objective for the downstream task. In contrast to the discrete prior, a continuous prior can lead to *posterior collapse*. The representation is ignored by the downstream task model because it is either too noisy or uninformative. This effect is amplified when the data is discrete, as in the language domain [14].

To that end, we propose the Depthwise Quantization (DQ) method that quantizes each decomposed feature sub-tensor with a different quantizer. We use rate-distortion theory to interpret a quantizer as an encoding function with limited capacity. We provide a theoretical upper bound on the capacity in relation to the quantization cost when DQ is applied on a decoupled feature tensor, as opposed to a coupled feature ten-

sor. We evaluate the performance of DQ on the feature space of ImageNet for an image classification backbone. Lastly, we apply DQ to a hierarchical Auto-Encoder with DQ as a bottleneck for different hierarchies and train it end-to-end. DQ outperforms explicit models in likelihood estimation. In detail:

- We propose Depthwise Quantization (DQ) and decompose a feature tensor along the axis of weak statistical dependence.

- We provide a theoretical analysis on the improved quantization performance and experimentally corroborate our theoretical results.

- We introduce an improved hierarchical AutoEncoder model Depth-Quantized Auto-Encoder where DQ is applied to the feature representation at different hierarchies.

- We extend the parametric Mutual Information (MI) quantization estimators for DNNs when the prior is *learned*. We experimentally verify that the learned prior is *implicitly decoupled*.

Our approach can be applied to previous works that use quantization. We demonstrate with our experiments that DQ

performs significantly better when the assumption on cross-correlation is strong in both post-hoc analysis and end-to-end training settings. When trained end-to-end, DQ reduces the cross-correlation among the decomposed feature tensors ("implicitly decouple") and improves reconstruction loss and likelihood estimation. Our code is publicly available[1].

2. Related Work

Our work is closely related to previous studies on feature decomposition and quantization optimization in visual tasks using DNNs.

Feature decomposition approaches include *Separable Convolutions* (SP) [31], which factorizes a convolutional kernel to the *spatial* dimensions. SP reduces the number of computations required to calculate the filter output. Inception [43], another approach to feature decomposition, factorizes a feature representation implicitly with a "Network-in-Network" (NiN) [27] branch of convolutions. Thus, Inception learns spatial cross-correlations and feature cross-correlations independently. *Depthwise* Separable Convolution (DSC) [9, 43] is a method of feature decomposition that uses a single "spatial" convolution followed by multiple vanilla convolutions on a decomposed "segment". Xception [9] is based on the "Inception Hypothesis" for a decoupled space where DSC is applied to an *eXtreme*. Our work is based on a hypothesis similar to that of DSC and considers modeling cross-channel correlations and spatial correlations independently.

There are analyses on the *decoupling* of the feature space *implicitly* in the context of DSC as well as on NiN architectures. Blueprint Separable Convolutions (BSConv) [16] have been proposed as an alternative to DSC based on the observation of intra-kernel correlations. They propose a *pointwise* (1x1) convolution followed by a depthwise convolution. In contrast, DSC enforces cross-kernel correlations *implicitly*. Analysis on the variance of a convolution kernel shows that a DNN can perform better when cross-kernel redundancies decrease.

Other works *explicitly* factorize a convolutional filter. There are methods that use a low-rank approximation [20] or closed-form decomposition [15] on pre-trained networks to speed up the computation process. Previous works on speeding up networks [42] have used product quantization to quantize convolutional filters and take advantage of the redundancies. Previous analysis of the redundancy and cross-correlation of the feature space in DNNs is complementary to our work.

Improvements in quantization learning approaches include Optimized Product Quantization [13] that decomposes the feature vector in a parametric manner. In addition, Ad-ditive Quantization [2] improves on the computational efficiency of PQ for high dimensional vector search by decomposing the vectors into a sum instead of a concatenation of their sub-vectors. In contrast, our work can be applied to feature tensors and the quantizer is trained end-to-end with a DNN.

Kobayashi *et al.* [25] train a quantizer end-to-end with a DNN. They use multiple codebooks and train each codebook independently for a different supervised task. However, as opposed to our method, the codebooks are decoupled in a supervised manner. Moreover, the quantized representations are used by different networks for different downstream tasks as opposed to interacting for a single downstream task. Lastly, vector decomposition is applied to feature vectors as opposed to feature sub-tensors as in our work.

PQ-VAE [45] also decomposes the latent representations to sub-vectors and uses different quantizers for each sub-vector. Kaiser *et al.* [21] introduces "sliced quantization" that is identical to PQ-VAE but uses the discrete representation post-hoc with a latent variable model. By contrast, DQ decomposes the feature space to sub-tensors as opposed to sub-vectors and thus improves the reconstruction loss by implicitly increasing the statistical dependence within the sub-tensor.

The works most similar to ours are those of Razavi *et al.* [38] and Dhariwal *et al.* [11]. VQ-VAE-2 [38] applies quantization to the feature representation of multiple hierarchies on a VAE. Similar to our method, they train VQ end-to-end with a VAE. However, we apply DQ as opposed to VQ for the quantization method. VQ-VAE-2 can suffer from an uninformative top prior. Subsequent models such as "Jukebox" [11] mitigate the issue by modeling each hierarchy with an independent encoder-decoder architecture. We also avoid the issue of an uninformative top prior but do not model each hierarchy with a different model. Instead, we introduce a model architecture DQ-AE.

3. Background

Auto-Encoder (AE) is an unsupervised class of DNN architectures that learns compressed feature representations from high dimensional data. Work by Kingma *et al.* [22] extends AE to *Deep Latent Variable Models* with variants such as Variational Auto-Encoder (*VAE*). For some input x and a latent space z, VAE is composed of a decoder $p(x|z)$, a prior $p(z)$, and an encoder $q(z|x)$. VAE is a probabilistic model that implicitly learns underlying variables used to generate the data and their *latent factors* by minimizing the divergence between the encoded representation $q(z|x)$ and the true data manifold $p(z)$. To evaluate AE, we can use Mutual Information (MI) which is a statistical dependence metric between two variables s.t. $I(X;Y) = H(X) - H(X|Y)$, where $H(X)$ is the *information entropy* of X. The optimization objective of a VAE [3, 18] is an upper
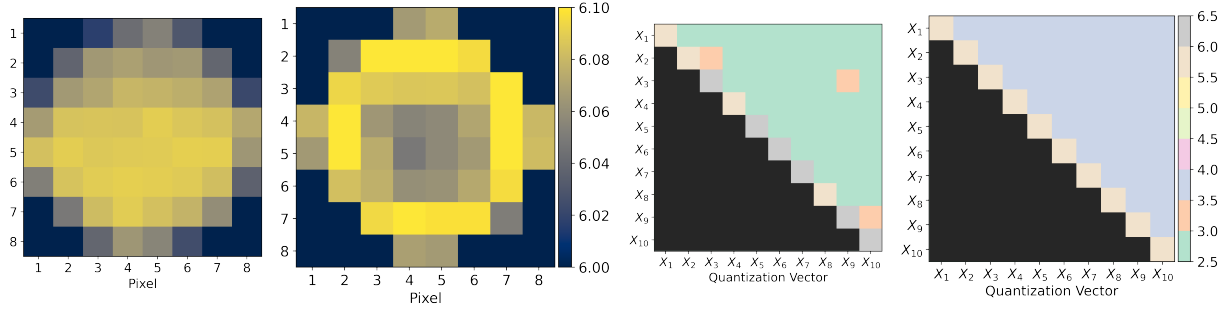

Figure 3. Effects of implicit marginalization on the learned quantized features. The mean entropy ("informativeness") of the quantization vectors for each pixel for DQ (first) is higher when compared to VQ (second); higher is better. The mean MI ("redundancy") scores between each quantization codebook for DQ (third) is lower when compared to VQ(fourth); lower is better. The diagonal represent the entropy for each quantization vector, lower half of the diagonal is empty.

bound to

$$max[I(z; p(x|z)) - \beta I(x; z)] \quad (1)$$

that maximizes the mutual information between latent representation and decoded data, and discards information from x that is not informative to decoding $p(x|z)$. As such, maximizing Eq. (1) also maximizes the entropy of z or "informativeness" [19].

Our view on quantization is based on the interpretation by Richardson *et al.* [39] and MacKay *et al.* [30]. For the sake of brevity, we refer the reader to their work for a detailed analysis and attach our own analysis and proofs in the supplementary material.

Scalar Quantizer (SQ) with a vocabulary of size K is an encoding function for an element X_i from a sequence $X \in R^N$ of length N such that $f(X) = \{1 \ldots K\}^N$. SQ quantizes every element of the sequence in a *memoryless* fashion with the same encoding function. SQ cannot place assumptions on cross-correlation between different sequence elements. SQ performs optimally when the probability density function (pdf) of all data is known in advance. An encoding function that follows a uniform distribution (i.e. floor function) will perform optimally when all $X_{Ui} \in X_U$ are also uniformly distributed and bounded such that $X_{Ui} \in [a, b]$. When X_{Ui} has an unknown pdf, SQ will *assign* probability *mass* on unlikely regions in $[a, b]$.

Vector Quantization (VQ) "learns" a mapping between $X \in R^N$, and K quantization vectors, or *codes*. A **Codebook** is the set of *codes* $c \in R^N$ such that $C = \{c_i : i \in 1, \ldots, K\}$. The VQ *decoding* function returns the code c with the lowest *decoding error* d between c and the vector X such that $\hat{X} = VQ(X) = c_{j_{min}}$ where $j_{min} = argmin\{d(X, c) : c \in C\}$. The objective function is to minimize the error of the closest codebook vector c to the feature vector X and can be summarized as $l_{VQ} = \min_{c \in C} d(X, \hat{X})$.

Product Quantization (PQ) decomposes a one dimensional vector $X \in R^N$ to sub-vectors $\{X_j : j = 1, \ldots, M\}$ and optimizes for a unique pair of a VQ and the sub-vector space. For M different Codebooks $C_j : j \in 1, \ldots, M$ there is a one-to-one mapping with each X_j. The PQ decoding function is the concatenation or addition of all VQ decoding $VQ_j = \hat{X}_j$ for codebook C_j such that $PQ(X) = \|_{\forall j \in M} VQ_j(X_j)$. We adopt the *feature decomposition* from PQ and extend it to high dimensional feature vectors to reduce the statistical independence among latent features.

Cost of a quantizer is the number of Codebook vectors s.t. $C_{cost} = K \times M$, for PQ. *Representation Capacity* (C_R) defines an upper bound on the sample space from the number of discrete latent factors that can be represented by the quantizer for independent random variables X_j, such that $S = K^M$ for K codes and M decomposed sub-vectors. For redundant X_j, the sample space is reduced to $S^{new} = (K-1)^M$ and thus the capacity is bounded by the sample space s.t.

$$C_R = -H(\mathbf{X}) \quad (2)$$

Note that for PQ, C_{cost} grows linearly while C_R grows exponentially, in contrast to a VQ which has linear growth for both, and thus has an exponential cost with an identical capacity to PQ. More detailed analysis and proofs can be found in the Appendix.

Distribution of Prior can have an effect on the decoding performance of the quantizer. For example, VQ with X_U from before can achieve identical decoding error as SQ but at a significant cost of K^N as compared to K for a *memoryless* SQ. The assumption on the distribution of the prior can determine the cost and the representation capacity of a quantizer.

The difference between PQ and VQ is the assumption of co-variance among features. Contrary to VQ, PQ takes advantage of the low co-variance among feature sub-vectors.
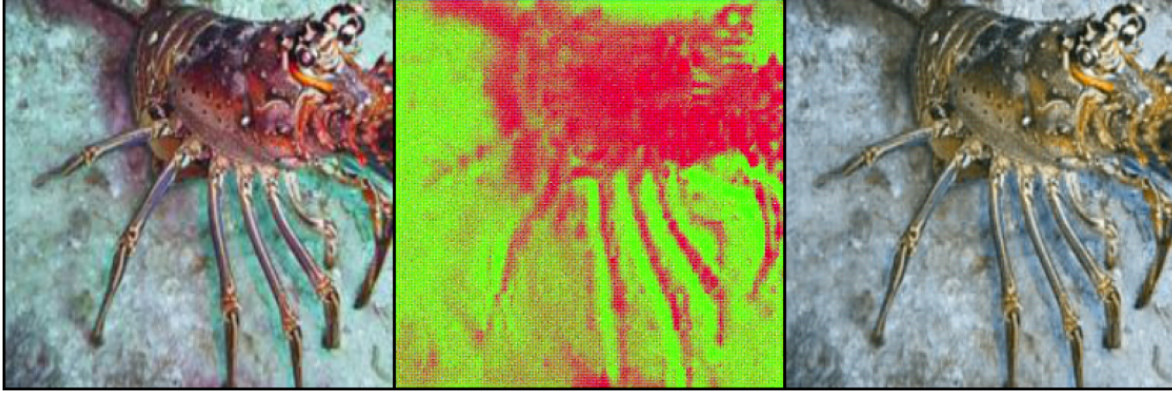
Figure 4. Original image (left) is reconstructed using only **top** level codes (middle) and only **bottom** level codes (right). Top level hierarchy contains structural information, while bottom level hierarchy contains details.

4. Depthwise Quantization

Given an output feature tensor from encoder $\mathbf{X} \in R$ with rank r, Depthwise Quantization (DQ) applies M quantizers VQ_i pair-wise on decomposed tensor *slices* $\mathbf{X}_i = \mathbf{X}_i^{\alpha}$ along an axis α with quantization *dimension* $D = |\mathbf{X}_i^{\alpha}|$.

$$\text{DQ}(\mathbf{X}) = \{VQ_i(\mathbf{X}_i) : i = 1, \ldots, M\} \quad (3)$$

Each VQ_i optimizes Codebook C_i and uses l_{VQ} to define the error between $\mathbf{X}_i$ and closest quantization vector $\hat{\mathbf{X}}_i = Q_i(\mathbf{X}_i)$. The optimization objective is the joint optimization over each codebook such that

$$\min_{C^1, \ldots, C^M} [L_{DQ} = \sum_{\forall \hat{\mathbf{X}}_i, \mathbf{X}_i} l_{VQ}(\mathbf{X}_i, \hat{\mathbf{X}}_i)] \quad (4)$$

We use the L_2 norm as a similarity metric for l_{VQ} between each $\mathbf{X}_i$ and the *local* quantization vector $\hat{\mathbf{X}}_i$. The DQ loss is then added to the reconstruction loss of the DNN and the gradients are copied from the quantized vector $\hat{\mathbf{X}}_i$ to X using auto-grad [37]. The loss function of DQ becomes

$$L = L_{DNN} + L_{DQ}(sg(\mathbf{X}), \hat{\mathbf{X}}) + \beta L_{DQ}(\mathbf{X}, sg(\hat{\mathbf{X}})) \quad (5)$$

where sg stands for stop-gradient operator that stops the operand from updating during the training phase. Similar to the setting in VQ-VAE, the first loss term is used to lower the reconstruction error, the second term adjusts the codebook corresponding to the encoder output, and the third term is used to prevent the output of the encoder from growing arbitrarily. Note that the KL divergence is a constant equal to $Mlog(K)$ as DQ assumes a uniform prior distribution of latent embeddings. Therefore, the KL divergence term is dropped from the optimization objective of our framework. A detailed explanation is provided in Sec. 4.1

Note that for a feature tensor of rank one, DQ is identical to SQ when a single codebook of dimensionality one is used.

When more than one codebook is used, DQ is identical to PQ and Additive Quantization with addition as the decoding function. The advantage of DQ over other quantization methods comes from the decomposition of a tensor to sub-tensors along the axis of weak statistical dependence.

For 2-D Convolutional Neural Networks (CNN), $\mathbf{X}$ is a feature tensor of rank 3. Fig. 2 provides an illustration of the DQ process for a 3-rank tensor. Different quantizers are applied for each slice of the channel axis, but the same quantizer is applied on the sub-vectors of a feature sub-tensor, such as decomposing along the spatial dimension.

4.1. Decoupled Feature Space

Decoupled refers to the statistical independence between features and **Coupled** refers to the statistical dependence between features. We use *Information Theory* to analyze quantization as an encoding function with an information bottleneck on a signal.

Eq. (1) provides the basis of the VAE optimization objective that can be formulated as a lower bound to the channel capacity as $\mathbb{L} \geq \mathbb{E}_{q(z|x)} \log p(x|z) - \beta D_{KL}(q(z|x)||p(z))$ [3, 18] where β is the Lagrange multiplier. β-VAE assumes a Gaussian prior $p(z) \sim \mathcal{N}(0, I)$, and DQ assumes a uniform prior. Thus the KL-Divergence of the uniform distribution and decoder is the capacity of the quantizer $D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) = C_R$. The detailed proof can be found in Appendix 1.

$$max[\mathbb{E}_{q(z|x)} \log p(x|z) - C_R] \quad (6)$$

Reducing the capacity of the information bottleneck in VAE encourages disentangled representations in β-VAE. In a similar fashion, reducing C_R encourages disentangled representations for each codebook, with the upper bound controlled by K and M. By doing so, significantly compressed representations can be learned for an improved downstream training objective.

C_R and by extension $H(z)$ in the discrete case are not differentiable with respect to the DNN parameters and can not be explicitly minimized. We observe *index collapse* and performance degradation on hierarchical deep quantization variants. Index collapse causes the quantizer to utilize only limited number of codes. We enforce a uniform prior through an approximate solution and discuss the implications in Sec. 4.2 and Sec. 6.

4.2. Implicit Decoupling

Implicit decoupling of the feature space is the surrogate optimization objective derived by the explicit minimization of the decoding error. There is a joint optimization objective when DQ is applied to the intermediate feature representations in the context of DNN and trained end-to-end. DQ minimizes the decoding error along the DNN objective function. DQ works as a bottleneck on intermediate feature representations between subsequent layers of the network. We use the result from work by β-VAE on the interpretation of AE as the information bottleneck.

β-VAE uses $q(z|x)$ to learn a set of additive channels z_i where their capacity is maximized when all z_i are independent. This provides an implicit optimization objective by optimizing Eq. (1), which is the *equivalent* in the quantized case as optimizing Eq. (6). There is an equivalence between each z_i and the codebook as they both perform as additive information channels that, when combined, reconstruct an original signal. Lastly, both z_i and the quantizer are parametric density estimators or smaller networks that can be considered as part of a generic Network-in-Network (NiN) family of models.

Feature independence improves downstream task performance when learned implicitly in NiN models. Xception uses Depthwise Seperable Convolution (DSC) to outperform *coupled* variants on ablation studies on Mobile-Net [17]. Additional previous analysis on the intra-kernel correlations [16] has demonstrated the benefits of a decoupled feature space along the channels of an image feature tensor. We corroborate the analysis with MI estimation on a static prior to determine the axis of weak statistical dependence and apply DQ along the channel dimension ("depth-wise") and spatial dimension ("pixel-wise") in the context of DNN.

Uniform Prior In contrast to the traditional Variational Auto-Encoder, DQ relies on the assumption of uniform distribution of quantized vectors $p(z)$. However, the assumption of a uniform prior is not strong, which can potentially lead to degrading performance and be sensitive to the random initialization. A non-uniform prior will cause *codebook collapse* where only few codes are utilized in a codebook. To mitigate this issue, we follow previous work, and use Exponential Moving Average (EMA) and random re-initialization of codes. We re-initialize codes with low usage frequency counts that are below a threshold. Although pre-

vious works [5, 21] discuss the equivalence of quantization with EMA and the β-VAE objective, there is no exact relationship between the two. VQ-VAE is an approximation to the Varitional Information Bottleneck (VIB) when trained with soft Expectation Maximization (EM). The E-step on the update rule of DQ is approximated with EMA over mini-batches of data [5, 6]. This is in contrast to hard - EM where quantization is deterministic [40]. Soft - EM provides a probabilistic discrete information bottleneck as discussed in work by Roy *et al.* [41] and Wu *et al.* [44].

We use entropy of the quantization vectors to measure their information density. A successful decoupling method should generate feature vectors with high entropy. **Entropy Estimation** on continuous distributions is intractable, but a signal can be discretized by quantization with both parametric and non-parametric optimization on the quantization interval. The entropy is then computed on the quantized discrete distribution.

$$I(X;Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p_{(X,Y)}(x,y) \log \left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\, p_Y(y)} \right) \tag{7}$$

We use the quantization regions of VQ as a density estimator for entropy, and thus mutual information on a continuous prior. When DQ is learned end-to-end, entropy can be calculated directly by the frequency count of each code vector over a sample set. Our approach in approximating MI is similar to previous work that uses Kernel Density Estimators [32] and is performed post-hoc on a trained network or by training a different DQ. Quantization post-hoc is sensitive to sample size but performs at par with other state-of-the-art approaches [12, 23, 24, 35].

4.3. Depth-Quantized AutoEncoder

Depthwise Quantized Auto-Encoder (DQ-AE) uses DQ at different hierarchical feature representations. The full algorithm that defines the training process is found in the supplementary material. In summary, we decode each quantized representation conditioned only on the quantized representation of the previous level. We perform this operation top-bottom and use Eq. (6) as the optimization objective of each DQ. Through experiments, we find that lower capacity *bottom*-level hierarchy enforces the utilization of top-level hierarchies and that the problem of under-utilization of top or bottom level hierarchies can also be a consequence of over-fitting. During the early stages of training, both hierarchies are used equivalently, but at later stages, *top*-level prior collapse. Our architecture leads to informative top and bottom level hierarchies as can be seen in Fig. 4.
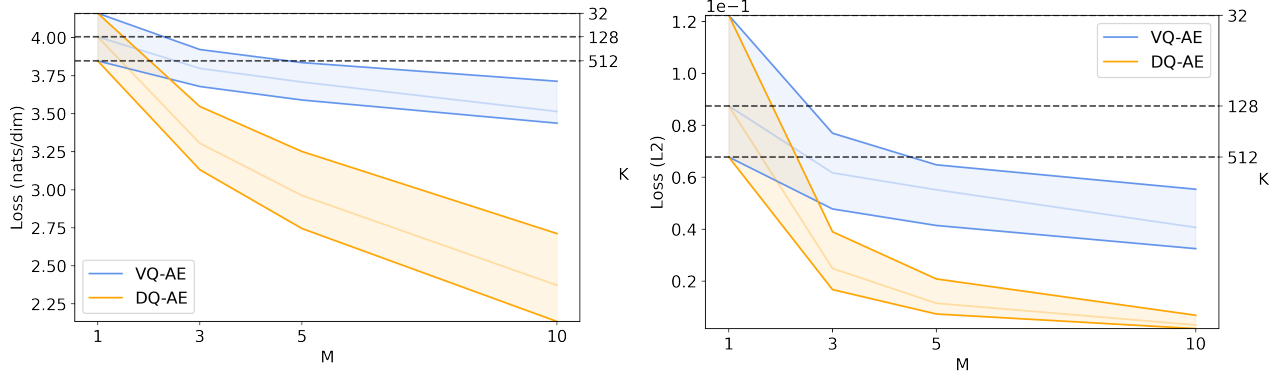
Figure 5. Ablation study model comparison. For M=1, DQ is identical to VQ. We train models under the following settings $K = \{32, 128, 512\}$, $M = \{1, 3, 5, 10\}$ and optimize for different reconstruction losses NLL (left) and L_2 (right). We report the average of the final convergence loss from multiple runs (10). The top bold line for each polygon corresponds to $K = 32$, the bottom line to $K = 512$

	CIFAR-10		**ImageNet-32**		**ImageNet-64**
Model (Param.)	bits/dim	Param.	bits/dim	Param.	bits/dim
S-Tr.[1] (59M)	2.80	Img-Tr.[2] (-)	3.77	S-Tr.[1] (152M)	3.44
VDVAE[3] (39M)	2.80	119M	3.80	125M	3.52
(**Ours**) (**22M**)	**2.52**	**22M**	**3.12**	**22M**	**2.89**

Table 1. Baselines: [1]Sparse Transformer [8] [2]Image Transformer [36] [3]VD-VAE [7]. "Ours" is a 2 hierarchical DQ-AE with K set to 256 and 128 for top and bottom codebooks respectively.

5. Experiment

In our experiment, we evaluate DQ in two settings: on a static prior, and when trained end-to-end with a DNN, on a learned prior. We first evaluate our theoretical claim on a static prior and perform an ablation study on DQ and DQ-AE. We report the details of the training and network hyper-parameters in the supplementary material.

5.1. Density Estimation

We experimentally verify our claims on the decoupled feature space from Sec. 4.1. The penultimate feature representation from pre-trained VGG-16 model[2] is used on ImageNet [10]. DQ decomposes the feature representation "channel-wise" (DQ_C) and "pixel-wise" (DQ_S). The penultimate feature tensor with shape $[512 \times 7 \times 7]$ is sliced along the channel axis into 7 segments and zero padded with $D = 74$ and $M = 7$. The quantizers for both networks independently quantize each row for DQ_S in contrast to each slice for DQ_C.

We train DQ as a quantizer for multiple random runs (10) and we report the mean L_2 norm between $\mathbf{X}$ and the reconstruction $\hat{\mathbf{X}}$. For each quantization method, we approximate the entropy of the codes to determine their respective information density using Eq. (7). The results of our experiments can be found in Tab. 2.

_[2] https://pytorch.org/vision/stable/models.html

Quant.	K	D	$L_{DQ} \downarrow$	$\sim H(\mathbf{X}) \uparrow$
Pixel	32	74	0.192±0.002	1.98±0.01
Channel	32	74	**0.184**±0.001	**2.53**±0.01
Pixel	1024	74	0.523±0.003	3.64±0.01
Channel	1024	74	**0.480**±0.001	**3.99**±0.01

Table 2. Density estimation on ImageNet feature space extracted from VGG-16. Results are from 10 train runs with random code initialization. DQ applied along the channels (Channel Quantization) as opposed to the spatial dimension (Pixel Quantization). D is the size of the feature vector, K the discrete codes used, L_2 the reconstruction error, and $H(\mathbf{X})$ is the mean entropy of the feature tensor. Channel Quantization surpasses Pixel Quantization in all respects.

We find that DQ can achieve better density estimation along the channel dimension as opposed to the spatial dimension. The lower entropy $H(\mathbf{X})$ of the feature tensor is due to a higher redundancy among feature sub-tensor and corresponds to a higher reconstruction error. DQ can perform better when decomposing on the channel axis and our results agree with previous analyses on intra-kernel correlations [16].

5.2. Implicit Decoupling

We train a DQ-AE and a VQ-VAE [33] for $M = 10$ and $K = 512$ with an identical network configuration, methodology and hyper-parameters. The difference between architectures is highlighted in Fig. 2. We measure the likelihood estimation of the two approaches on CIFAR-10 [26] and quantize each image to $8 \times 8 \times 10$ codes. VQ-VAE NLL is **4.36** bits/dim as compared to **3.14** bits/dim for single hierarchy DQ-AE, a 28% decrease.

High Entropy We show that the learned features of DQ have high entropy which indicates low statistical dependence among them. In contrast, VQ appears to have few very informative features and many uninformative ones. The mean entropy of the prior is $H(z) = 6.03$ nats/pixel for DQ as compared to $H(z) = 5.86$ nats/pixel for VQ. The entropy distribution among spatial features of the prior can be found in the left two sub-figures in Fig. 3.

Low MI We estimate the pairwise MI of the quantization vectors along the depth of the feature tensor with mean score of 1.93 and 2.36 nats/vector respectively. A comparison matrix can be found in the right two sub-figures in Fig. 3. For DQ the MI between quantization vectors is significantly lower as visualized by the mostly empty upper triangular matrix. In contrast, for VQ there seems to be higher redundancy among quantization vectors. The diagonal of the matrix represents the entropy of each quantization vector. The MI estimate on the quantization vector shows that the redundancies are significantly higher in the learned representation for VQ.

5.3. Ablation study

We study the effect of K and D on the model performance. The model is more sensitive to the dimensionality D of the sub-vector and less sensitive to K. DQ outperforms coupled variants on likelihood estimation and reconstruction loss in all settings. C_R grows exponentially with M as opposed to K. Fewer code vectors can be used to quantize without performance degradation. For example, when $M = 3$, DQ outperforms a VQ variant by 35% and uses 25% fewer code vectors. Fig. 5 shows a summary of the loss for different K and M values. A detailed table of the results can be found in the Appendix.

5.4. Likelihood Estimation

For likelihood estimation, we compare DQ-AE with other likelihood estimator models and report the numbers from their work. We use Very Deep VAE ("VD-VAE") [7] as a continuous AutoEncoder baseline and Sparse Transformer ("S-Tr") [8] as an Auto-Regressive baseline.

For experiments on ImageNet, we add a number on the image resolution at which we train the model at the end of the dataset name. For our model, we use an identical architecture and number of hierarchies for all resolution of the dataset. The detailed results are in Tab. 1. We outperform all previous state-of-the-art models when measuring the loss in bits/dim, we also report C_R separately. The estimate for $C_R \sim 0.2$ nats. Visual inspection of both top and bottom hierarchies confirm that they encode different granularity of features and are utilized (Fig. 4), and perceptual quality is improved (Fig. 1). Additional high resolution images are attached in the supplementary materials.

When compared to the hierarchical model by Razavi *et al.* [38], DQ-AE also outperforms in reconstruction error for L_2 on ImageNet-256. On CIFAR-10, the DQ-AE loss is 0.019 compared to 0.044 for VQ-VAE. For ImageNet-256, DQ-AE loss is 0.0032 compared to 0.005 for VQ-VAE-2.

6. Discussion

We thoroughly evaluate the theoretical claims of our work and empirically verify our method in likelihood estimation. Evaluation of the discrete representation on a downstream task such as latent interpolations is domain specific. Sampling from the multi-resolution and high dimensional discrete codebooks requires training additional models post-hoc. As such, there are multiple open problems in how to design such a model. We leave this for future work.

The direct evaluation and comparison on NLL between explicit likelihood models can be non-equivalent. Our model makes different assumptions on the prior distribution and as such the direct comparison can be flawed. Previous work [1] has suggested that the ELBO might be a poor metric to evaluate deep latent variable models. We mitigated the issue and followed the theoretical result and experimental methodology to previous work [7]. We consider the proper evaluation of the discrete prior with other model variants as an open problem.

7. Conclusion

We analyze the effects of decomposing an image feature tensor along an axis of statistical independence. Decomposition and quantization among independent features outperforms coupled feature variants. Our theoretical insights focus on feature decoupling for decomposed image feature tensors along the channel axis. Our results corroborate previous analyses and explain the advantage of NiN applications which can be interpreted as an information bottleneck.

Based on our theoretical insight, we propose Depthwise Quantization (DQ) that provides significantly more efficient bottleneck capacity by eliminating redundancies implicitly in the feature axis. DQ is trained end-to-end with a Hierarchical Auto-Encoder (DQ-AE) and learns improved hierarchical discrete representations. Our method is domain agnostic, and we consider the evaluation on a specific task for future work.

References

[1] Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In *International Conference on Machine Learning*, pages 159–168. PMLR, 2018. 8

[2] Artem Babenko and Victor Lempitsky. Additive quantization for extreme vector compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 931–938, 2014. 3

[3] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018. 3, 5

[4] Yue Cao, Mingsheng Long, Jianmin Wang, and Shichen Liu. Deep visual-semantic quantization for efficient image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1328–1337, 2017. 1

[5] Olivier Cappé and Eric Moulines. On-line expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, Jun 2009. 6

[6] Jianfei Chen, Jun Zhu, Yee Whye Teh, and Tong Zhang. Stochastic expectation maximization with variance reduction. In *NeurIPS*, pages 7978–7988, 2018. 6

[7] Rewon Child. Very deep {vae}s generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*, 2021. 7, 8

[8] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 7, 8

[9] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 3

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[11] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020. 1, 3

[12] Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Estimating mutual information for discrete-continuous mixtures. *arXiv preprint arXiv:1709.06212*, 2017. 6

[13] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):744–755, 2013. 3

[14] Anirudh Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. Z-forcing: Training stochastic recurrent networks. *arXiv preprint arXiv:1711.05411*, 2017. 2

[15] Jianbo Guo, Yuxi Li, Weiyao Lin, Yurong Chen, and Jianguo Li. Network decoupling: From regular to depthwise separable convolutions. *arXiv preprint arXiv:1808.05517*, 2018. 3

[16] Daniel Haase and Manuel Amthor. Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14600–14609, 2020. 3, 6, 7

[17] Daniel Haase and Manuel Amthor. Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14600–14609, 2020. 6

[18] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016. 3, 5

[19] Marcus Hutter. Distribution of mutual information. *Advances in neural information processing systems*, 1:399–406, 2002. 4

[20] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014. 3

[21] Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. In *International Conference on Machine Learning*, pages 2390–2399. PMLR, 2018. 3, 6

[22] Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*, 2019. 3

[23] Zeger F Knops, JB Antoine Maintz, Max A Viergever, and Josien PW Pluim. Normalized mutual information-based registration using k-means clustering-based histogram binning. In *Medical Imaging 2003: Image Processing*, volume 5032, pages 1072–1080. International Society for Optics and Photonics, 2003. 6

[24] Zeger F Knops, JB Antoine Maintz, Max A Viergever, and Josien PW Pluim. Normalized mutual information based registration using k-means clustering and shading correction. *Medical image analysis*, 10(3):432–439, 2006. 6

[25] Kazuma Kobayashi, Ryuichiro Hataya, Yusuke Kurose, Mototaka Miyake, Masamichi Takahashi, Akiko Nakagawa, Tatsuya Harada, and Ryuji Hamamoto. Decomposing normal and abnormal features of medical images into discrete latent codes for content-based image retrieval. *arXiv preprint arXiv:2103.12328*, 2021. 3

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 8

[27] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 3

[28] Zhizhe Liu, Xingxing Zhang, Zhenfeng Zhu, Shuai Zheng, Yao Zhao, and Jian Cheng. Convolutional prototype learning for zero-shot recognition. *Image and Vision Computing*, 98:103924, 2020. 1

[29] Xiaotong Lu, Heng Wang, Weisheng Dong, Fangfang Wu, Zhonglong Zheng, and Guangming Shi. Learning a deep vector quantization network for image compression. *IEEE Access*, 7:118815–118825, 2019. 1

[30] David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003. 4

[31] Franck Mamalet and Christophe Garcia. Simplifying convnets for fast learning. In *International Conference on Artificial Neural Networks*, pages 58–65. Springer, 2012. 3

[32] Young-Il Moon, Balaji Rajagopalan, and Upmanu Lall. Estimation of mutual information using kernel density estimators. *Physical Review E*, 52(3):2318, 1995. 6

[33] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017. 1, 2, 8

[34] Tom Le Paine, Pooya Khorrami, Shiyu Chang, Yang Zhang, Prajit Ramachandran, Mark A Hasegawa-Johnson, and Thomas S Huang. Fast wavenet generation algorithm. *arXiv preprint arXiv:1611.09482*, 2016. 2

[35] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003. 6

[36] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018. 7

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation package - torch.autograd, 2017. 5

[38] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019. 1, 3, 8

[39] Tom Richardson and Ruediger Urbanke. *Modern coding theory*. Cambridge university press, 2008. 4

[40] Aurko Roy, Ashish Vaswani, Arvind Neelakantan, and Niki Parmar. Theory and experiments on vector quantized autoencoders. *arXiv preprint arXiv:1805.11063*, 2018. 6

[41] Aurko Roy, Ashish Vaswani, Niki Parmar, and Arvind Neelakantan. Towards a better understanding of vector quantized autoencoders. 2018. 6

[42] Pierre Stock, Armand Joulin, Rémi Gribonval, Benjamin Graham, and Hervé Jégou. And the bit goes down: Revisiting the quantization of neural networks. *arXiv preprint arXiv:1907.05686*, 2019. 3

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 3

[44] Hanwei Wu and Markus Flierl. Variational information bottleneck on vector quantized autoencoders. *arXiv preprint arXiv:1808.01048*, 2018. 6

[45] Hanwei Wu and Markus Flierl. Learning product codebooks using vector-quantized autoencoders for image retrieval. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5. IEEE, 2019. 3

[46] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3474–3482, 2018. 1