FAKULTÄT FÜR
INFORMATIK

Otto-von-Guericke-University Magdeburg

Faculty of Computer Science

Department of Human Centered Artificial Intelligence

Analyzing the differences in efficiency and performance when combining Matryoshka Representations with Quantization

Bachelor Thesis

Author:

Kai Ponel

Examiner and Supervisor:

Prof. Ernesto William De Luca

2nd Examiner:

Prof. Marco Polignano

Magdeburg, 21.08.2024

Contents

Acknowledgements

This bachelor thesis concludes yet another chapter in my life, perhaps the most challenging but also the best one so far. I would like to express my deepest gratitude to everyone who made the last four years at Otto-von-Guericke University Magdeburg so special. I want to thank my mother and father for always supporting me throughout the years. I would also like to thank my friends for countless hours of fun and laughter, and for always being there for me. Furthermore, I'm grateful for any professor or coordinator who felt confident enough in my abilities to teach various courses over the years. In addition to this, I would also like to thank the Otto-von-Guericke University and IWD Market Research GmbH for supporting me with scholarships in the past two years. This thesis also wouldn't be possible without the support of IBM. In this regard, I would like to especially thank Aylin Julia Temizkan and Marinela Bilic-Nosic for their trust and support.

Lastly, as perhaps the most important persons in regard to this thesis, I would like to thank my supervisors Professor Ernesto William De Luca and Professor Marco Polignano for their guidance and support. This thesis wouldn't be possible without the countless iterations of feedback, especially provided by Professor Polignano in the past months.

Abstract

Embeddings are a crucial component of many natural language processing (NLP) models, which have seen a surge in popularity in recent years. As methods that rely heavily on embeddings, such as Retrieval Augmented Generation (RAG) pipelines, become more common, the costs associated with learned representations are becoming increasingly significant. Both 1D and 2D Matryoshka representations as well as quantization have been proposed as methods to address the computational costs of embeddings. This thesis investigates the trade-offs between memory consumption, inference time, and performance when combining quantization with Matryoshka representations. The results show that inference time can be more than halved, with memory usage decreased by a factor of 32, while only incurring an 18.1% loss in relative accuracy.

Abbreviations and Notation

Abbreviations

dim	Dimensionality
emb	Embedding
LLM	Large Language Model
MR	Matryoshka representation
1D MR	1 Dimensional (Traditional) Matryoshka representation
2D MR	2 Dimensional (Extended) Matryoshka representation
quant	Quantization
bin	Binary
int8	8-Bit Integer
fp32, **float32**	32-Bit Floating-Point
CO_2	Carbon Dioxide

Notations

d	**Dimensionality**	Size of the embedding vector
m	**Matryoshka size**	Number of dimensions used from the embedding vector
l	**Matryoshka layer**	Layer used for inference to obtain the embedding vector
q	**Quantization technique**	Quantization technique applied to embeddings

1 Introduction

Deep Learning systems have seen a tremendous increase in popularity and usage in recent years. Thanks to recent advancements in the field, AI systems now play a key role in many aspects of our daily lives [BEE+24]. In these AI systems, learned representations, often referred to as "embeddings", are a way to represent data of any kind (e.g. words, images, or documents) in a vector consisting of real numbers. The goal of embeddings is to represent objects in a way that brings similar items closer together in the embedding space while pushing dissimilar items further apart. These representations can then be used for a variety of downstream tasks, such as classification, clustering, or retrieval. [Ben12, GBCB16]

1.1 History of Embeddings

Advancements in neural networks, which are a class of machine learning models initially inspired by the human brain [GBCB16], first introduced the concept of learning embeddings from data during the 1980s. Prominent to this day is the concept of an AutoEncoder, which is a neural network that learns to encode data into a lower-dimensional space and then decode it back to its original form [RHW86, Bal87]. The network, which consists of an encoder and a decoder, is trained to minimize the reconstruction error. Since the space between the encoder and the decoder is very limited, the network is being forced to learn a rich representation of the data, as illustrated by Figure 1.1.

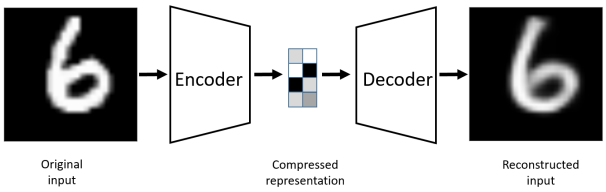


Figure 1.1: An illustration of the AutoEncoder architecture. The encoder learns to compress the input data into a lower-dimensional space while the decoder tries to rebuild the high-dimensional data from the lower-dimensional representation. Figure excerpted from [BKG21].

Another significant advancement in the field of embeddings was the introduction of Word2Vec in 2013 [MCCD13]. Word2Vec is a model that learns to represent words as dense vectors. It is trained on a large corpus of text data and tries to learn the meaning of a word based on its surrounding words. A year later, in 2014, this was followed by GloVe [PSM14], which in contrast to Word2Vec, learns embeddings based on the global co-occurrence statistics of words in a corpus. The resulting vectors capture semantic and

syntactic information about the words, which can be demonstrated by applying arithmetic operations on the vectors, such as $Paris - France + Italy$ to which the closest word-vector is *Rome* [MCCD13].

Following the introduction of the original Transformer model in 2017 [VSP+23], the field of embeddings saw another significant advancement with the introduction of BERT in 2018 [DCLT18]. "**B**idirectional **E**ncoder **R**epresentations from **T**ransformers" marked a shift towards context-dependent embeddings. In context-dependent embeddings, the meaning of a word could change based on the other words in a sentence.

Since then, there have been many more advancements in the field of text-embeddings, such as the introduction of Sentence-BERT [RG19], which learns embeddings for entire sentences, or the introduction of SimCSE [GYC21], which maximizes the similarity between augmented versions of the same sentence to learn embeddings. Besides natural language, embeddings are also used in other domains such as computer vision, where models like CLIP [RKH+21] learn embeddings for both images and text to enable cross-modal retrieval.

1.2 Use Cases of Embeddings

Embeddings can be utilized for a number of machine-learning tasks. Some of the most common applications include:

- **Classification**: By training a classifier on top of the embeddings, the model can learn to predict the class of a given input. [ARW+15]
- **Clustering**: By measuring the similarity between embeddings, items that are close together in the embedding space can be put into the same group. [JZT+17]
- **Retrieval**: Retrieve the most relevant items from a large set based on the similarity between their embeddings and a query. [GXG+24]

One use case where retrieval is applied is **Retrieval Augmented Generation (RAG)** [LPP+21] which became increasingly popular since the release of the initial paper in 2021. The main idea of RAG is to augment generative Large Language Models (LLMs) such as GPT-4 with non-parametric memory (information from documents embedded in vectors) to improve the quality of the generated text and to avoid hallucinations. This way it is possible for LLMs to generate accurate answers for highly specific questions without having to retrain the model on the new data. [GXG+24]

A naive RAG implementation, based on [GXG+24] and illustrated in Figure 1.2, consists of the following steps:

1. **Document Embedding:** Documents are split into chunks, encoded into vectors, and stored in a vector database.
2. **Retrieval:** The top k chunks most relevant to the query are retrieved based on semantic similarity.
3. **Generation:** The original query and the retrieved chunks are then input into a Large Language Model (LLM) to generate the final answer.

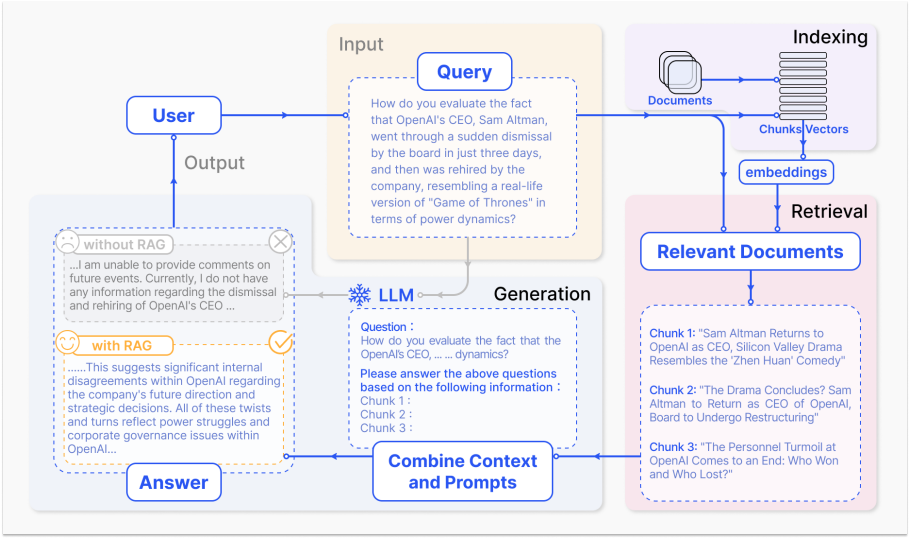


Figure 1.2: Illustration of a naive RAG implementation. Documents are split into chunks, encoded into vectors, and stored in a vector database. The top k chunks most relevant to the question are retrieved based on semantic similarity. The original question and the retrieved chunks are then input into a Large Language Model (LLM) to generate the final answer. Figure excerpted from [GXG+24].

Retrieval Augmented Generation is a prime example of a workflow that couldn't be possible without embeddings. Given that there are potentially millions of documents that need to be embedded and compared to a query, the computational costs of embeddings can become a bottleneck.

1.3 Costs of Embeddings

In order to use embeddings on a large scale, it is crucial to consider the computational costs associated with them. Learned Representations cause costs in the following ways:

- **Inference**: Inference describes the process of using a trained model to make predictions on new data. In the case of learned representations, this means computing the embeddings for the input data. However, this has to be done only once per element since the calculated embeddings can be stored. [HZRS16]
- **Downstream Tasks**: After computing the embeddings, they are often used for downstream tasks. As an example, consider a naive Retrieval Augmented Generation pipeline, introduced in the previous section. Every document only needs to be embedded once. However, during the lifetime of the system, this embedding might be compared to many others hundreds if not thousands of times. Each time such a comparison takes place, the embeddings have to be loaded into memory and the similarity between them needs to be calculated. [KS21, Var19]

1.4 Main Contribution and Research Question

Considering the immense importance that embeddings hold for various tasks, paired with the computational costs associated with them, it is crucial to find ways to improve the efficiency of learned representations while maintaining high performance at whatever task may be at hand. This has been acknowledged by the research community and has led to the development of various methods to reduce the computational costs of embeddings. Some of the recent approaches to address this issue are:

- **1D Matryoshka representations**: Inspired by the Matryoshka dolls, the idea of Matryoshka representations is to store embeddings in a nested manner, where each level of the hierarchy stores a more compressed version. This way, the desired representation size can be chosen during inference, enabling a high degree of flexibility. The authors of the original paper have shown that, for example, halving the embedding size only results in a small decrease in accuracy. [KBR+22]
- **2D Matryoshka representations**: Building on the idea of 1D Matryoshka representations, 2D Matryoshka representations extend the concept to allow even more flexibility. In addition to the properties of the previous approach, 2D Matryoshka representations also allow for the layer from which the embeddings are taken to be chosen during inference without sacrificing much accuracy. [LLL+24]
- **Quantized embeddings**: Quantization is a well-known technique in machine learning, where the goal is to reduce the number of bits used to represent a value while maintaining a high degree of accuracy. For instance, when quantizing embeddings, which are normally stored as 32-bit floating-point numbers, to 8-bit integers, the memory footprint is reduced by a factor of 4. [NFA+21, Doc24]

While there has been some work outside the scientific community on combining these approaches, there has not been any academic research on this topic covering the performance (i.e. the quality of embeddings) and efficiency (i.e. how many resources they require) in various scenarios. Thus, the goal of this thesis is to combine Matryoshka representations with quantization to fill this gap by answering the following research question:

In the context of Large Language Models, can Matryoshka representations and quantization be combined to improve the computational efficiency of embeddings while maintaining high performance at downstream tasks?

1.5 Thesis Outline

Chapter 2 will discuss the previous work in the field of efficient representation learning, specifically focusing on Matryoshka representations and quantized embeddings. In Chapter 3 the methodology of this work will be introduced and the approach to combine these techniques will be discussed. Following this, in Chapter 4 the experimental setup will be described, followed by a presentation of the findings. Finally, Chapter 5 will contain a summary of the findings alongside an interpretation and ideas for future work.

2 Related Work

This section provides an overview of the related work, explaining the techniques that will be combined. First, in Section 2.1, the concept of 1D Matryoshka representations is introduced, a technique enabling flexible embedding size during inference. Building on that, the concept of 2D Matryoshka representations is explained in Section 2.2, which allows for flexible embedding size and layer selection during inference. Finally, Section 2.3 introduces the concept of quantization.

2.1 1D Matryoshka Representations

This section is based on the paper "Matryoshka Representation Learning" by Aditya et al. [KBR+22].

Figure 2.1: Matryoshka dolls, a set of wooden dolls of decreasing size placed one inside another. Excerpted from cambridgemaths.org

.

In the context of machine learning, an embedding z is a representation of data in a d-dimensional, real-valued space:

$$z \in \mathbb{R}^d$$

Traditional embeddings do not allow for flexibility when it comes to the dimensionality of the representation. After training a model to convert data into a d-dimensional vector, the dimensionality of this vector is fixed and all d dimensions have to be used in downstream tasks to avoid huge drops in performance. In the past, there have been attempts to address this issue, however, each one comes with its drawbacks:

- **Principle Component Analysis** [WEG87]: Fast but performance drops are too high on non-linear data.
- **Jointly-Optimized Sub Networks** [CGW+19, YYX+18]: Computationally expensive and complex training.

1D Matryoshka representations resemble a new approach to add flexibility, such that the dimensionality of embeddings can be adjusted on the fly. Inspired by the Russian Matryoshka dolls (exemplified in Figure 2.1), which are a set of wooden dolls of decreasing size placed one inside another, the authors suggest letting the model learn embeddings of different sizes within the same high-dimensional vector. These "coarse-to-fine" representations are as accurate as their independently trained counterparts while being more efficient in terms of computation and storage. At inference time, the user can choose the most suitable embedding size for the task at hand with no additional cost.

2.1.1 Formal Description

Let $d \in \mathbb{N}$. Furthermore, the authors define $z \in \mathbb{R}^d$ to be the high-dimensional embedding of an element. For every $m \in \mathcal{M}$, MRL (in the following also referred to as "1D Matryoshka representation" or "1D MR") enables the first m dimensions of the embedding vector $z_{1:m} \in \mathbb{R}^m$ to be used as a general purpose representation of the input data x. The authors state that the typical set $\mathcal{M}$ is composed of consistent reductions by half of the embedding size, starting from the full capacity embedding size d until a low information threshold is reached. As an example, for $d = 2048$ the authors chose $\mathcal{M} = \{2048, 1024, 512, 256, 128, 64, 32, 16, 8\}$ as nesting dimensions.

The authors then propose a loss function to train the model to learn these nested embeddings. For this, they suppose a labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ is given, where x_i is the input and y_i is the label. The authors define the loss function as:

$$\min_{\{\mathbf{W}^{(m)}\}_{m\in\mathcal{M}},\ \theta_F} \frac{1}{N} \sum_{i\in[N]} \sum_{m\in\mathcal{M}} c_m \cdot \mathcal{L}\left(\mathbf{W}^{(m)} \cdot F(x_i;\theta_F)_{1:m}\ ;\ y_i\right)\ , \tag{2.1}$$

- $F(x_i;\theta_F)$: Deep Neural Network with parameters θ_F which maps the input x_i to the high-dimensional embedding space
- L: Number of labels
- $\mathcal{L}$: Multi-class softmax cross-entropy loss function
- c_m: Relative importance score for each embedding size (The authors set $c_m = 1, \forall m \in \mathcal{M}$)
- $\mathbf{W}^{(m)} \in \mathbb{R}^{L\times m}$: Weights of the separate linear classifier for the embedding size m
- N: Number of samples in the dataset

Essentially, this loss function calculates the weighted average of the losses for each embedding size in $\mathcal{M}$. The researchers add that this can be made more efficient by weight-tying the linear classifiers for different embedding sizes, such that:

$$\mathbf{W}^{(m)} = \mathbf{W}_{1:m} \tag{2.2}$$

for a set of shared weights $\mathbf{W} \in \mathbb{R}^{L \times d}$. In the following, this is referred to as *Efficient* Matryoshka Representation Learning (**MRL-E**).

2.1.2 Experiments and Results

The authors compare their MRL and MRL-E approach to the following baselines:

- **FF**: Independently trained (fixed-feature) representations
- **SVD**: Dimensionality reduction using Singular Value Decomposition [KL80]
- **Slim. Net**: Sub-net method (Slimmable Networks [YYX+18])
- **Rand. FS**: Randomly selected features of the highest capacity FF model

Classification Results

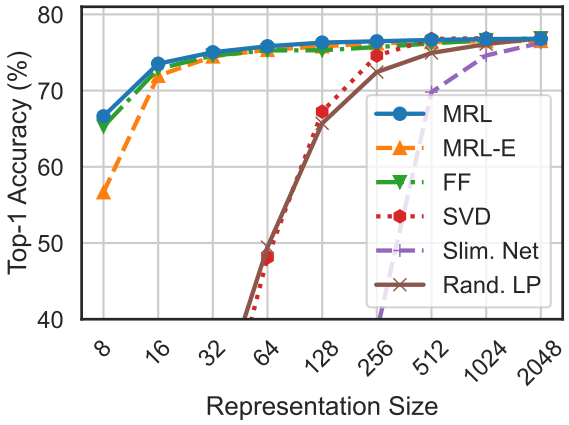


Figure 2.2: ImageNet-1K linear classification accuracy of ResNet50 models. MRL is as accurate as the independently trained FF models for every representation size. Figure and caption excerpted from the original paper [KBR+22].

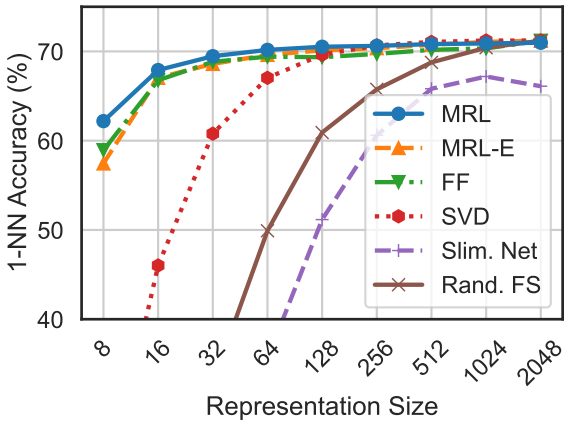


Figure 2.3: ImageNet-1K 1-NN accuracy of ResNet50 models measuring the representation quality for downstream task. MRL outperforms all the baselines across all representation sizes. Figure and caption excerpted from the original paper [KBR+22].

As can be seen in Figure 2.2, the MRL model has an equal or higher accuracy compared to each Fixed Feature (FF) model for all embedding sizes $\mathcal{M}$. From 16 dimensions onwards, the MRL-E model is within 1% of the FF model. Similar results can be found when comparing the 1 Nearest Neighbor (1-NN) accuracy, another metric used for evaluating the performance of a classifier, shown in Figure 2.3. The 1D MR model achieves the same accuracy as the individually trained FF models for high dimensionalities while even being up to 2% more accurate on lower dimensions.

The authors also investigated how well 1D Matryoshka representations perform on embedding sizes which were not elements of $\mathcal{M}$. Figure 2.4 shows, that even though

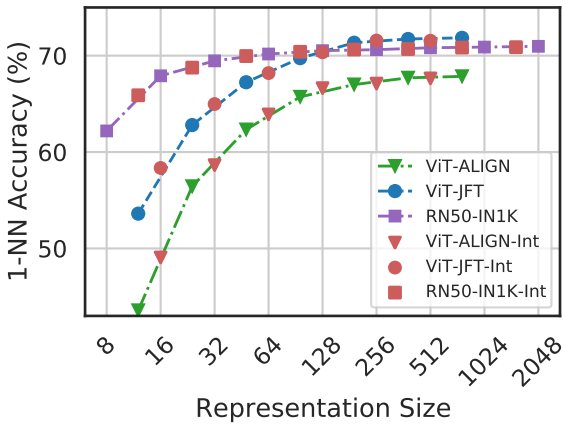


Figure 2.4: Despite optimizing MRL only for $O(\log(d))$ dimensions for ResNet50 and ViT-B/16 models; the accuracy in the intermediate dimensions shows interpolating behavior. Figure and caption excerpted from the original paper [KBR+22].

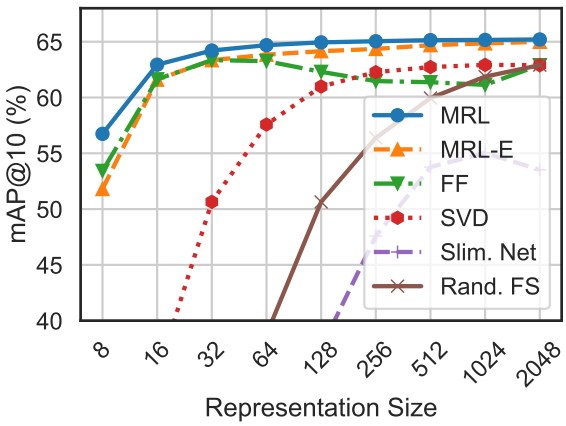


Figure 2.5: mAP@10 for Image Retrieval on ImageNet-1K with ResNet50. MRL consistently produces better retrieval performance over the baselines across all the representation sizes. Figure and caption excerpted from the original paper [KBR+22].

MRL has only been trained on $O(log(d))$ dimensions, the accuracy across the intermediate dimensions shows **interpolating behavior**. The authors state that these findings allow for maximum flexibility in choosing the right embedding size for the task at hand.

Retrieval Results

For retrieval tasks, the authors discovered similar findings. Figure 2.5 shows that MRL consistently produces better retrieval performance over the baselines across all the representation sizes. MRL-E matches the performance of the FF model for $d < 128$ and exceeds it for $d \geq 128$.

2.2 2D Matryoshka Representations

This section is based on the paper "2D Matryoshka Sentence Embeddings" by Li et al. [LLL+24].

While 1D Matryoshka representations greatly improve the efficiency of learned representations for downstream tasks, the costs related to inference remain unchanged. Regardless of the number of dimensions used in the output vector, the forward pass required to compute these embeddings is still expensive since the model has to process the entire input sequence through all layers before the embeddings are output. Adding to the flexibility of 1D MR, 2DMSE (in the following also referred to as "2D Matryoshka representation"

or "2D MR") allows the user to choose the number of layers **and** the size of the output embeddings at inference time, without any additional costs.

2.2.1 Formal Description

The authors of the original paper build their work on top of a pre-trained language model to transform text into dense sentence embeddings. In specific, they use $BERT_{base}$ [DCLT19] paired with CLS pooling following previous work [GYC21, LL23]:

$$x_m^n = \text{BERT}_{1:n}^{cls}(x)_{1:m} \in \mathbb{R}^m \tag{2.3}$$

Here, $m \in [1, d]$ represents the first m dimensions in the d-dimensional embedding vector, as has been shown in Section 2.1. In addition to this, the authors added a new parameter $n \in [1, N]$ which denotes the n-th layer of the underlying transformer model containing N layers. Given this, the user of the model now has the flexibility to choose:

- the number of layers and
- the size of the output embeddings.

A conceptual comparison between the original embeddings, 1D Matryoshka embeddings, and 2D Matryoshka embeddings is shown in Figure 2.6.

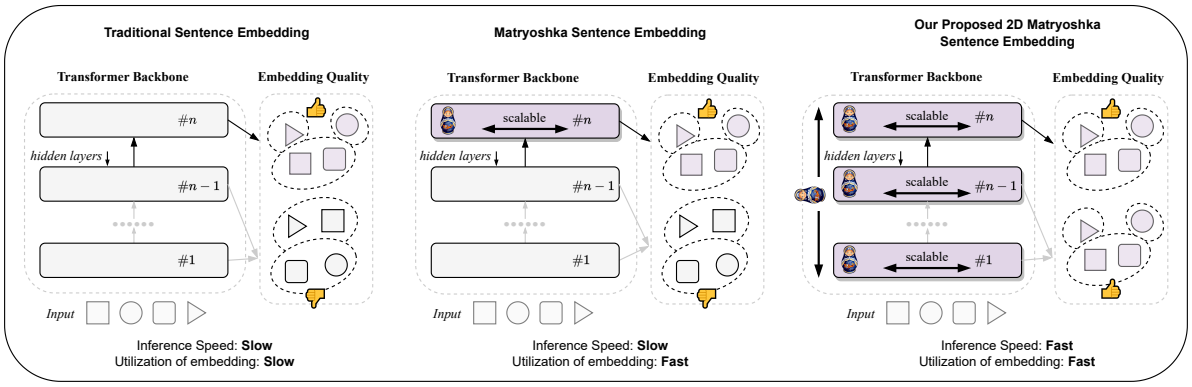


Figure 2.6: A visual comparison of various sentence embedding methods. The gray blocks represent Transformer layers fine-tuned with AnglE, which are not optimized for Matryoshka representation. The purple block represents Transformer layers fine-tuned with AnglE together with matryoshka loss. Figure and caption excerpted from the original paper [LLL+24].

Training

The training procedure initially follows a conventional approach for representation learning by training full-capacity embeddings from the last attention layer [RG19, GYC21,

LL23] x_N^D. This results in the objective:

$$\mathcal{L}_N^d = \text{loss}(x_N^d; A) \tag{2.4}$$

where A is the auxiliary information used for computing the loss. Any loss function which is suitable for representation learning can be used.
In the same training step, a random layer $n \sim \mathcal{U}(1, N-1)$ is selected uniformly and its full embedding vector is also used for representation learning, just like above:

$$\mathcal{L}_n^d = \text{loss}(x_n^d; A) \tag{2.5}$$

Now a random Matryoshka size m is also being sampled uniformly to train nested low-dimensional vectors at both the last layer N and the previously randomly selected layer n. This results in two additional loss terms:

$$\mathcal{L}_N^m = \text{loss}(x_N^m; A) \tag{2.6}$$

$$\mathcal{L}_n^m = \text{loss}(x_n^m; A) \tag{2.7}$$

Next, the authors of the original paper proposed embedding alignment. They reason that based on the scaling laws for neural language models [KM+20] more transformer layers imply better language understanding capabilities. Thus, sampled intermediate layers are alligned to the last layer for both the full d-dimensional embeddings as well as for the lower m-dimensional vectors. This is done by minimizing the KL-Divergence and introduces a fifth loss term:

$$\mathcal{L}_{align} = \text{KLD}(\mathcal{L}_n^d, \mathcal{L}_N^d) + \text{KLD}(\mathcal{L}_n^m, \mathcal{L}_N^m) \tag{2.8}$$

Finally, all training objectives are combined to form one final objective to optimize:

$$\mathcal{L} = \sum_L^S \lambda_L L \tag{2.9}$$

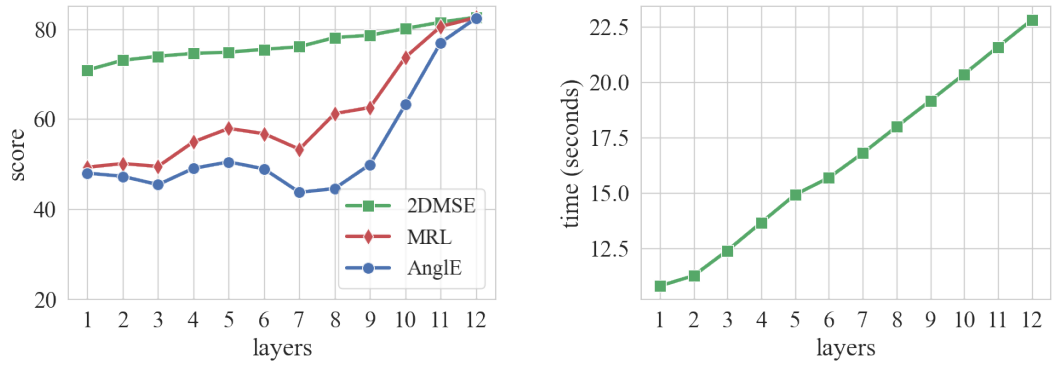
with $S = \{\mathcal{L}_N^d, \mathcal{L}_n^d, \mathcal{L}_N^m, \mathcal{L}_n^m, \mathcal{L}_{align}\}$ and hyperparameters λ_L which represent weights for each individual objective.

2.2.2 Experiments and Results

The authors of the original paper compare the performance of the proposed 2D Matryoshka representation model against the following baselines at each intermediate layer:

- **AnglE**: A model that was trained using the AnglE loss function. The AnglE loss only considers the entire embedding vector of the last layer [LL23].
- **MRL**: A model that was trained using the 1D Matryoshka representation loss function (refer to Equation 2.1) [KBR+22].

Based on the results illustrated in Figure 2.7a (for $d = 768$), the authors conclude that 2D MRs improve the quality of embeddings in intermediate layers when compared to MRL and AnglE. While the performance of all models is similar for the last layer, both MRL and AnglE perform worse on intermediate layers. Adding to this, the authors found that



(a) Score on STS *vs* number of layers. **(b)** Inference time *vs* number of layers.

Figure 2.7: Subfigure (a) displays the average Spearman's correlation scores of different layers. Both (a) and (b) use an embedding size of 768 and the standard STS benchmark dataset. Subfigure (b) illustrates the time taken to use embeddings from different layers to encode the entire STS benchmarks. Figure and caption excerpted from the original paper [LLL+24].

the performance of MRL and AnglE fluctuates as the number of layers increases while on the other hand, 2DMSE's performance improves consistently as the layers deepen. As an example, the 2D MR model reaches a score of 70.09 after the first layer while AnglE requires 11 layers to reach a score above 70. The authors also investigated the performance per layer for different embedding sizes and found similar results, as can be seen in Figure 8 in the appendix.

Furthermore, the time it took to do inference at each intermediate layer was measured. As Figure 2.7b shows, the inference time increases linearly with the number of layers used. While achieving an approximate real-world speed-up of 1.46x when using the middle layer ($n = 6$) compared to the last layer ($n = 12$), the authors found that the performance of 2D MRs dropped by 7.15 points on the STS benchmark. In the same scenario, MRL's performance was reduced by 25.79 points, while AnglE's performance dropped by 33.44 points compared to the performance of the last layer.

2.3 Quantization

Quantization originates from the field of signal processing and describes the conversion of a continuous spectrum into a discrete form [GN98]. Within the context of machine learning, quantization describes the concept of reducing the precision of numerical values. This concept is often applied to the parameters of a model to reduce the memory footprint and make the model more efficient, allowing it to be deployed on devices with limited memory and computational resources [NFA+21]. Similar to model parameters, the go-to data type for representing embeddings is a floating-point number, usually with a precision of 32 bits (fp32) [IEE08]. Given that the embedding vector consists of d dimensions, the total number of bits required to represent the embedding vector is $d \cdot 32$. As an example, with $d = 1024$ dimensions, the total number of bits required to represent the embedding vector is $1024 \cdot 32 = 32768$ bits which is equal to 4KiB.

3 Methodology

The main contribution of this thesis is to analyze the trade-off between efficiency and performance when both 1D and 2D Matryoshka representations are quantized. For this, benchmarks will be used to evaluate the performance, inference speed, and memory consumption of different Matryoshka representations with varying levels of quantization.

Section 3.1 outlines the core concept of this thesis, while Section 3.2 details the evaluation metrics used. The quantization techniques applied in the experiments are covered in Section 3.3. Section 3.4 introduces the benchmarks used for the experiments before Section 3.5 concludes this chapter by describing the models used in the experiments.

3.1 Principal Idea

To investigate the trade-off between efficiency and performance when combining Matryoshka representations with quantization, each combination of the two approaches will be tested for each model. The overall idea is illustrated by Algorithm 1.

Algorithm 1 Conceptual overview of the workflow

```
for each Model f ∈ F do
    for each Matryoshka representation m ∈ M(f) do
        for each Quantization Technique q ∈ Q do
            for each inference layer l ∈ L(f) do
                MTEB(f[: l], m, q)
            end for
        end for
    end for
end for
analyze(F, M, Q, L)
```

Note that for traditional 1D Matryoshka representations, there is only one inference layer which is the last layer of the model. However, for 2D Matryoshka representations, there are multiple inference layers that serve as the last layer of the respective submodel. This results in the total computational complexity of:

$$\mathcal{O}(|T| \times |M| \times |Q| \times (|F_{1D}| + |F_{2D}| \times |L|)) \tag{3.1}$$

with:

- $|T|$: Number of tasks in the benchmark

- $|M|$: Number of Matryoshka representations
- $|Q|$: Number of quantization techniques
- $|F_{1D}|$: Number of models supporting 1D Matryoshka representations
- $|F_{2D}|$: Number of models supporting 2D Matryoshka representations
- $|L|$: Number of inference layers for 2D Matryoshka representations

3.2 Evaluation Metrics

For each combination of model f, Matryoshka representation m, quantization technique q and (if applicable) sampled layer l, the following metrics are collected and analyzed:

- **Performance**: The effectiveness of the embedding in a downstream task. The metrics used to measure the performance depend on the specific type of task and are provided in Section 3.4.
- **Memory Usage**: The theoretical memory usage of the embedding, calculated based on the number of bits needed to represent the embedding vector. This is calculated as the product of the embedding vector's dimensions and the number of bits allocated per value.

$$\text{Memory} = \text{Dimensionality} \times \text{Bits per value}$$

- **Evaluation Time**: The time required to run the downstream task, measured in seconds.

While these metrics individually provide valuable insights, a detailed analysis of the trade-offs between performance, memory usage and evaluation time will also be conducted. Additionally, since the focus of this study is not on assessing the absolute performance or evaluation time of the models, but rather on comparing the different quantization techniques and Matryoshka representations, the following normalizations will be applied:

- **Relative Performance**: The absolute performance of a specific configuration on a certain task will be normalized by the performance of the model's default configuration on the same task.
- **Relative Evaluation Time**: The absolute evaluation time of a specific configuration on a certain task will be normalized by the evaluation time of the model's default configuration on the same task.

The default configuration refers to the model's configuration with the highest dimensional embeddings, no quantization, and the last layer of the model used as the inference layer.

For models that support 2D Matryoshka representations, the same metrics as for 1D Matryoshka representations can be collected. Since 2D MR models cannot only reduce the

cost of using embeddings in downstream tasks but also have an impact on the generation of the embeddings, the analysis of the task time should be paid special attention to.

3.3 Quantization Techniques

As discussed in Section 2.3, quantization is the process of converting some data type to a lower precision data type to save memory and computation. Besides the default floating-point 32 (fp32) data type, the following quantization techniques will be considered:

8-Bit Integers

An 8-bit integer (int8) is a data type that uses eight bits to represent an integer number which can be signed or unsigned. In both cases there are 256 possible values that can be represented, ranging from -128 to 127 for signed integers and from 0 to 255 for unsigned integers. The conversion from floating-points to int8 is done using bucketing, which splits the continuous range of floating-point numbers into 256 equally large buckets, where each bucket is represented by a single int8 number. In the following, int8 refers to signed 8-bit integers. [Doc24]

Binary numbers

Binary numbers are the building blocks of any other datatype and can only represent either a 0 or a 1. Conversion from floating-point to binary numbers is done by setting the binary number to 1 if the floating-point number is positive and to 0 if the floating-point number is negative:

$$\text{Binary}(x_{fp}) = \begin{cases} 0 & \text{if } x_{fp} \leq 0 \\ 1 & \text{if } x_{fp} > 0 \end{cases}$$

[Doc24]

3.4 Benchmarks

The Massive Text Embedding Benchmark (MTEB) includes 58 benchmarks across 8 different tasks, making it the largest and most comprehensive benchmark for text embeddings to date [MTMR23]. A leaderboard for the MTEB has been established on Hugging Face, which can be accessed at https://huggingface.co/competitions/mteb/leaderboard. Hugging Face is a US-based company that has set its mission to "democratize good machine learning" [Fac24]. Given its popularity and status as the largest benchmark for text embeddings, the MTEB was selected for the experiments in this thesis. An overview of the MTEB is shown in Figure 3.1. In the following, the types of tasks and datasets are briefly explained based on the information provided in the original paper. Note that the

Bitext Mining task category is not used for evaluating embeddings on the Hugging Face leaderboard and is therefore not considered in the following.

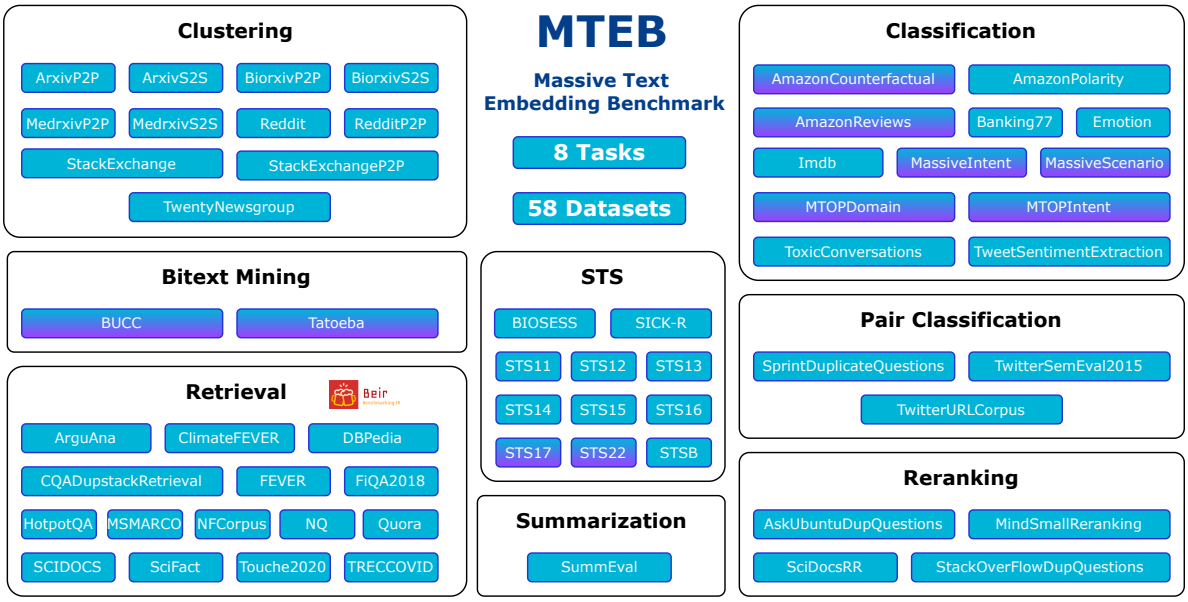


Figure 3.1: Overview of the Massive Text Embedding Benchmark (MTEB). Figure excerpted from the original paper [MTMR23].

3.4.1 Types of Tasks

This overview is based on information provided in the original paper [MTMR23].

- **Clustering**: Group a set of texts such that texts in the same group are more similar to each other than to texts in other groups. MTEB utilizes the k-means algorithm [Llo82, M+67] for this task and scores the models using the v-measure metric [RH07].
- **Pair Classification**: The task of assigning a label to a pair of texts. MTEB uses the average precision score [ZZ09] using cosine similarity as the main metric for this task.
- **Summarization**: Given human-written and machine-generated summaries, the task is to score the machine-generated summaries. MTEB uses the Spearman correlation based on cosine similarity [RBG16] as the main metric.
- **Re-ranking**: Given a query and a set of relevant and irrelevant texts, the task is to rank the texts in a way that the relevant texts are ranked higher. MTEB uses the mean average precision [ZZ09] for this task as the main metric.
- **STS**: Given a pair of sentences, compute the similarity between the sentences. As for summarization, MTEB uses the Spearman correlation using cosine similarity [RBG16] as the main metric.
- **Classification**: Classify a text into a predefined category after training a logistic regression model. MTEB uses accuracy as the main metric.

- **Retrieval**: Given a corpus of documents and a query, the task is to retrieve the most relevant documents. MTEB uses the normalized discounted cumulative gain (NDCG) [JK02] as the main metric.

3.4.2 Types of Datasets

This overview is based on information provided in the original paper [MTMR23].

All datasets of the MTEB can be put into one of three categories:

- **S2S**: The embedding of a sentence is compared to the embedding of another sentence. In MTEB, all STS tasks are S2S tasks.
- **P2P**: The embedding of a paragraph is compared to the embedding of another paragraph. The authors of the paper note that there is no upper limit for the length of the paragraphs.
- **S2P**: There are a few retrieval datasets that compare the embedding of a sentence to the embedding of a paragraph.

3.5 Models

In the following, the characteristics of the models which are used in the experiments are described. For an overview of how these models were selected, please refer to Subsection 4.1.1.

Stella

The stella_en_400M_v5 model is a transformer model with 400 million parameters. The author of the model is "dunzhang" and the model is allegedly trained based on the Alibaba-NLP/gte-large-en-v1.5 and Alibaba-NLP/gte-Qwen2-1.5B-instruct models. According to the author, the model has been trained using the Matryoshka representation objective, with the default output dimensionality being 8192. Unfortunately, the author did not provide any further information about the model, which is why any results obtained from this model should be taken with caution. [dun24]

MBAI-1D

The mxbai-embed-large-v1 model is a transformer model with 335 million parameters, released by the company Mixedbread-AI. Mixedbread-AI specializes in embedding and reranking models and has released several models on Hugging Face. The authors confirm that the model has been trained using the Matryoshka representation training objective but do not provide further information about the exact dimensionalities used for training. The model is designed for tasks in the English language, has 24 layers and a default embedding size of 1024. The authors state that for training the model, they made sure that the training data has no overlap with the MTEB. Allegedly, they have trained the

model using contrastive learning on 700 million pairs and over 30 million triplets using the AnglE objective [LL23]. The company also noted that this model is "well suited" for binary embeddings. [ma24b]

Nomic

The NOMIC-EMBED-TEXT-V1.5 model is a transformer model with 137 million parameters. The author of the model is an organization called "nomic-ai," which has released a total of 19 models on Hugging Face. They confirm that the model has been trained using the Matryoshka representation training objective with a default output dimensionality of $d = 768$. The authors have trained this model by fine-tuning another in-house model called "NOMIC-EMBED-TEXT-UNSUPERVISED" on the "NOMIC-EMBED-TEXT" dataset, consisting of 200 million examples. Similar to MBAI-1D, the authors state that the model supports binary embeddings. [na24]

MBAI-2D

The MXBAI-EMBED-2D-LARGE-V1 model is a transformer model with 335 million parameters, released by the company Mixedbread-AI. In contrast to the other models, this model also supports 2D Matryoshka representations. Besides the adjusted training objective, the model is similar to MBAI-1D in terms of architecture and training data. [ma24a]

4 Evaluation

This chapter will evaluate the combination of Matryoshka representations and quantization with respect to performance, evaluation time, and memory consumption. First, in Section 4.1, the experimental setup will be elaborated on, which includes the model selection, the parameters used for the experiments, the benchmarks selected as well as the software and hardware used for the experiments. Furthermore, that section will introduce sub-questions which will help to answer the main research question of this thesis. The answers to those sub-questions will be presented in Section 4.2 and Section 4.3, where the results for 1D and 2D Matryoshka representations paired with quantization will be discussed respectively. Finally, information about the CO_2 emissions of the experiments will be provided in Section 4.4.

4.1 Experimental Setup

4.1.1 Model Selection

Since Hugging Face provides the best source for pre-trained and open-source models, the selection of the models was based on the Hugging Face leaderboard for the Massive Text Embeddings Benchmark (MTEB).

As explained in Chapter 2, both types of Matryoshka representations are enforced by a modified training objective. The authors of the original paper have shown that models that have not been trained using the Matryoshka representation objective naturally do not perform well when a subspace of the embedding vector is used. Similar findings have been reported in the paper which first introduced the concept of 2D Matryoshka representations. Thus, the models selected for this thesis are models that have been trained using either the 1D or 2D Matryoshka representation objective, as discussed in Sections 2.1 and 2.2.

The Hugging Face leaderboard also contains many models that exceed billions of parameters. Due to the high amount of repetition caused by the complexity of the experiments, described in Equation 3.1 in the previous chapter, an upper bound on the model size is set to 1 billion parameters. This cutoff point was chosen as it serves as a good separation between the many small and few large models on the MTEB leaderboard and allows all models considered to be run on the same hardware. Algorithm 2 provides a conceptual overview of the model selection process.

Table 3 in the appendix lists the top 100 models on the MTEB Hugging Face leaderboard at the time of writing this thesis. Furthermore, Table 4 in the appendix shows all models which do not exceed the 1 billion parameter limit. All models which are listed in bold

Algorithm 2 Conceptual overview of the model selection process

```
for each entry in the MTEB Hugging Face leaderboard up to 100 do
    if model parameters ≤ 10^9 then
        if (model supports 1D MR) OR (model supports 2D MR) then
            select model
        end if
    end if
end for
```

in Table 4 support either 1D or 2D Matryoshka representations and are thus selected for the experiments. The selected models are:

- **stella_en_400M_v5** also referred to as **Stella** [dun24]
- **mxbai-embed-large-v1** also referred to as **MBAI-1D** [ma24b]
- **nomic-embed-text-v1.5** also referred to as **Nomic** [na24]
- **mxbai-embed-2d-large-v1** also referred to as **MBAI-2D** [ma24a]

For a description of the models, please refer to Section 3.5.

4.1.2 Parameters

Matryoshka representations

As discussed in Chapter 3, most creators of the models do not provide information about the exact Matryoshka sizes used for training. However, since the authors of the original paper have shown that there is interpolating behavior for Matryoshka representations (see Figure 2.4), knowledge about the exact dimensionalities used for training is not necessary for the experiments.

For each model, eight Matryoshka sizes will be used, with a reduction to the next lower power of two for each step starting with the maximum output dimensionality of the model. This follows the procedure of the authors from the original 1D MR paper, which was discussed in Section 2.1. The Matryoshka sizes considered for each model can be seen in Table 4.1.

Model	**Matryoshka sizes**
Stella ($d = 8192$)	8192, 4096, 2048, 1024, 512, 256, 128, 64
MBAI-1D ($d = 1024$)	1024, 512, 256, 128, 64, 32, 16, 8
Nomic ($d = 768$)	768, 512, 256, 128, 64, 32, 16, 8
MBAI-2D ($d = 1024$)	1024, 512, 256, 128, 64, 32, 16, 8

Table 4.1: Matryoshka sizes used for each model. d refers to the maximum (default) output dimensionality of the model.

For MBAI-2D, which is the only 2D Matryoshka representation model, the intermediate layers 20, 16, and 12 will be evaluated. This procedure is based on the linear degradation in performance observed in the original paper as discussed in Section 2.2 and visualized

in Figure 2.7a. To test if the behavior of the full model is consistent with the others, the results will also be tracked for layer 24 (the last layer of the model).

Quantization Techniques

Following previous work on the quantization of embeddings, (signed) int8 and binary quantization techniques will be used for the experiments alongside the default floating-point 32-bit (float32) embeddings. For an explanation of the quantization techniques, please refer to Section 3.3.

4.1.3 Benchmark Selection

The Massive Text Embeddings Benchmark (MTEB) [LL23] will be used to evaluate the performance of the quantized Matryoshka representations. However, as the name of the benchmark hints, running the entire benchmark is computationally expensive and is not feasible within the scope of this thesis. Thus, a subset of the benchmark will be selected based on tangible criteria.

Task Category	**Tasks**
Clustering	ArXivHierarchicalClusteringP2P.v2, ArXivHierarchicalClusteringS2S.v2, BiorxivClusteringP2P.v2, BiorxivClusteringS2S.v2, MedrxivClusteringP2P.v2, MedrxivClusteringS2S.v2, RedditClustering.v2, StackExchangeClustering.v2, StackExchangeClusteringP2P.v2, TwentyNewsgroupsClustering.v2
Pair Classification	SprintDuplicateQuestions, TwitterSemEval2015, TwitterURLCorpus
Summarization	SummEval
Reranking	AskUbuntuDupQuestions, MindSmallReranking, StackOverflowDupQuestions
STS	BIOSSES, SICK-R, STS12, STS13, STS14, STS15, STS16, STSBenchmark, STS17, STS22
Classification	Banking77Classification, EmotionClassification, TweetSentimentExtractionClassification, AmazonCounterfactualClassification, MassiveIntentClassification, MassiveScenarioClassification, MTOPDomainClassification, MTOPIntentClassification
Retrieval	ArguAna, CQADupstackWebmastersRetrieval, NFCorpus

Table 4.2: Selected tasks from the MTEB for each task category.

First, all tasks that are part of the MTEB subset used for the Hugging Face leaderboard were considered. After that, all tasks which were too expensive in terms of computation were omitted. Recall the computational complexity of the experiments described in Equation 3.1. In total, four models will be evaluated. Three of them are 1D MR models and one is a 2D MR model, which will be analyzed for four different layers. This means that there are seven model configurations that are evaluated for eight Matryoshka sizes and three quantization techniques. This results in a single task being repeated for $8 \times 7 \times 3 = 168$ times.

To determine whether a task was too expensive, the task was run once on every model and the time it took to complete the task was measured. For this, the default configurations of the respective models were used (i.e., float32 embeddings, default embedding size, and last layer for inference). It was decided that each task that took longer than 15 minutes *on any model* to complete would be omitted from the subset.

The selected tasks are summarized in Table 4.2. The evaluations will be performed on the respective test sets of the tasks, as this is also being done in the MTEB leaderboard and the intended workflow as described by the authors of the benchmark. Furthermore, the default main metric for each task category, as described in Section 3.4, will be used to evaluate the performance of the models. For more information regarding the individual tasks, please refer to the original MTEB paper ([MTMR23]).

Partially excluded tasks

Unfortunately, due to some implementation issues causing GPU memory problems, the Nomic model could not be evaluated on the following tasks:

- MedrxivClusteringP2P.v2 (Clustering)
- STS22 (Semantic Textual Similarity)
- CQADupstackWebmastersRetrieval (Retrieval)
- NFCorpus (Retrieval)

4.1.4 Software

All experiments will be conducted in the programming language Python. Python is widely used in the field of machine learning thanks to its simplicity and large ecosystem of libraries. Furthermore, the Hugging Face Transformers library [WDS+20], which builds on top of PyTorch, will be utilized [PGM+19]. PyTorch, developed by Meta, is one of the two most popular deep learning frameworks, the other being TensorFlow developed by Google. Alongside the Transformers library, Hugging Face also provides datasets, common utility functions, and a discussion forum for the community. Besides PyTorch and the Transformers library, the following libraries will be used:

- MTEB [MTMR23] for the benchmarks
- NumPy [HMvdW+20] for numerical operations
- Matplotlib [Hun07] for plotting

- Pandas [pdt20] for exporting data into tables

All experiments will be conducted in Jupyter notebooks, which can be accessed via a GiHub repository.

4.1.5 Hardware

The code will be executed on Google Colab, which is a remote Jupyter notebook environment provided by Google. For this thesis, the Pro version of Google Colab will be used, which provides a higher amount of RAM, longer runtime, and easier access to GPUs. For the sake of reproducibility, all experiments will be conducted on an NVIDIA Tesla T4 GPU with 16 GB of video memory. All notebooks used will have 50 GB of system RAM available.

4.1.6 Research Questions

Sub-questions have been formulated to help answer the main research question of this thesis which was introduced in Chapter 1. The following three sub-questions will be considered for the combination of 1D Matryoshka representations with quantization in Section 4.2:

- **RQ-1D-1**: How does the combination of 1D Matryoshka representations and quantization affect the **performance** across tasks?
- **RQ-1D-2**: How does the combination of 1D Matryoshka representations and quantization affect the **evaluation time** across tasks?
- **RQ-1D-3**: What is the trade-off between **performance** and **memory consumption** for the combination of 1D Matryoshka representations and quantization?

In Section 4.3, these questions will help to evaluate the combination of 2D Matryoshka representations with quantization:

- **RQ-2D-1**: How does the combination of 2D Matryoshka representations and quantization affect the **performance** across tasks?
- **RQ-2D-2**: How does the combination of 2D Matryoshka representations and quantization affect the **evaluation time** across tasks?
- **RQ-2D-3**: What is the trade-off between **performance**, **memory consumption**, and **evaluation time** for the combination of 2D Matryoshka representations and quantization?

Finally, all results will be combined in Section 5.1 to answer the main research question.

4.2 Quantized 1D Matryoshka Representations

In the following the combination of 1D Matryoshka representations and quantization will be evaluated to answer the research questions **RQ-1D-1**, **RQ-1D-2**, and **RQ-1D-3** presented in Subsection 4.1.6.

For this, the following models will be considered:

- MXBAI-EMBED-LARGE-V1
- STELLA_EN_400M_V5
- NOMIC-EMBED-TEXT-V1.5

In each upcoming subsection, the results for the MBAI-1D model will be discussed first, followed by an elaboration on how the results for the other two models differ.

4.2.1 Performance

This section aims to answer the first research question, **RQ-1D-1**, analyzing how the performance of the models changes when using 1D Matryoshka representations and quantization.

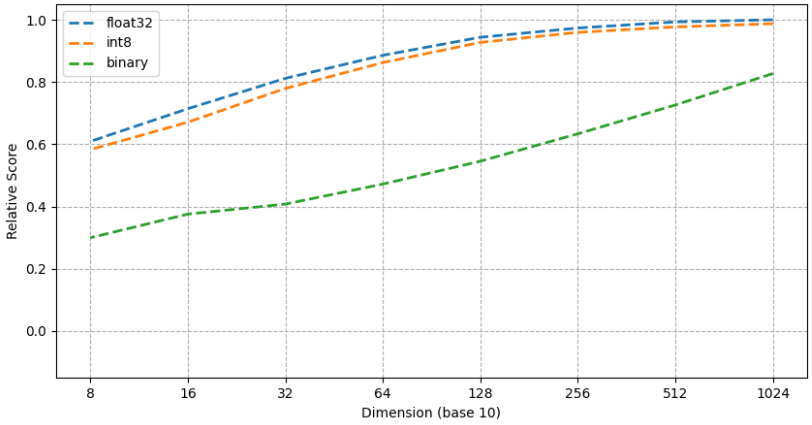


Figure 4.1: Average performance of the MBAI-1D model over all selected MTEB tasks. The x-axis represents the embedding size while the y-axis represents the relative performance, averaged over all tasks. For each task the reference point is the performance of the float32 embeddings at $m = 1024$. Blue: float32, Orange: int8, Green: binary.

Figure 4.1 illustrates the performance of the MBAI-1D model across all selected MTEB tasks for all combinations of Matryoshka representations and quantization techniques. To accurately compare the performance across different tasks, the performance scores are normalized to obtain a relative performance metric. This is done by dividing the score of each task when using a specific Matryoshka size and quantization technique by the score achieved with the full output dimensionality ($m = d$) and without quantization. Note that "score" refers to the main metric of the respective task category, as described

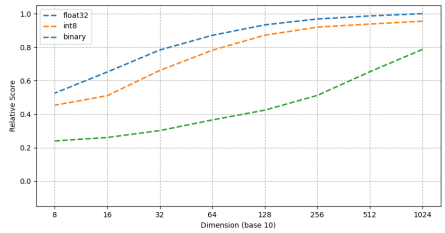
in Section 3.4. Following this, the relative performance is averaged over all tasks for each Matryoshka size and quantization technique. This approach ensures a more balanced comparison, as the absolute performance of the model is irrelevant and only the relative performance decrease when quantization and Matryoshka representations are applied, is of interest.

Looking at the plot (Figure 4.1), the following observations can be made:

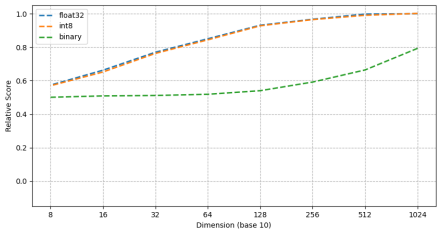
- **Float32 embeddings:** The float32 embeddings behave just like described by the authors of the original Matryoshka representations paper. For the first few reductions from 1024 all the way down to 128, there is only a very minor reduction, accounting for only 4.8 % relative to the original performance. As the Matryoshka representations keep getting smaller, the relative loss in performance increases. For example, with a reduction to 64 dimensions and below almost 10 % of the original performance is lost.
- **Int8 embeddings:** The int8 embeddings almost match the performance of the float32 ones, trailing by only 6.6 % at most.
- **Binary embeddings:** The binary embeddings perform significantly worse at all Matryoshka sizes. At $m = 1024$, the performance difference between the float32 and binary embeddings accounts for over 17% and increases as the Matryoshka size gets smaller.

Looking more closely, Figure 4.2 shows the relative performance of the MBAI-1D model for each task category in the MTEB separately. Again, the reference point for each task is the performance of the model with the full output embedding size and no quantization. The only difference to the previous plot is that now the relative performance is no longer averaged over all tasks, but only over tasks of the same category. The key observations from inspecting these seven subfigures are as follows: Regardless of the task, performance consistently decreases as the Matryoshka size gets smaller. Among the quantization techniques, binary embeddings perform significantly worse than the other two. As illustrated in the figure, int8 embeddings perform nearly as well as float32 embeddings across all task categories, except for classification tasks (Figure 4.2a), where a larger discrepancy can be observed. For all other tasks, the difference in relative performance between float32 and int8 embeddings is at most 4.5 % (regardless of the Matryoshka size considered), with most differences being below 1%. Note that the summarization tasks (Figure 4.2g) show the smallest decay in performance across all quantization techniques, however since the MTEB only offers a single task for this category, these results should be taken with caution.

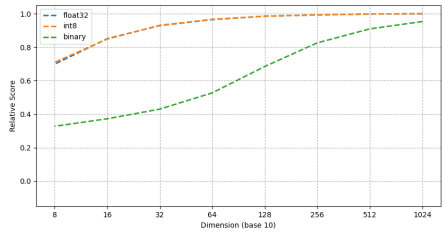
The other two models, Stella and Nomic, show a similar behavior to the MBAI-1D model. Figure 4.3 shows the average performance of the Stella and Nomic models on the selected MTEB tasks in the same fashion as for the first model. While the Nomic model (Figure 4.3b) supports all the observations made so far, the Stella model shows a slightly different behavior. As shown in Figure 4.3a, a noticeable difference is the constant performance for all quantization techniques when reducing the output embedding size from 8192 to 1024. This observation does not fall in line with the other models and the expected behavior for 1D MRs where one expects the performance to constantly decrease as the Matryoshka size gets smaller. Due to the limited information about the model, as discussed in Section 3.5, this phenomenon cannot be explained and these results should



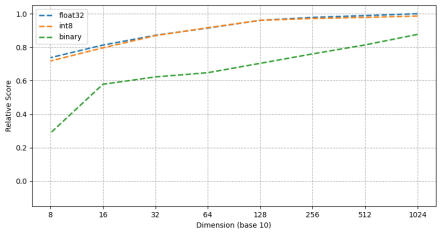
(a) Classification



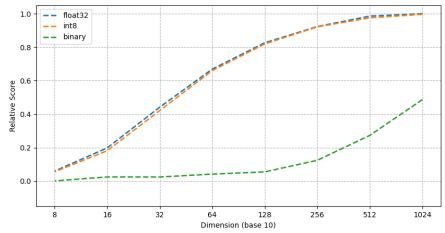
(b) Clustering



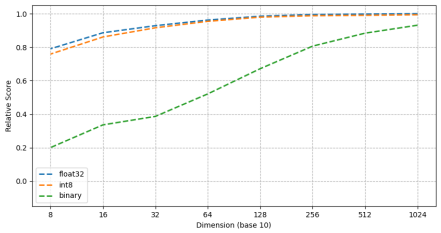
(c) Pair Classification



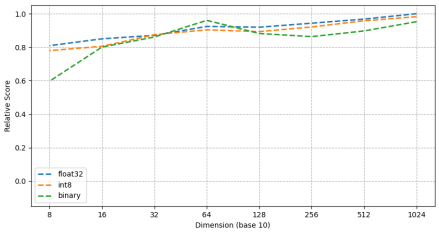
(d) Re-Ranking



(e) Retrieval



(f) Semantic Textual Similarity



(g) Summarization

Figure 4.2: Relative performance of the MBAI-1D model for each task category in the MTEB. The x-axis of each subfigure represents the embedding size while the y-axis represents the relative performance. For each task the reference point is the performance of the float32 embeddings at the full embedding size. Blue: float32, Orange: int8, Green: binary.

be taken with caution. From $m = 1024$ and below, the trajectory of the performance mimics the behavior of the MBAI-1D model, with the binary embeddings seeing a linear decrease in performance as the Matryoshka size decreases and the int8 embeddings performing almost as well as the float32 ones.

(a) Stella **(b)** Nomic

Figure 4.3: Relative performance of the Stella and Nomic models over all selected MTEB tasks. Subfigure (a) shows the performance of the Stella model, while subfigure (b) shows the performance of the Nomic model. In both subfigures, the x-axis represents the embedding size while the y-axis represents the relative performance, averaged over all tasks. For each task the reference point is the performance of the float32 embeddings at $m = 8192$ for the Stella model and $m = 768$ for the Nomic model. Blue: float32, Orange: int8, Green: binary.

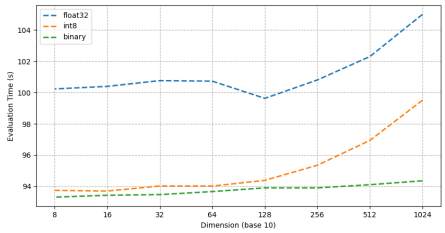
Concluding **RQ-1D-1**, the combination of 1D Matryoshka representations and quantization affects the performance across tasks. Both the reduction of the embedding size as well as the quantization of the embeddings lead to a decrease in performance. The int8 embeddings perform almost as well as the float32 ones for all Matryoshka sizes investigated, with the only exception being the classification tasks. The binary embeddings perform significantly worse than the other two quantization techniques.

The appendix provides detailed information about the performance of the models, both in relative and absolute terms as well as for each task and task category. For the MBAI-1D model please refer to Tables 11 to 14. For the Stella and Nomic models, please refer to Tables 19 to 22 and Tables 27 to 30 respectively. Furthermore, the plots for the relative performance of the Stella and Nomic models for each task category can be found in Figure 10 and Figure 13 in the appendix.

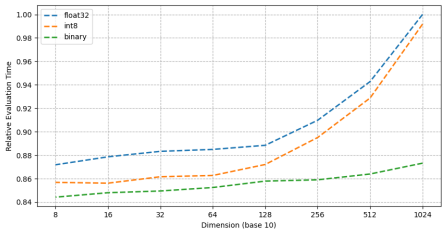
4.2.2 Evaluation Time

Following the analysis of the performance, this section will answer **RQ-1D-2** and discuss how the evaluation time of the models changes when combining MRs with quantization. The evaluation time was measured in seconds and describes the time it takes a model with a specific Matryoshka representation and quantization technique to complete a task in the Massive Text Embeddings Benchmark.

Figure 4.4 shows the average evaluation time for the MBAI-1D model over all selected MTEB tasks. The absolute averages, where the time is measured in seconds, are displayed in Figure 4.4a. The subfigure suggests that the int8 embeddings provide a significant speed-up compared to the float32 embeddings, regardless of the Matryoshka size. However, it is important to note that some tasks take longer to complete than others. An example of this can be seen in Figure 4.5, where the average evaluation time for semantic text similarity (STS) and retrieval tasks is compared with the time on the y-axis being measured in seconds. For float32 embeddings with $m = 1024$, retrieval tasks require up



(a) Absolute

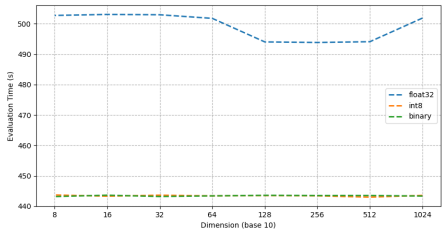


(b) Relative

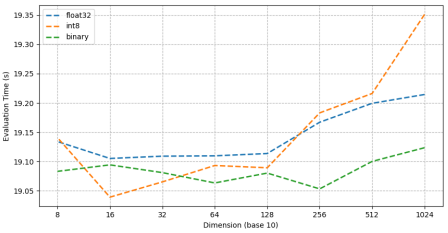
Figure 4.4: Absolute and relative evaluation time of MBAI-1D, averaged over all MTEB tasks. Subfigure (a) shows the absolute evaluation time in seconds, while Subfigure (b) shows the relative evaluation time on the y-axis. In both subfigures, the x-axis represents the embedding size. The reference point for the relative evaluation time for each task is the evaluation time of the float32 embeddings at $m = 1024$. Blue: float32, Orange: int8, Green: binary.

to 500+ seconds while STS tasks only require about 19.21 seconds on average using the MBAI-1D model.

To account for this, the relative evaluation time is calculated in a similar fashion as the relative score in the performance analysis. The relative evaluation time is determined by comparing the time required for each specific Matryoshka representation and quantization technique against the evaluation time of the float32 embeddings at the default output dimensionality (here $m = 1024$). By dividing the evaluation time of each configuration by this baseline the results are being normalized, making it possible to compare evaluation times across different tasks.



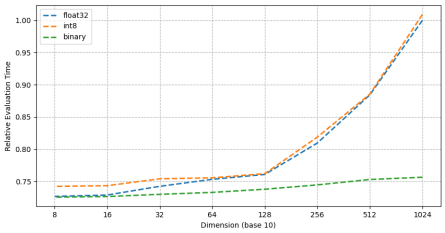
(a) Retrieval



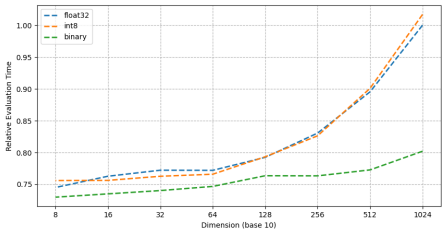
(b) STS

Figure 4.5: Comparison of the absolute evaluation time of MBAI-1D on the retrieval and semantic textual similarity tasks. subfigure (a) shows the absolute evaluation time for the retrieval tasks, while subfigure (b) shows the absolute evaluation time for the semantic textual similarity tasks. In both subfigures the x-axis represents the embedding size while the y-axis represents the evaluation time in seconds. Blue: float32, Orange: int8, Green: binary.

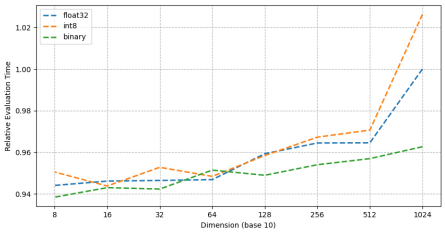
As can be seen in Figure 4.4b, the int8 embeddings are faster than the float32 embeddings for all Matryoshka sizes on average, with the difference increasing as the Matryoshka size decreases. From the same plot, it can also be derived that binary embeddings are more than 10% faster compared to the float32 ones for $m = 1024$, with the difference becoming smaller as the Matryoshka size decreases.



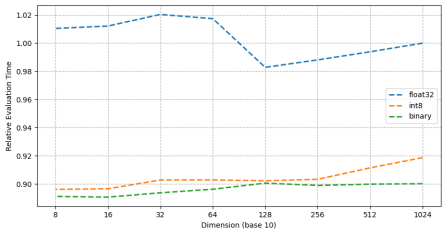
(a) Classification



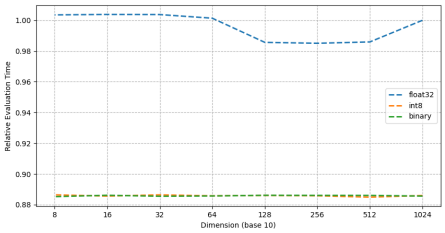
(b) Clustering



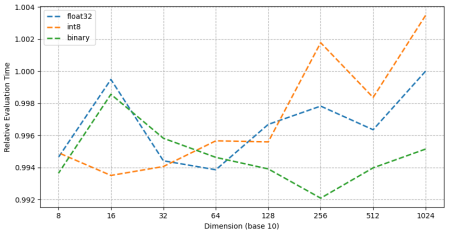
(c) Pair Classification



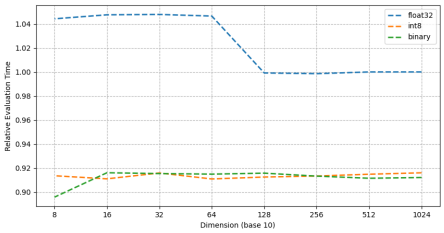
(d) Re-Ranking



(e) Retrieval



(f) Semantic Textual Similarity



(g) Summarization

Figure 4.6: Relative evaluation time of the MBAI-1D model for each task category in the MTEB. The x-axis of each subfigure represents the embedding size while the y-axis represents the relative evaluation time. For each task the reference point is the evaluation time of the float32 embeddings at the full embedding size. Blue: float32, Orange: int8, Green: binary.

One might have noticed in Figure 4.5b, that the int8 embeddings are slower than the float32 embeddings for the semantic text similarity (STS) tasks. Comparing the relative evaluation time for each task category, displayed in Figure 4.6, one can see that for some Matryoshka sizes the int8 embeddings are slower than the float32 embeddings for classification (Figure 4.6a), clustering (Figure 4.6b), pair classification (Figure 4.6c) and semantic

textual similarity (Figure 4.6f) tasks. This could be due to the fact that the quantization of the embeddings requires additional computation, which in some scenarios may outweigh the benefits of the cheaper calculations. Another possible explanation for this effect can be the different task complexities as well as the specific implementation of the tasks in the MTEB. When inspecting the seven subplots, it also becomes apparent that classification and clustering are the task categories where the greatest speed-ups (up to 25%) can be achieved when lowering the Matryoshka size. In most cases, the binary embeddings provide a significant speed-up compared to the float32 embeddings, with the difference being the largest for the classification tasks for $m = 1024$.

(a) Stella **(b)** Nomic

Figure 4.7: Relative evaluaton time of the Stella and Nomic models over all selected MTEB tasks. Subfigure (a) shows the evluation time of the Stella model, while subfigure (b) shows the evluation time of the Nomic model. In both subfigures, the x-axis represents the embedding size while the y-axis represents the relative evluation time, averaged over all tasks. For each task the reference point is the evluation time of the float32 embeddings at $m = 8192$ for the Stella model and $m = 768$ for the Nomic model. Blue: float32, Orange: int8, Green: binary.

Figure 4.7 shows the average evaluation time for the Stella and Nomic models over all selected MTEB tasks relative to the evaluation time of the float32 embeddings at the default output dimensionality of the models. The Stella model, which is shown in Figure 4.7a, shows the same key characteristics as the MBAI-1D model when it comes to evaluation time. However, similar to the performance analysis, there is no noticeable difference in average evaluation time for any quantization technique when reducing the output embedding size from 8192 to 1024.

The Nomic model, shown in Figure 4.7b, is the only model where the int8 embeddings are not faster than the float32 embeddings for all Matryoshka sizes when averaged over all tasks. As mentioned in Subsection 4.1.1, the Nomic model lacks results for a few benchmarks, especially in the retrieval category. Since there are no major differences observable in the other task categories, as shown by Figure 14 in the appendix, this difference is likely caused by the absence of the four benchmarks.

Concluding **RQ-1D-2**, the combination of 1D Matryoshka representations can provide a speed-up in evaluation time. Between int8 and float32 embeddings, there is no clear difference in evaluation time, with the int8 embeddings being faster by a tiny margin on most but not all task-model combinations. Binary embeddings on the other hand do provide a very noticeable speed-up compared to the float32 embeddings. On average, the evaluation time decreases as the Matryoshka size gets smaller, with the effect being especially noticeable on classification and clustering tasks.

The appendix provides detailed information about the evaluation time of the models, both in relative and absolute terms as well as for each task and task category. For the MBAI-1D model please refer to Tables 7 to 10. For the STELLA and NOMIC models, please refer to Tables 15 to 18 and Tables 23 to 26 respectively. Furthermore, the plots for the relative evaluation time of the STELLA and NOMIC models for each task category can be found in Figure 11 and Figure 14 in the appendix.

4.2.3 Performance and Memory Consumption Trade-Off

To finish the experiments on 1D Matryoshka representations paired with quantization, the relationship between memory consumption and the performance of the models will be discussed to answer the third research question, **RQ-1D-3**. Introduced in Section 3.2, the memory consumption of embeddings scales with both their size and the quantization technique used. In the following, the memory consumption will be denoted using bits and describe the theoretical amount of memory required to store the embeddings for a specific Matryoshka size and quantization technique. For example, the memory consumption of an embedding with $m = 1024$ and no quantization (float32) is $1024 \times 32 = 32768 = 2^{15}$ bits.

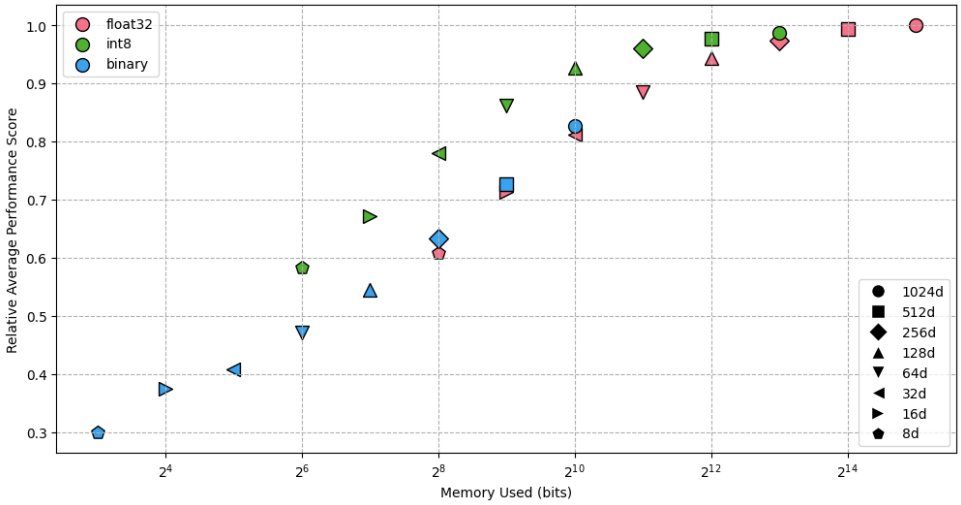
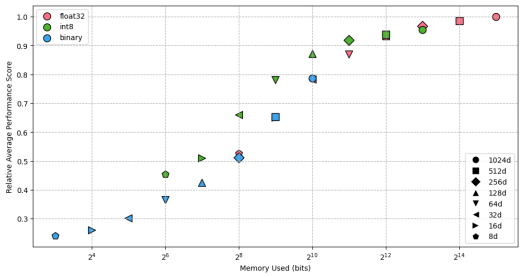
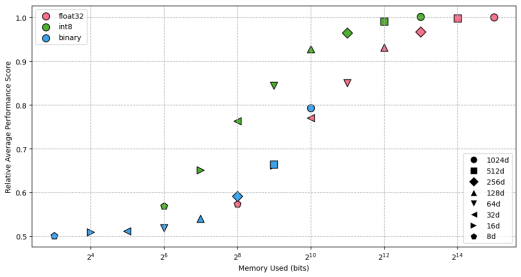


Figure 4.8: Relative accuracy compute trade-off of the MBAI-1D model, averaged over all selected MTEB tasks. The x-axis represents the memory consumption of a specific MR-quantization combination measured in bits and is scaled logarithmically. The y-axis represents the relative performance of the respective MR-quantization combination, averaged over all tasks. The reference point for the relative performance for each task is the performance of the float32 embeddings at $m = 1024$. The color of the markers represent their quantization technique, while the shape represents the embedding size.

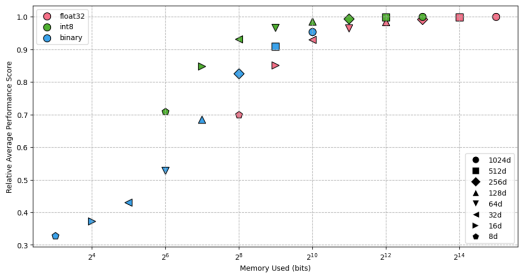
Figure 4.8 shows the relative performance, averaged over all selected MTEB tasks for the MBAI-1D model. Since the x-axis now represents the memory consumption in bits, this can be seen as a multicriteria optimization problem, where the goal is to maximize the performance while minimizing the memory consumption. The top left corner of the plot represents the best performance with the lowest memory consumption.



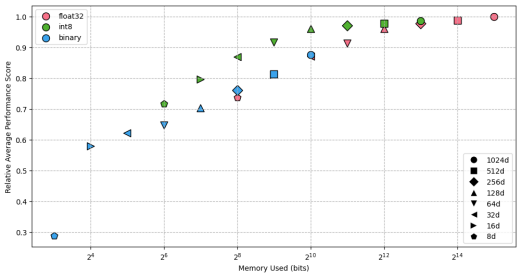
(a) Classification



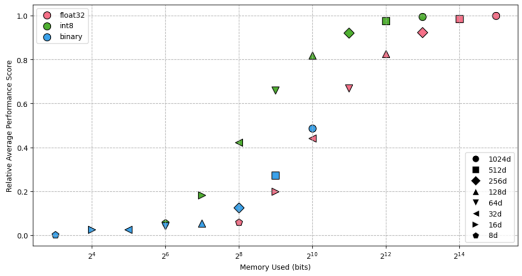
(b) Clustering



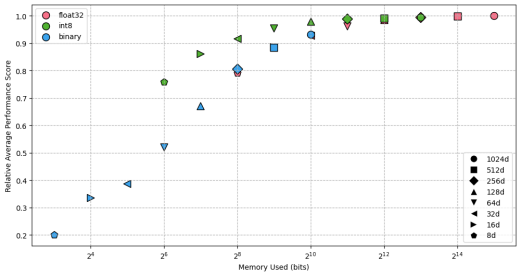
(c) Pair Classification



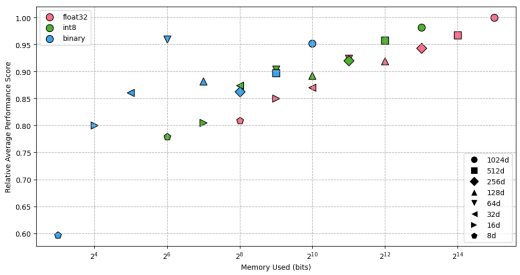
(d) Re-Ranking



(e) Retrieval



(f) Semantic Textual Similarity



(g) Summarization

Figure 4.9: Relative accuracy compute trade-off of the MBAI-1D model, averaged over all selected MTEB tasks within each category. The x-axis represents memory consumption (log scale) and the y-axis represents relative performance (averaged over tasks within the category). Reference point: float32 embeddings at full size.

As was already known from the previous sections, the most memory-intensive float32 embeddings with the full embedding size of $m = 1024$ provide the best performance. There are some memory budgets for which only a single quantization technique is available, as the other quantization techniques do not provide embeddings with the same mem-

ory consumption. However, for all compute budgets between 2^{13} and 2^6 bits, there are multiple combinations requiring the same amount of memory. For example, this is the case for float32 with $m = 256$ and int8 with $m = 1024$ at a memory consumption of 2^{13} bits.

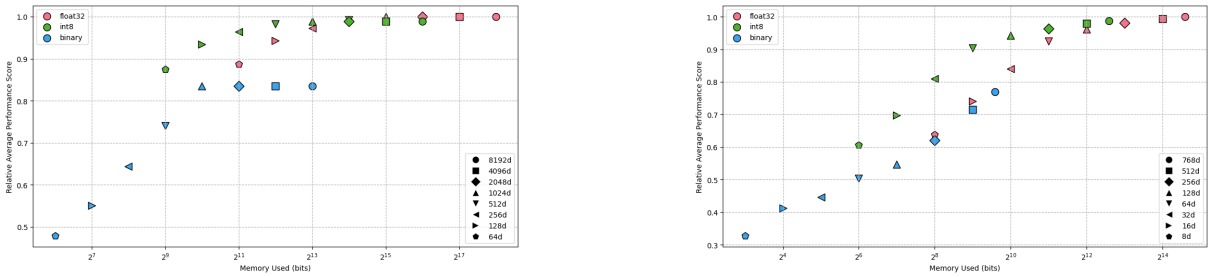
As can be seen in Figure 4.8, both of these embeddings have the same memory consumption (2^{13} bits), but the performance of the int8 embeddings is slightly better than that of the float32 ones. The difference in performance between the int8 and float32 embeddings then increases for pairs with equal memory consumption as the memory consumption decreases. For all memory budgets between 2^{13} and 2^6 bits, the int8 embeddings provide the best performance of all quantization techniques tested. For instance, the int8 representation for $m = 256$ achieves a relative performance of 96% while only using 2^{11} bits of memory. This means that the memory can be reduced by a factor of 16 while only accounting for a 4% decrease in performance when compared to the full output dimensionality and no quantization.

There appears to be very limited use for binary embeddings, as their performance is superseded by int8 quantization for all memory budgets between 2^{10} and 2^6 bits. Even though the binary embeddings for $m = 32, 16, 8$ are Pareto optimal (as there is no other embedding with the same memory consumption) and very efficient, they retain less than 50% of the performance when compared to the baseline.

Figure 4.9 shows the relative performance for the MBAI-1D model, averaged over all tasks within the same task category. Again, it is worth noting that classification (Figure 4.9a) is the only task category where the float32 embeddings provide a noticeable performance increase over the int8 embeddings with a relative difference of almost 4.5 percent. However, for all other tasks considered, the use of high-dimensional float32 embeddings is hard to justify, as the performance of the int8 embeddings is at most 1.8 % worse than that of the float32 ones. Overall, the characteristics of the plots are consistent for all task categories except for summarization. Again, this is likely due to the fact that the MTEB only offers a single task for this category.

Figure 4.10a shows the average performance over all selected MTEB tasks for the Stella model, which overall shows a similar trend as observed for the MBAI-1D model. However, as was already observed in both the performance and evaluation time analysis, the Stella model shows no clear changes when decreasing the Matryoshka size from 8192 to 1024. This behavior, which is not in line with the expected behavior for 1D Matryoshka representations, increases the trade-off between performance and memory consumption. For a memory budget of 2^{12} bits, the int8 embeddings with $m = 512$ only perform 1.8% worse than the float32 embeddings with $m = 8192$ while reducing the bits required by a factor of 64. Similar to the MBAI-1D model, the binary embeddings can provide even greater memory reduction but at the cost of a significant decrease in performance.

The Nomic model, which is displayed in Figure 4.10b, shows the same characteristics as the MBAI-1D model.



(a) STELLA **(b)** NOMIC

Figure 4.10: Relative accuracy compute trade-off of the STELLA and NOMIC models, averaged over all selected MTEB tasks. Subfigure (a) shows the trade-off for the STELLA model, while subfigure (b) shows the trade-off for the NOMIC model. In both subfigures, the x-axis represents the memory consumption of a specific MR-quantization combination measured in bits and is scaled logarithmically. The y-axis represents the relative performance of the respective MR-quantization combination, averaged over all tasks. The reference point for the relative performance for each task is the performance of the float32 embeddings at $m = 8192$ for the STELLA model and $m = 768$ for the NOMIC model. The color of the markers represent their quantization technique, while the shape represents the embedding size.

Finally, answering **RQ-1D-3**, the combination of 1D Matryoshka representations and quantization provides an opportunity to reduce the memory consumption of the models while maintaining high performance. Int8 embeddings present the best trade-off between performance and memory consumption, offering almost the performance of float32 embeddings at a fraction of the memory consumption. Binary representations can offer an even greater reduction in memory consumption but at the cost of a significant decrease in performance.

The appendix provides detailed information about the performance and memory consumption of the models, both in relative and absolute terms as well as for each task and task category. For the MBAI-1D model please refer to Tables 11 to 14. For the STELLA and NOMIC models, please refer to Tables 19 to 22 and Tables 27 to 30 respectively. Furthermore, the plots for the relative accuracy-compute tradeoff of the STELLA and NOMIC models for each task category can be found in Figure 12 and Figure 15 in the appendix.

4.3 Quantized 2D Matryoshka Representations

This section will evaluate the combination of 2D Matryoshka representations and quantization, to answer the research questions **RQ-2D-1**, **RQ-2D-2**, and **RQ-2D-3** presented in Subsection 4.1.6. All following experiments are based on the evaluation of the MXBAI-EMBED-2D-LARGE-V1 model, as it is the only model that supports 2D Matryoshka representations.

4.3.1 Performance

Starting with **RQ-2D-1**, the impact of 2D MR paired with quantization on the performance of the model will be analyzed.

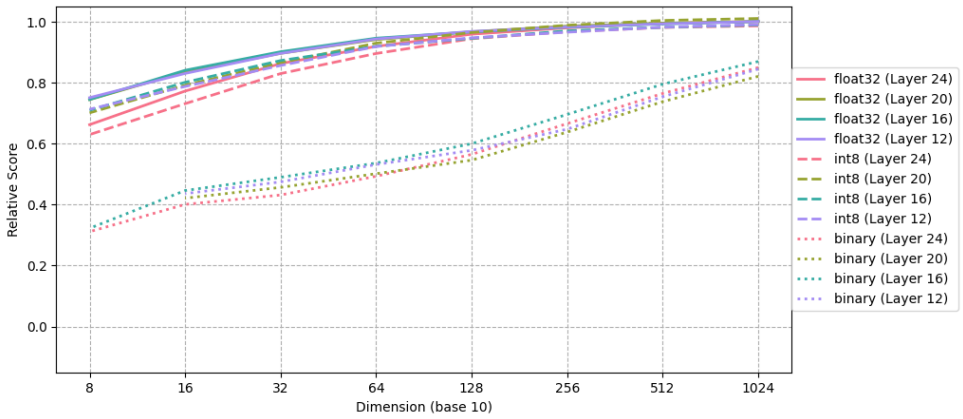
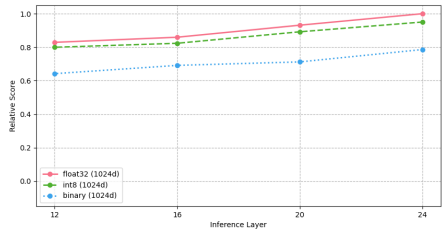


Figure 4.11: Relative performance of the MBAI-2D model, averaged over all selected MTEB tasks. The color of the line represents the layer used for inference, while the type of line represents the quantization technique. Red lines: 24th layer, yellow-green lines: 20th layer, turquoise lines: 16th layer, purple lines: 12th layer. Solid lines: float32 embeddings, dashed lines: int8 embeddings, dotted lines: binary embeddings.

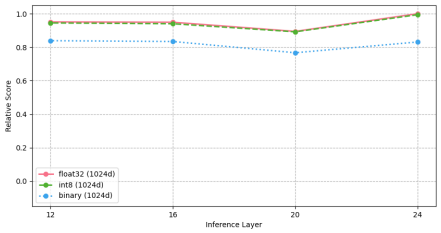
When only considering the last layer of the model, the performance of the MBAI-2D behaves similarly to the MBAI-1D model, which was discussed in the previous section. Comparing the numerical results for both models, provided in the appendix (Tables 11 and 39 to 41) respectively, one can see that the performance of the MBAI-2D model is within a 1 percent margin of the MBAI-1D model for each combination of embedding size, and quantization technique.

Figure 4.11 shows the relative performance of the MBAI-2D model across various Matryoshka sizes, quantization techniques **and layers used for sampling**, averaged over all tasks. Similar to the models investigated in the previous section, the int8 embeddings perform almost as well as the float32 ones, while the binary embeddings perform significantly worse. Also, the degradation in performance when reducing the output embedding size is consistent with the results from the previous section, regardless of the layer used for inference.

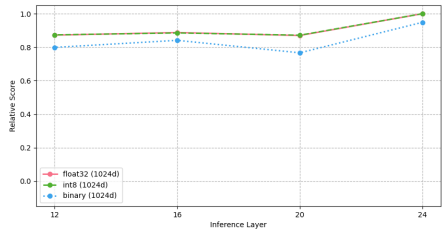
Figure 4.12 shows the relative performance of the MBAI-2D model, averaged over all selected tasks of the MTEB which are within the same task category. For this, the Matryoshka size is fixed to $m = 1024$, as the results from Figure 4.11 have shown that the performance decrease for a reduction in the Matryoshka size is consistent across all layers. Especially interesting is the dip in performance at layer 20 for all quantization techniques, which is most noticeable in the retrieval category (Figure 4.12e). Note that these drops are also to be found in the other task categories, although they are less pronounced. This is somewhat unexpected as the authors of the original paper have shown that the performance degradation when choosing a layer closer to the input layer behaves linearly and without sudden changes in performance (shown in Section 2.2 in Figure 2.7a). However, zooming out it is clear that the relative performance for both float32 and int8 embeddings



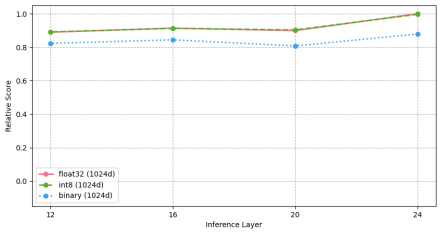
(a) Classification



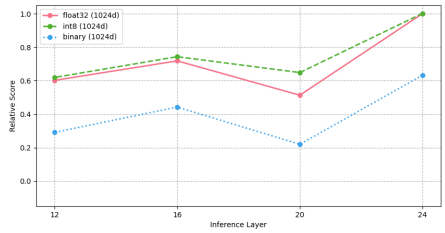
(b) Clustering



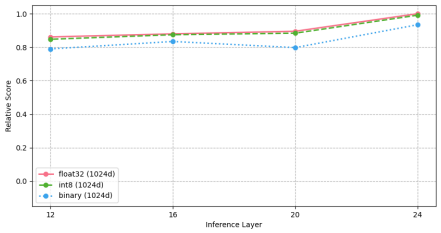
(c) Pair Classification



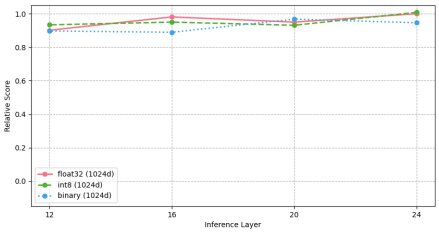
(d) Re-Ranking



(e) Retrieval



(f) Semantic Textual Similarity



(g) Summarization

Figure 4.12: Average performance of MBAI-2D with $m = 1024$ over all selected MTEB tasks for each task category and each sampled layer. The x-axis represents the layer number, while the y-axis represents the relative performance. The reference point for the relative performance for each task is the performance of the float32 embeddings at the full embedding size with the last layer used for inference. The red lines represent the float32 embeddings, the green line the int8 embeddings, and the blue line the binary embeddings.

decreases as the layer number decreases. Binary embeddings follow that trend, with the exception of the clustering (Figure 4.12b) and summarization (Figure 4.12g) tasks, where the performance of the 24th layer seems comparable to the performance of the 12th layer.

To conclude **RQ-2D-1**, the performance of the MBAI-2D model decreases as the layer used for inference moves closer to the input layer. This is true for all quantization techniques when averaged over all tasks. The int8 embeddings perform almost as well as the float32 ones for all Matryoshka sizes and inference layers investigated, while the binary embeddings perform significantly worse. Furthermore, the experiments have shown that there is a drop in performance when sampling from the 20th layer.

Tables 39 to 46 in the appendix provide more information about the performance of the model, both in relative and absolute terms as well as for each task and task category.

4.3.2 Evaluation Time

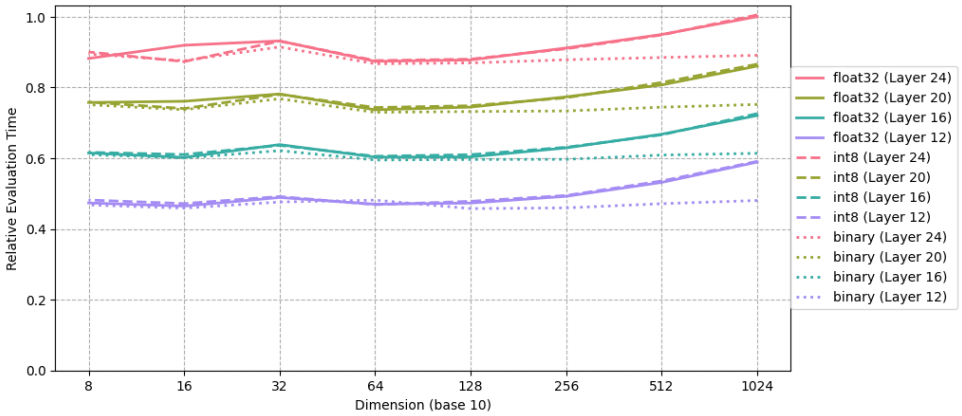
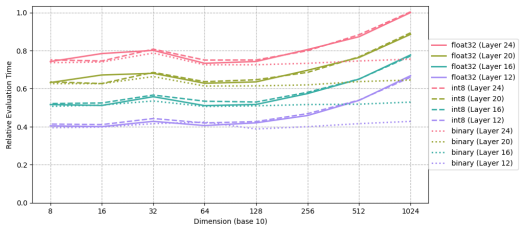


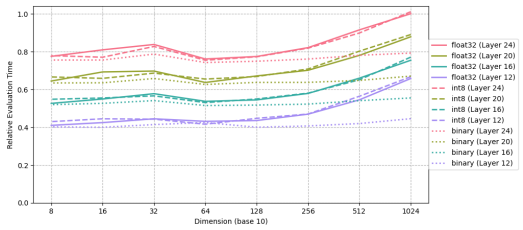
Figure 4.13: Relative evaluation time of MBAI-2D, averaged over all selected MTEB tasks. The x-axis represents the embedding size while the y-axis represents the relative evaluation time. The color of the line represents the layer used for inference, while the type of line represents the quantization technique. Red lines: 24th layer, yellow-green lines: 20th layer, turquoise lines: 16th layer, purple lines: 12th layer. Solid lines: float32 embeddings, dashed lines: int8 embeddings, dotted lines: binary embeddings.

As discussed in the introduction of 2D Matryoshka representations (Section 2.2), their main advantage opposing 1D MRs is the reduced time it takes to compute the embeddings. This section will answer **RQ-2D-2** and discuss how the evaluation time of the MBAI-2D model changes when using 2D Matryoshka representations and quantization.

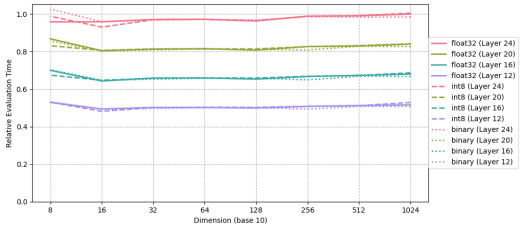
Figure 4.13 shows the relative evaluation time of the MBAI-2D model averaged over all selected MTEB tasks. As the figure shows, the evaluation time averaged across all tasks decreases in a linear fashion as the chosen layer for inference moves closer to



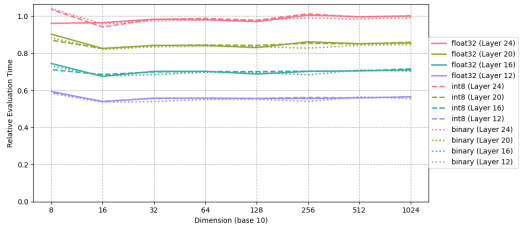
(a) Classification



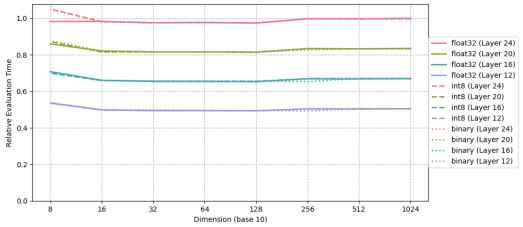
(b) Clustering



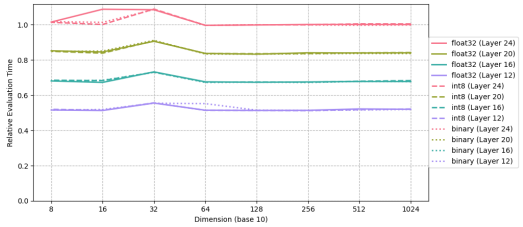
(c) Pair Classification



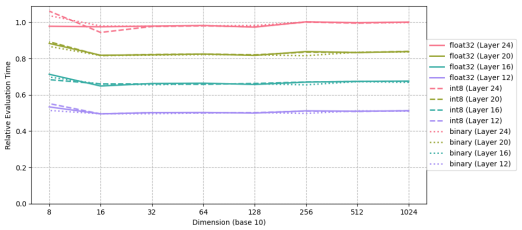
(d) Re-Ranking



(e) Retrieval



(f) Semantic Textual Similarity



(g) Summarization

Figure 4.14: Relative evaluation time of MBAI-2D, averaged over all selected MTEB tasks for each task category. The x-axis represents the embedding size while the y-axis represents the relative evaluation time. The color of the line represents the layer used for inference, while the type of line represents the quantization technique. Red lines: 24th layer, yellow-green lines: 20th layer, turquoise lines: 16th layer, purple lines: 12th layer. Solid lines: float32 embeddings, dashed lines: int8 embeddings, dotted lines: binary embeddings.

the input layer. One also finds that the inference times for float32 and int8 embeddings are almost identical, while the evaluation time for binary embeddings is slightly lower.

Considering the relative evaluation time for each task category separately, as shown in Figure 4.14, the same overall linear decrease in evaluation time with respect to the layer

number can be observed. This mirrors the findings of the original authors. Similar to the results for 1D Matryoshka representations, classification (Figure 4.14a) and clustering (Figure 4.14b) tasks are the only task categories where smaller Matryoshka sizes provide a noticeable speed-up in evaluation time.

To conclude **RQ-2D-2**, the evaluation time of the MBAI-2D model decreases linearly as the layer number decreases. There is no noticeable difference in evaluation time between float32 and int8 embeddings, while the evaluation time for binary embeddings is slightly lower. The choice of the Matryoshka size also does have an impact on the evaluation time, but it is not as significant as the choice of the layer used for inference.

Tables 31 to 38 in the appendix provide more information about the evaluation time of the model, both in relative and absolute terms as well as for each task and task category.

4.3.3 Performance, Memory Consumption, and Evaluation Time Trade-Off

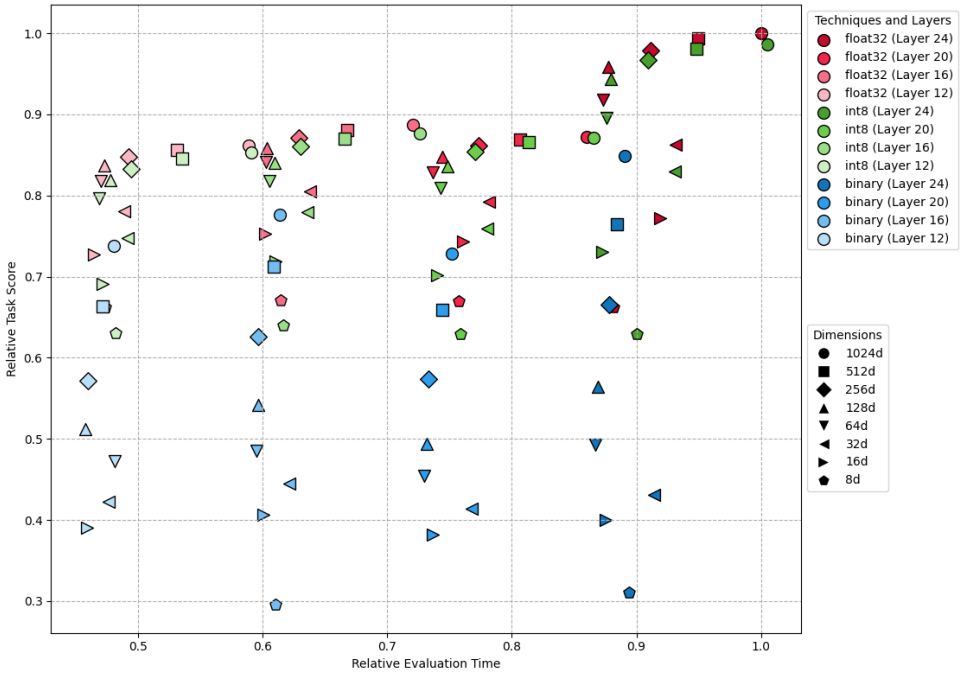


Figure 4.15: Trade-off between memory consumption, evaluation time, and performance of MBAI-2D, averaged over all selected MTEB tasks. The x-axis represents the relative evaluation time while the y-axis represents the relative performance. The reference points for both axes are the evaluation time and performance of the float32 embeddings at the full output size with the last layer used for inference and is located in the graph at (1,1). The color of the markers represent the quantization technique and inference layer, while the shape represents the embedding size of the respective MR-quantization combination.

Finally, the trade-off between performance, memory consumption, and evaluation time will be discussed to answer the research question **RQ-2D-3** and conclude the experiments on 2D Matryoshka representations paired with quantization.

Figure 4.15 shows the relationship between the performance and evaluation time of the MBAI-2D model averaged over all selected MTEB tasks. Similar to the plots where the memory consumption was plotted against the performance, the top-left corner of this plot represents the optimum. For each task, the reference point for both the evaluation time and task score is the evaluation time and task score of the float32 embeddings at the 24th layer with $m = 1024$.

The linear behavior in evaluation time, which was observed in Figure 4.13, is also reflected in this figure. For example within the interval of 0.87 and 1.0 on the x-axis, only results from the 24th layer are to be found. Between 0.7 and 0.87 on the x-axis, only results from the 20th layer are to be found, etc. This linearity which is present on the x-axis however is not present on the y-axis. For example, full dimensional float32 embeddings inferred from layer 12 achieve a relative performance of 86.2% while only requiring 59% of the evaluation time when compared to the reference point.

Finally, this can be combined with quantized, lower dimensional embeddings inferred from the 12th layer to achieve an even greater speed-up paired with a staggering reduction in memory consumption. As an example, when using int8 embeddings with $m = 128$ from the 12th layer, one would:

- Maintain 81.9% of the performance, averaged over all tasks
- Require 47.8% of the evaluation time
- Reduce the memory consumption by a factor of 32

compared to using float32 embeddings with $m = 1024$ drawn from the 24th layer of the model.

Task Category	**Relative Performance**	**Relative Evaluation Time**
Classification	0.728	0.427
Clustering	0.930	0.447
STS	0.830	0.513
Pair Classification	0.861	0.503
Retrieval	0.538	0.493
Re-Ranking	0.875	0.556
Summarization	0.883	0.501
Averaged over all tasks	**0.819**	**0.478**

Table 4.3: Relative performance and evaluation time across task categories, averaged over all tasks within the same category for the MBAI-2D model. The values are calculated using int8 embeddings with $m = 128$ from the 12th layer with the reference point being the float32 embeddings with $m = 1024$ from the 24th layer.

Due to formatting constraints, the plots showing the trade-off between performance, memory consumption, and evaluation time for each task category are shown in the appendix in Figure 16. However, Table 4.3 provides an overview of this trade-off when using the same

configuration as before (int8 embeddings with $m = 128$ from the 12th layer). The table shows that retrieval tasks provide the worst trade-off between performance and evaluation time, followed by the classification tasks. Opposing this are the clustering tasks, for which 93% of the performance can be maintained while only requiring 44.7% of the evaluation time.

To conclude **RQ-2D-3**, pairing 2D Matryoshka representations with quantization provides an opportunity to reduce the evaluation time and memory consumption of the models while maintaining a high performance. Given a moderate decrease in performance, the evaluation time can be reduced by a factor greater than 2 and the memory consumption by a factor of 32. The type of task has an impact on the trade-off between performance and evaluation time, with retrieval tasks providing the worst trade-off and clustering tasks providing the best.

Tables 31 to 46 in the appendix provide more information about the performance, memory consumption, and evaluation time of the model, both in relative and absolute terms as well as for each task and task category.

4.4 CO_2 Emissions

Experiment	Time (hours)	kWh	CO_2 eq. (kg)
Stella	27.3	1.91	0.64
Nomic	24.6	1.72	0.58
MBAI-1D	24.4	1.71	0.58
MBAI-2D	78.3	5.48	1.85
Total	**154.6**	**10.83**	**3.65**

Table 4.4: Estimated CO_2 equivalents emitted during the experiments. All computation was conducted on an NVIDIA T4 GPU which has a maximum power consumption of 70 watts. The grid carbon intensity used for the estimation is 0.337 kg CO_2 equivalents per kWh.

Table 4.4 shows the estimated CO_2 equivalents emitted during the experiments. As the experiments were split into multiple notebooks, each connected to an arbitrary runtime, the exact locations of the servers used are unknown. Google provides an overview of all their data centers and their energy sources. As the end-user has no influence on which data center the runtime connects, the average grid carbon intensity of all Google data centers was used to estimate the CO_2 equivalents emitted during the experiments. Averaging over all data centers, the grid carbon intensity is 0.337 kg CO_2 equivalents per kWh [Goo]. For all experiments, the NVIDIA T4 GPU was used, which has a maximum power consumption of 70 watts.

In total, the experiments emitted 3.65 kg CO_2 equivalents. As expected the tests for the MBAI-2D model emitted the most CO_2 equivalents, as all experiments were repeated for each layer considered for inference. According to the European Environment Agency, this is the equivalent of driving an average car for 33.7 kilometers. [Age23]

5 Conclusion

5.1 Summary

Chapter 4 has shown that Matryoshka representations and quantization can be combined to achieve compelling efficiency versus performance trade-offs.

RQ-1D-1 has shown that the performance does degrade when combining 1D Matryoshka representations with quantization compared to using full precision, high dimensional embeddings. For each Matryoshka size, the int8 embeddings achieved almost the same performance as the float32 ones, while the binary embeddings performed significantly worse.

Considering the evaluation time, addressed in **RQ-1D-2**, it was found that using smaller Matryoshka sizes can reduce inference time to some extent. Regarding quantization, it can be said that int8 embeddings only provide a small speed-up compared to float32 ones, while binary embeddings are the fastest across all models and tasks.

RQ-1D-3 has shown that 1D Matryoshka representations paired with quantization offer an excellent trade-off between memory usage and performance. Especially, the combination of int8 embeddings with a Matryoshka size 4 to 8 times smaller compared to the full dimensional embeddings seems to provide a good trade-off. Using the MBAI-1D model as an example, the memory footprints can be reduced by a factor of 16 while only losing 4% in relative performance.

Moving to the combination of 2D Matryoshka representations with quantization, **RQ-2D-1** has shown that the performance does degrade when inferring the embeddings from internal layers compared to the last layer. Also, for embeddings grasped from intermediate layers, the performance of int8 embeddings is almost the same as that of the float32 ones, with the binary embeddings performing significantly worse. The degradation in performance when lowering the Matryoshka size behaves similarly for all layers and quantization techniques investigated.

RQ-2D-2 confirmed that the evaluation time with respect to the layer chosen for inference decreases in a linear fashion. The choice of the Matryoshka size and quantization technique used does have an impact on the inference time, but it is not as significant as the choice of the layer.

Finally, **RQ-2D-3** investigated both the speed-up in inference time and the reduction in memory usage, while considering the loss in performance. These results have shown that the user can decide whether a more efficient but less accurate or a more accurate but less efficient model is desired, exemplified by Table 5.1. Should one only be interested in a reduction of memory usage without a significant loss in performance, 1D MR, meaning using the last layer of the model with a medium Matryoshka size (e.g. $m = 128$ for the

Dimensionality	Quantization	Inference layer	Bits reduced	Inference Time (relative)	Performance (relative)
1024	float32	24	1×	100 %	100 %
128	int8	24	32×	88.0 %	94.4 %
128	int8	12	32×	47.8 %	81.9 %
512	binary	12	64×	47.1 %	66.3 %

Table 5.1: Selected results from the MBAI-2D model, highlighting the trade-offs between memory usage, inference time and performance.

MBAI-1D model) and int8 embeddings is a good choice. The memory can be reduced by a factor of 32 while only losing 5.6% in relative performance and needing 12 % less time for inference. When a significant reduction in inference time is crucial as well, using an intermediate layer (2D MR) with a smaller Matryoshka size and int8 embeddings is a good choice. The table shows that using the same configuration as before with an internal layer (12) instead of the last layer (24) results in a 52.2% lower inference time while losing 18.1 % in relative performance compared to the full dimensional float32 embeddings from the last layer. Note that binary embeddings can provide even more efficiency but at the cost of sacrificing more performance. The results discussed only serve as examples and one can select the desired trade-off between memory consumption, inference time, and performance based on the requirements of the task at hand. Thus, Matryoshka representations and quantization can be combined to achieve compelling efficiency gains while maintaining high performance at downstream tasks.

The appendix provides the entirety of the underlying data for the results presented in this thesis.

5.2 Interpretation

Considering the combination of 1D Matryoshka representations with quantization, the results indicate that a huge decrease in memory usage can be achieved with only a small loss in performance. Unless the highest performance is required, the combination of 1D MRs and quantization is a good choice, especially in large-scale applications where memory usage is a concern. As popular vector storage providers partially charge users based on the total storage used, this would mean that for enterprises the cost of storing embeddings could be reduced by a factor of 16 when following the same example from the previous section [Pin24, Clo24]. For 2D Matryoshka representations, which additionally offer a speed-up in inference time, the drops in performance tend to be more significant. A possible solution for this is presented in Subsection 5.3.1. However, as the costs for computations usually also follow a pay-as-you-go pricing model, saving 50 % in inference time paired with the reduction in memory usage could be very appealing for many organizations [Ser24, Azu24].

Implicitly, both the combination of 1D and 2D Matryoshka representations with quantization lower the costs of AI systems that heavily rely on embeddings like Retrieval Augmented Generation pipelines. This in return reduces the entry barriers for smaller

organizations or even single individuals who want to build AI systems but are limited by the costs of computation.

Ultimately, the true power of this joint approach lies in the flexibility it offers. In times when Large Language Models and especially RAG pipelines are used in more and more applications, the ability to adapt the system to the requirements of the application in real-time and with no additional cost can be seen as a huge advantage. Utilizing this approach is interesting for many different kinds of people, ranging from ones who want to build AI that minimizes the impact on the environment to startups who want to avoid their Amazon Web Services (AWS) bill skyrocketing as they undergo a rapid user growth overnight.

5.3 Future Work and Limitations

This section will suggest some directions for future work and discuss some limitations of the research presented in this thesis.

5.3.1 Adaptive use of Matryoshka Representations

As was suggested by the authors of the 1D Matryoshka representations paper, an **adaptive** approach could be deployed [KBR+22]. This approach can be extended to the combination of 2D Matryoshka representations with quantization and would utilize two versions of the model: one that prioritizes efficiency at the cost of performance, and another that sacrifices efficiency for higher performance. For instance in Retrieval Augmented Generation (RAG) pipelines, one could utilize the concept of re-ranking, initially proposed in 2022 to improve the performance of the system [GRC+22]. However, this approach can also be utilized to increase the efficiency of the pipeline. The flexibility of both Matryoshka representations and quantization is perfectly suited for this and would only require the training of a single model. In the first step, the model with a lower number of layers, a small embedding size, and quantization could be used to get a preselection of the most relevant documents. Following this, the same model with a higher number of layers, a larger embedding size, and without quantization could be utilized to generate the final output. For really large-scale applications, one could even add more steps between the two extremes to further optimize the efficiency while assuring high performance.

5.3.2 Semantic Compression

Recall from Section 1.2 that in a RAG pipeline documents are often divided into smaller chunks which can then be processed by the embedding model. An alternative to this is truncation where everything that exceeds the maximum token length is simply ignored. However since the last approach can result in a loss of information, chunking is preferred in practice. A problem with chunking is that it requires multiple inference calls to the models per document, one for each chunk. Suppose a document consists of n tokens, a model with the context size m will require at least $\lceil \frac{n}{m} \rceil$ inference calls. Also, in a RAG

pipeline, the more chunks there are, the more comparison between the query and the chunks are required. Thus, reducing the size of the content which has to be chunked and embedded can both reduce the number of inference calls and the number of comparisons required.

An interesting approach to achieve this is using Semantic Compression [GSS+23]. The authors argue that a Large Language Model like GPT-4 can be used to compress the documents such that the same information can be expressed using less tokens. They have shown that semantic compression using an LLM manages to achieve a compression ratio of up to 77.2 % on average, while Zlib, a traditional compression approach, only achieves a compression ratio of 48.8 %.

A compression rate of 77.2% means that the number of tokens in the compressed document is reduced to only 22.8% of the tokens in the original document. This could reduce the number of inference calls to the embedding model by a factor of 4.4, as well as the number of comparisons required in a RAG pipeline by the same factor **for each query**. To exemplify how the result of semantic compression looks like, the first paragraph of this section was compressed using the encoding query for GPT-4 provided in the original paper (refer to Section G. in the appendix for the prompt used):

"RAG pipelines split documents into smaller chunks for embedding, avoiding information loss from truncation. Chunking, though requiring multiple inferences per document, prevents data loss. With n tokens and model context size m, $\lceil \frac{n}{m} \rceil$ calls are needed. Reducing content size decreases both inference calls and query-chunk comparisons."

Analyzing the performance-compute trade-off of semantic compression is an interesting direction for future work. Furthermore, if these results look promising, combining semantic compression with Matryoshka representations and quantization could be a powerful approach to further reduce the costs of AI systems that rely on embeddings.

5.3.3 Expanding the Scope of Experiments

Lastly, it is important to consider that the results presented in this thesis are based on a limited number of experiments. In total, only four models were investigated, with only one supporting 2D Matryoshka representations. Additionally, only a limited number of tasks were considered due to the computational resources available within the scope of this thesis. Future research could investigate a broader range of models and tasks to determine whether the results presented in this thesis are generalizable to other models and the entire MTEB suite. Furthermore, this thesis has only tested the most common quantization techniques in the context of embeddings. Future work could also analyze the performance-compute trade-offs when using integers with less than 8 bits or lower-precision floating-point numbers [KBR+24].

Appendices

This appendix contains additional figures and tables for the thesis. Note that some tables are very small and might be hard to read in the printed version of the thesis. It is recommended to view the tables in the digital version of the thesis, available on GitHub alongside the other resources: https://github.com/KaiPonel/MRLQuantization.

Shortening	Task Name
q	Datatype of Embedding (Quantization Technique)
m	Dimensionality of Embedding (Matryoshka size)
l	Inference Layer Number
f32	32-Bit Floating-Point
int	8-Bit Integer
bin	Binary
bits req.	Bits Required to store embedding
CLA-1	Banking77Classification
CLA-2	EmotionClassification
CLA-3	TweetSentimentExtractionClassification
CLA-4	AmazonCounterfactualClassification
CLA-5	MassiveIntentClassification
CLA-6	MassiveScenarioClassification
CLA-7	MTOPDomainClassification
CLA-8	MTOPIntentClassification
CLU-1	ArXivHierarchicalClusteringP2P
CLU-2	ArXivHierarchicalClusteringS2S
CLU-3	BiorxivClusteringP2P.v2
CLU-4	BiorxivClusteringS2S.v2
CLU-5	MedrxivClusteringP2P.v2
CLU-6	MedrxivClusteringS2S.v2
CLU-7	RedditClustering.v2
CLU-8	StackExchangeClustering.v2
CLU-9	StackExchangeClusteringP2P.v2
CLU-10	TwentyNewsgroupsClustering.v2
STS-1	BIOSSES
STS-2	SICK-R
STS-3	STS12
STS-4	STS13
STS-5	STS14
STS-6	STS15
STS-7	STS16
STS-8	STSBenchmark
STS-9	STS17
STS-10	STS22
PCL-1	SprintDuplicateQuestions
PCL-2	TwitterSemEval2015
PCL-3	TwitterURLCorpus
RET-1	ArguAna
RET-2	CQADupstackWebmastersRetrieval
RET-3	NFCorpus
RER-1	AskUbuntuDupQuestions
RER-2	MindSmallReranking
RER-3	StackOverflowDupQuestions
SUM-1	SummEval

Table 2: Shortening and Task Name Mapping

A. Model Selection

Table 3: Top-100 models on the MTEB leaderboard as of 03.08.2024 at 12:00pm. Model size is measured in million parameters. Dimensions refers to the default embedding size of the model.

Rank	Model	Model Size	Memory Usage	Dimensions	Max Tokens	Avg. Score
1	bge-en-icl	7111	26.49	4096	32768	71.67
2	stella_en_1.5B_v5	1543	5.75	8192	131072	71.19
3	SFR-Embedding-2_R	7111	26.49	4096	32768	70.31
4	gte-Qwen2-7B-instruct-Q4_K_M-GGUF					70.24
5	gte-Qwen2-7B-instruct	7613	28.36	3584	131072	70.24
6	stella_en_400M_v5	435	1.62	8192	8192	70.11
7	bge-multilingual-gemma2	9242	34.43	3584	8192	69.88
8	NV-Embed-v1	7851	29.25	4096	32768	69.32
9	voyage-large-2-instruct			1024	16000	68.23
10	Linq-Embed-Mistral	7111	26.49	4096	32768	68.17
11	SFR-Embedding-Mistral	7111	26.49	4096	32768	67.56
12	gte-Qwen1.5-7B-instruct	7099	26.45	4096	32768	67.34
13	gte-Qwen2-1.5B-instruct-Q4_0-GGUF					67.16
14	gte-Qwen2-1.5B-instruct	1776	6.62	1536	131072	67.16
15	voyage-lite-02-instruct	1220	4.54	1024	4000	67.13
16	GritLM-7B	7242	26.98	4096	32768	66.76
17	e5-mistral-7b-instruct	7111	26.49	4096	32768	66.63
18	google-gecko.text-embedding-preview-0409	1200	4.47	768	2048	66.31
19	TDTE					65.96
20	GritLM-8x7B	46703	173.98	4096	32768	65.66
21	gte-large-en-v1.5	434	1.62	1024	8192	65.39
22	LLM2Vec-Meta-Llama-3-supervised	7505	27.96	4096	8192	65.01
23	LLM2Vec-Mistral-supervised	7111	26.49	4096	32768	64.8
24	echo-mistral-7b-instruct-lasttoken	7111	26.49	4096	32768	64.68

Continued on next page

Rank	Model	Model Size	Memory Usage	Dimensions	Max Tokens	Avg. Score
25	mxbai-embed-large-v1	335	1.25	1024	512	64.68
26	UAE-Large-V1	335	1.25	1024	512	64.64
27	text-embedding-3-large			3072	8191	64.59
28	voyage-lite-01-instruct			1024	4000	64.49
29	Cohere-embed-english-v3.0			1024	512	64.47
30	multilingual-e5-large-instruct	560	2.09	1024	514	64.41
31	google-gecko-256.text-embedding-preview-0409	1200	4.47	256	2048	64.37
32	GIST-large-Embedding-v0	335	1.25	1024	512	64.34
33	bge-large-en-v1.5	335	1.25	1024	512	64.23
34	b1ade-embed	335	1.25	1024	512	64.21
35	MUG-B-1.6	335	1.25	1024	512	64.2
36	LLM2Vec-Llama-2-supervised	6607	24.61	4096	4096	64.14
37	gte-base-en-v1.5	137	0.51	768	8192	64.11
38	Cohere-embed-multilingual-v3.0			1024	512	64.01
39	GIST-Embedding-v0	109	0.41	768	512	63.71
40	bge-base-en-v1.5	438	1.63	768	512	63.56
41	privacy_embedding_rag_10k_base_final	109	0.41	768	512	63.55
42	privacy_embedding_rag_10k_base_12_final	109	0.41	768	512	63.55
43	privacy_embedding_rag_10k_base_15_final	109	0.41	768	512	63.55
44	privacy_embedding_rag_10k_base_checkpoint_2	109	0.41	768	512	63.55
45	ember-v1	335	1.25	1024	512	63.54
46	sf_model_e5	335	1.25	1024	512	63.34
47	mxbai-embed-2d-large-v1	335	1.25	1024	512	63.25
48	gte-large	335	1.25	1024	512	63.13
49	NoInstruct-small-Embedding-v0	33	0.12	384	512	63.12
50	GIST-small-Embedding-v0	33	0.12	384	512	62.72
51	stella-base-en-v2	55	0.2	768	512	62.61
52	gte-base	109	0.41	768	512	62.39
53	UniVaR-lambda-80	137	0.51	768	8192	62.39

Continued on next page

Rank	Model	Model Size	Memory Usage	Dimensions	Max Tokens	Avg. Score
54	UniVaR-lambda-20	137	0.51	768	8192	62.39
55	UniVaR-lambda-1	137	0.51	768	8192	62.39
56	nomic-embed-text-v1	137	0.51	768	8192	62.39
57	UniVaR-lambda-5	137	0.51	768	8192	62.39
58	nomic-embed-text-v1.5	137	0.51	768	8192	62.28
59	text-embedding-3-small			1536	8191	62.26
60	e5-large-v2	335	1.25	1024	512	62.25
61	bge-small-en-v1.5	33	0.12	384	512	62.17
62	Cohere-embed-english-light-v3.0					62.01
63	text-embedding-3-large-256			256	8191	62.00
64	nomic-embed-text-v1.5-512	138	0.51	512	8192	61.96
65	LLM2Vec-Sheared-Llama-supervised	1280	4.77	2048	4096	61.85
66	instructor-xl	1241	4.62	768	512	61.79
67	instructor-large	335	1.25	768	512	61.59
68	e5-base-v2	109	0.41	768	512	61.50
69	e5-base-4k	112	0.42	768	4096	61.50
70	multilingual-e5-large	560	2.09	1024	514	61.50
71	e5-large	335	1.25	1024	512	61.42
72	nomic-embed-text-v1-ablated	137	0.51	768	8192	61.36
73	gte-small	33	0.12	384	512	61.36
74	nomic-embed-text-v1.5-256	138	0.51	256	8192	61.04
75	text-embedding-ada-002			1536	8191	60.99
76	udever-bloom-7b1	7069	26.33	4096	2048	60.63
77	e5-base	109	0.41	768	512	60.44
78	jina-embeddings-v2-base-en	137	0.51	768	8192	60.38
79	Titan-text-embeddings-v2					60.37
80	snowflake-arctic-embed-m-long	137	0.51	768	8192	60.09
81	Cohere-embed-multilingual-light-v3.0			384	512	60.08
82	e5-small-v2	33	0.12	384	512	59.93

Continued on next page

Rank	Model	Model Size	Memory Usage	Dimensions	Max Tokens	Avg. Score
83	udever-bloom-3b	3003	11.19	2560	2048	59.86
84	nomic-embed-text-v1-unsupervised	137	0.51	768	8192	59.85
85	snowflake-arctic-embed-l	334	1.24	1024	512	59.84
86	snowflake-arctic-embed-m	109	0.41	768	512	59.79
87	instructor-base	110	0.41	768	512	59.54
88	sentence-t5-xxl	4865	18.12	768	512	59.51
89	multilingual-e5-base	278	1.04	768	514	59.45
90	nomic-embed-text-v1.5-128	138	0.51	128	8192	59.34
91	XLM-3B5-embedding					59.29
92	GIST-all-MiniLM-L6-v2	23	0.08	384	512	59.00
93	gtr-t5-xxl	4865	18.12	768	512	58.97
94	snowflake-arctic-embed-s	33	0.12	384	512	58.94
95	SGPT-5.8B-weightedmean-msmarco-specb-bitfit	5874	21.88	4096	2048	58.93
96	e5-small	33	0.12	384	512	58.89
97	gte-tiny	23	0.08	384	512	58.69
98	gtr-t5-xl	1240	4.62	768	512	58.42
99	udever-bloom-1b1			1536	2048	58.29
100	gtr-t5-large	168	0.63	768	512	58.28

Table 4: Top-100 models on the MTEB leaderboard as of 03.08.2024 at 12:00pm which do not exceed 1 billion parameters. Model size is measured in million parameters.

Rank	Model	Model Size	Is Opensource?	Supports MRL?	Additional Notes
6	**stella_en_400M_v5**	435	y	y	
21	gte-large-en-v1.5	434	y	n	
25	**mxbai-embed-large-v1**	335	y	y	
26	UAE-Large-V1	335	y	n	
30	multilingual-e5-large-instruct	560	y	n	
32	GIST-large-Embedding-v0	335	y	n	
33	bge-large-en-v1.5	335	y	n	
34	b1ade-embed	335	y	n	
35	MUG-B-1.6	335	y	n	
37	gte-base-en-v1.5	137	y	n	
38	Cohere-embed-multilingual-v3.0		n	n	
40	bge-base-en-v1.5	438	y	n	
41	privacy_embedding_rag_10k_base_final	109	y	n	
42	privacy_embedding_rag_10k_base_12_final	109	y	n	
43	privacy_embedding_rag_10k_base_15_final	109	y	n	
44	privacy_embedding_rag_10k_base_checkpoint_2	109	y	n	
45	ember-v1	335	y	n	
46	sf_model_e5	335	y	n	
47	**mxbai-embed-2d-large-v1**	335	y	y	2D-MRL Model
48	gte-large	335	y	n	
49	NoInstruct-small-Embedding-v0	33	y	n	
50	GIST-small-Embedding-v0	33	y	n	
51	stella-base-en-v2	55	y	n	
52	gte-base	109	y	n	
53	UniVaR-lambda-80	137	y	n	
54	UniVaR-lambda-20	137	y	n	

Continued on next page

Rank	Model	Model Size	Is Opensource?	Supports MRL?	Additional Notes
55	UniVaR-lambda-1	137	y	n	
56	nomic-embed-text-v1	137	y	n	
57	UniVaR-lambda-5	137	y	n	
58	**nomic-embed-text-v1.5**	137	y	y	
60	e5-large-v2	335	y	n	
61	bge-small-en-v1.5	33	y	n	
64	nomic-embed-text-v1.5-512	138	y	y	Same model as 58.
67	instructor-large	335	y	n	
68	e5-base-v2	109	y	n	
69	e5-base-4k	112	y	n	
70	multilingual-e5-large	560	y	n	
71	e5-large	335	y	n	
72	nomic-embed-text-v1-ablated	137	y	n	
73	gte-small	33	y	n	
74	nomic-embed-text-v1.5-256	138	y	y	Same model as 58.
77	e5-base	109	y	n	
78	jina-embeddings-v2-base-en	137	y	n	
80	snowflake-arctic-embed-m-long	137	y	n	
82	e5-small-v2	33	y	n	
84	nomic-embed-text-v1-unsupervised	137	y	n	
85	snowflake-arctic-embed-l	334	y	n	
86	snowflake-arctic-embed-m	109	y	n	
87	instructor-base	110	y	n	
89	multilingual-e5-base	278	y	n	
90	nomic-embed-text-v1.5-128	138	y	y	Same model as 58.
92	GIST-all-MiniLM-L6-v2	23	y	n	
94	snowflake-arctic-embed-s	33	y	n	
96	e5-small	33	y	n	
97	gte-tiny	23	y	n	

Continued on next page

Rank	Model	Model Size	Is Opensource?	Supports MRL?	Additional Notes
100	gtr-t5-large	168	y	n	

B. Additional Figures from Literature

B.1 MRL Figures

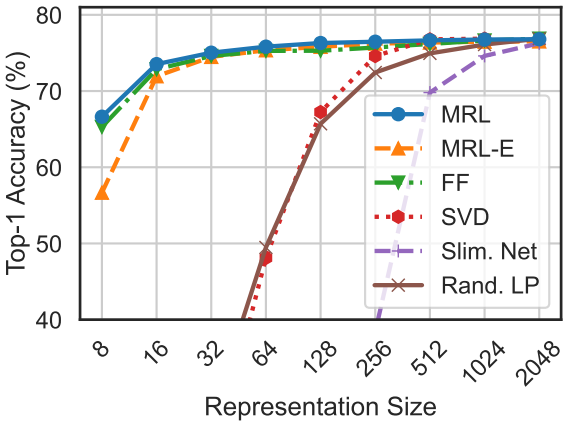


Figure 1: ImageNet-1K linear classification accuracy of ResNet50 models. MRL is as accurate as the independently trained FF models for every representation size. Figure and caption excerpted from the original paper [KBR+22].

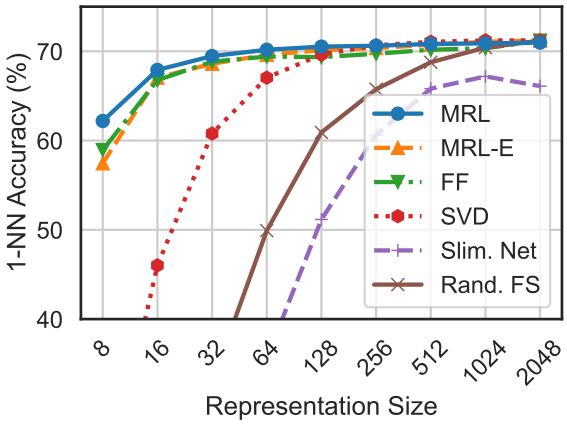


Figure 2: ImageNet-1K 1-NN accuracy of ResNet50 models measuring the representation quality for downstream task. MRL outperforms all the baselines across all representation sizes. Figure and caption excerpted from the original paper [KBR+22].

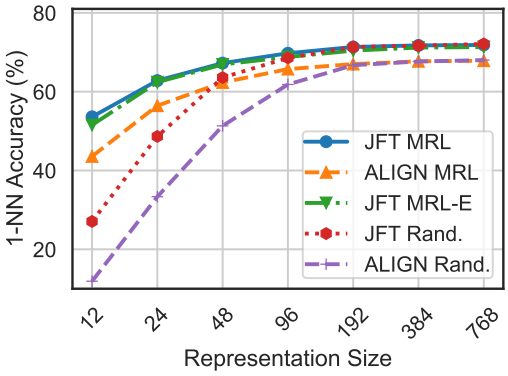


Figure 3: ImageNet-1K 1-NN accuracy for ViT-B/16 models trained on JFT-300M & as part of ALIGN. MRL scales seamlessly to web-scale with minimal training overhead. Figure and caption excerpted from the original paper [KBR+22].

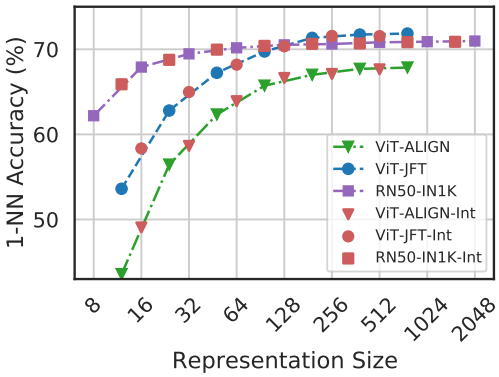


Figure 4: Despite optimizing MRL only for $O(\log(d))$ dimensions for ResNet50 and ViT-B/16 models; the accuracy in the intermediate dimensions shows interpolating behavior. Figure and caption excerpted from the original paper [KBR+22].

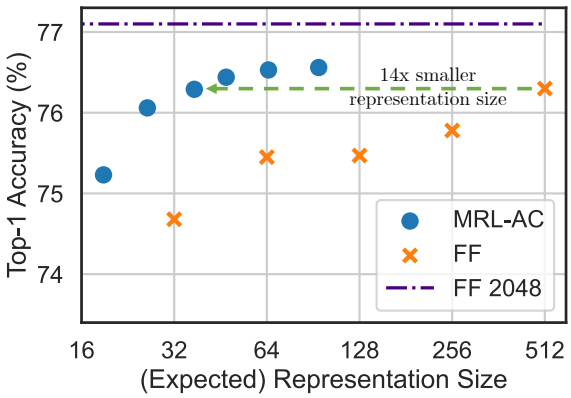


Figure 5: Adaptive classification on MRL ResNet50 using cascades results in 14× smaller representation size for the same level of accuracy on ImageNet-1K (∼ 37 vs 512 dims for 76.3%). Figure and caption excerpted from the original paper [KBR+22].

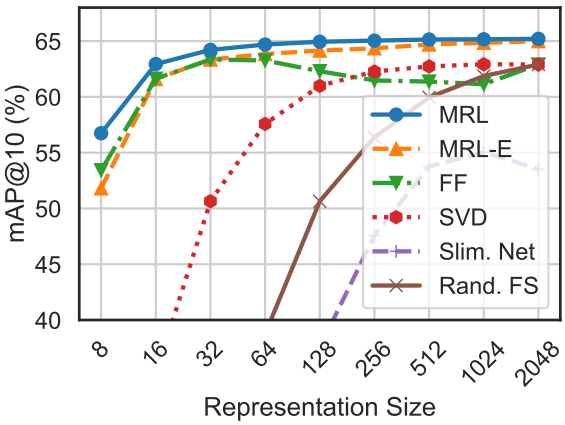
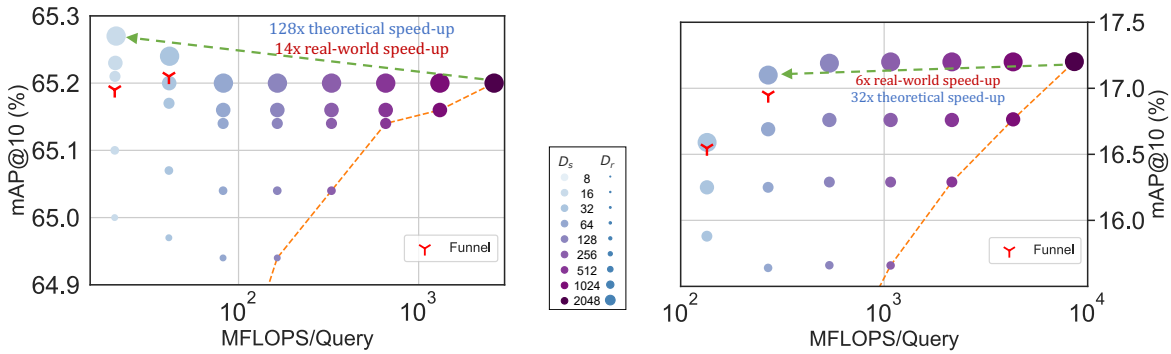


Figure 6: mAP@10 for Image Retrieval on ImageNet-1K with ResNet50. MRL consistently produces better retrieval performance over the baselines across all the representation sizes. Figure and caption excerpted from the original paper [KBR+22].



(a) ImageNet-1K

(b) ImageNet-4K

Figure 7: The trade-off between mAP@10 vs MFLOPs/Query for Adaptive Retrieval (AR) on ImageNet-1K (left) and ImageNet-4K (right). Every combination of D_s & D_r falls above the Pareto line (orange dots) of single-shot retrieval with a fixed representation size while having configurations that are as accurate while being up to 14× faster in real-world deployment. Funnel retrieval is almost as accurate as the baseline while alleviating some of the parameter choices of Adaptive Retrieval. Figure and caption excerpted from the original paper [KBR+22].

B.2 2DMSE Figures

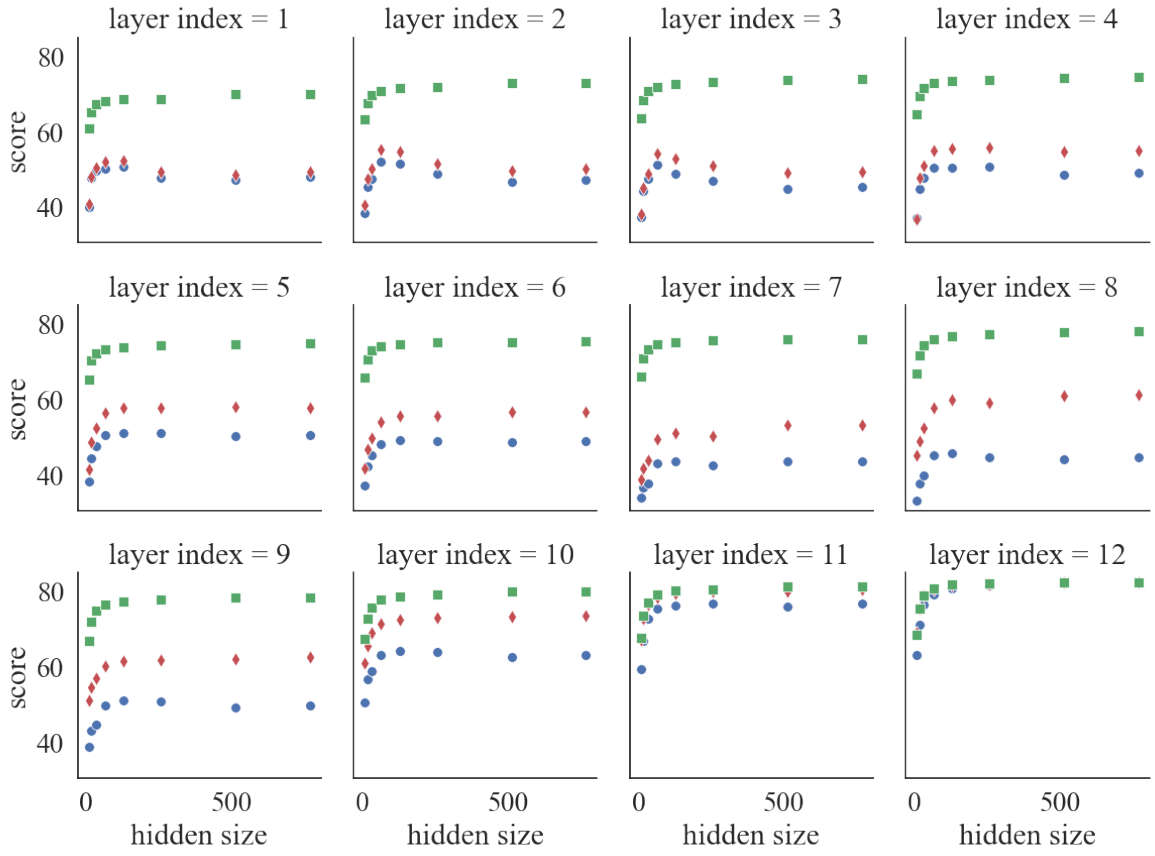
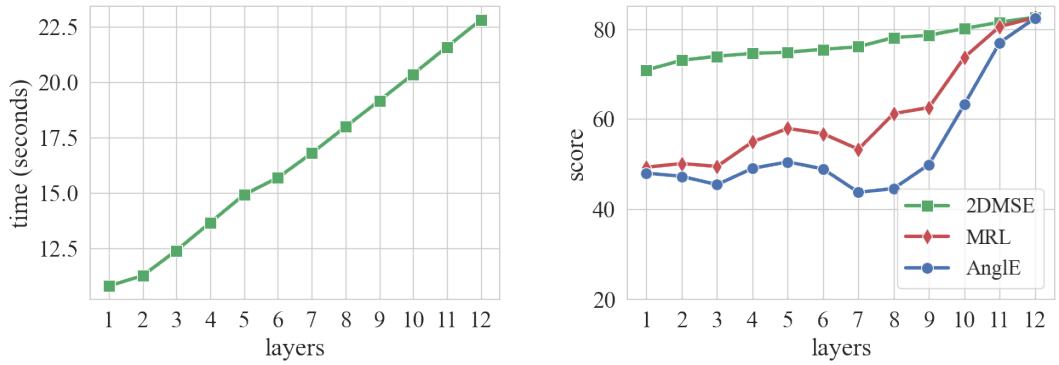


Figure 8: Results of the STS benchmark with a cascade of hidden sizes: $8 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 768$ from BERT_{base}. The score represents the average Spearman's correlation. BERT_{base} serves as the backbone for all models. The blue • indicates the results of sentence embeddings from AnglE without any scalable sentence embedding learning. The red ♦ represents the results of matryoshka sentence embeddings. The green ■ denotes the results of our proposed 2D Matryoshka Sentence Embeddings (2DMSE). The layer index $= i$ denotes the i-th attention layer. Figure and caption excerpted from the original paper [LLL+24].



(a) Inference time *vs* number of layers. **(b)** Score on STS *vs* number of layers.

Figure 9: Subfigure (a) illustrates the time taken to use embeddings from different layers to encode the entire STS benchmarks. Subfigure (b) displays the average Spearman's correlation scores of different layers. Both (a) and (b) use an embedding size of 768 and the standard STS benchmark dataset. Figure and caption excerpted from the original paper [LLL+24].

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
GloVe [RG19]	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
USE [RG19]	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
SBERT [RG19]	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SimCSE [GYC21]	75.30	84.67	80.19	85.40	80.82	84.25	80.39	81.57
AnglE [LL23]	75.09	85.56	80.66	86.44	**82.47**	85.16	**81.23**	82.37
MRL ($d = 768$) $\star$	**75.72**	**86.79**	81.89	**86.91**	81.74	85.50	79.44	82.57
2DMSE ($n = 12$, $d = 768$)	75.00	86.69	**82.30**	86.50	82.09	**85.79**	80.18	**82.65**

Table 5: Full-capacity sentence embedding performance on the standard STS benchmark. Results $\star$ denote our implementation. BERT_{base} serves as the backbone for all models. Figure and caption excerpted from the original paper [LLL+24].

Model	Avg. Spearman's Correlation
$n = 12$, $d = 768$	
2DMSE	**82.65**
w/o alignment $\mathcal{L}_{align}$	82.57 (−0.08)
w/o last layer $\mathcal{L}_N^D$	81.31 (−1.34)
$n = 8$, $d = 512$	
2DMSE	**78.02**
w/o alignment $\mathcal{L}_{align}$	77.94 (−0.08)
w/o last layer $\mathcal{L}_N^D$	76.52 (−1.50)
$n = 6$, $d = 384$	
2DMSE	**75.21**
w/o alignment $\mathcal{L}_{align}$	75.08 (−0.13)
w/o last layer $\mathcal{L}_N^D$	74.98 (−0.23)
$n = 4$, $d = 256$	
2DMSE	**73.93**
w/o alignment $\mathcal{L}_{align}$	73.69 (−0.24)
w/o last layer $\mathcal{L}_N^D$	73.00 (−0.93)

Table 6: Ablation study results of 2DMSE on the standard STS benchmark using BERT_{base}. n denotes the number of Transformer layers, and d stands for the embedding dimensions. Figure and caption excerpted from the original paper [LLL+24].

C. mxbai-embed-large-v1

This section contains additional figures and tables for the MBAI-1D model.

q	m	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	1024	61.98	18.26	27.28	10.67	42.54	26.21	17.41	68.60	14.53	9.32	163.71	18.00	129.11	15.91	233.31	223.57	185.62	11.57	2.31	56.65	24.57	9.96	26.52	21.28	9.19	9.94	1.42	30.30	42.98	55.13	236.84	469.87	741.35	294.39	32.99	294.05	313.21	39.10
f32	512	54.38	17.24	26.87	10.95	37.55	15.32	16.68	56.04	10.06	4.88	161.58	17.82	126.80	15.73	231.72	219.14	177.20	9.98	2.26	56.34	24.60	9.81	26.50	21.35	9.14	9.94	1.41	30.64	39.05	54.40	236.42	458.81	730.16	293.33	33.08	287.08	314.06	39.10
f32	256	50.33	16.98	26.18	10.67	25.08	14.30	15.81	50.18	7.71	4.43	160.39	15.10	124.23	12.48	228.31	215.48	172.08	10.01	2.32	56.33	24.49	9.87	26.41	21.29	9.20	9.91	1.41	30.44	39.09	54.32	236.47	458.58	730.34	292.64	33.15	281.41	314.05	39.04
f32	128	38.79	16.90	26.37	10.77	23.86	13.86	16.04	37.63	6.87	2.61	159.46	15.33	123.31	11.77	227.32	212.13	169.66	9.41	2.23	55.77	24.78	9.84	26.24	21.34	9.25	9.89	1.45	30.33	38.56	54.33	235.67	458.56	730.39	293.15	33.27	275.89	313.79	39.06
f32	64	38.04	16.85	26.45	10.84	23.34	13.55	15.75	36.54	6.60	2.22	159.61	14.24	122.98	12.26	227.95	211.36	166.29	8.33	2.23	55.39	24.82	9.85	26.41	21.28	9.18	9.91	1.41	30.62	36.93	54.37	235.64	465.25	741.75	298.34	33.83	283.23	333.10	40.91
f32	32	37.44	17.07	26.19	10.68	22.87	13.33	15.28	35.39	6.47	2.33	158.95	14.10	123.30	11.69	227.99	211.15	166.12	8.86	2.22	55.65	24.82	9.92	26.31	21.24	9.22	9.96	1.41	30.35	36.96	53.89	237.25	468.68	742.06	298.14	34.09	284.72	331.88	40.97
f32	16	36.47	16.68	25.54	10.63	22.20	13.33	15.16	33.95	6.35	1.94	158.90	13.97	122.90	11.65	226.98	210.39	166.11	8.58	2.26	55.63	24.93	9.94	26.21	21.16	9.27	9.93	1.46	30.26	37.01	54.20	235.43	469.49	741.94	297.75	33.83	280.40	331.17	40.96
f32	8	36.19	16.58	25.45	10.55	21.97	13.43	15.37	33.54	5.73	1.85	158.89	14.00	122.02	10.65	225.98	211.73	166.00	7.93	2.22	55.43	24.88	9.83	26.32	21.33	9.19	9.97	1.41	30.76	36.92	53.76	236.34	468.88	741.17	298.19	33.89	277.67	331.95	40.83
int	1024	64.05	17.85	27.01	10.85	45.47	26.17	16.74	70.43	12.55	8.88	164.43	19.88	130.29	16.84	232.87	223.47	185.83	13.62	2.29	57.22	24.80	9.92	26.72	21.31	9.18	9.97	1.43	30.67	46.07	55.10	238.55	413.14	653.41	264.14	29.59	274.76	289.77	35.82
int	512	55.05	17.54	26.62	10.77	37.80	14.85	16.72	57.25	10.64	5.07	161.29	17.72	127.16	15.50	231.59	218.32	175.31	10.30	2.28	56.58	24.60	9.90	26.53	21.24	9.21	9.93	1.41	30.48	39.80	54.40	236.66	412.53	652.47	263.96	29.73	267.62	289.14	35.77
int	256	50.65	16.86	26.48	11.04	25.21	14.40	16.07	50.75	8.91	4.31	159.81	15.23	123.87	12.27	228.02	212.87	168.82	9.20	2.31	56.20	24.73	9.91	26.43	21.23	9.20	9.92	1.46	30.43	39.39	54.37	236.57	412.79	653.14	264.33	29.59	261.74	289.11	35.71
int	128	38.96	17.24	26.51	10.80	24.15	13.57	15.60	38.22	7.36	3.32	159.80	14.62	123.46	11.93	228.84	212.04	166.97	8.64	2.23	55.51	24.84	9.90	26.21	21.25	9.25	9.96	1.43	30.32	38.60	53.97	236.37	412.67	653.30	264.55	29.70	260.12	288.84	35.68
int	64	37.73	16.86	26.14	10.96	23.32	14.03	16.00	36.03	6.55	2.39	159.27	14.18	122.94	11.28	226.83	211.26	165.12	8.37	2.25	55.59	24.77	9.87	26.22	21.29	9.25	9.93	1.43	30.34	37.31	54.10	235.77	413.02	652.97	264.21	29.64	260.84	289.24	35.62
int	32	37.78	17.01	26.64	11.02	23.01	13.47	15.72	36.00	6.51	1.92	159.27	14.10	122.65	11.16	227.40	210.62	165.07	8.78	2.24	55.33	24.82	9.90	26.20	21.28	9.27	9.92	1.41	30.28	37.78	54.26	235.66	412.87	653.31	264.72	29.61	261.13	289.14	35.81
int	16	37.03	17.03	25.83	10.76	22.97	13.44	15.61	35.13	6.28	2.17	158.77	14.15	122.51	11.01	226.20	211.01	164.93	8.08	2.28	55.31	24.80	9.83	26.18	21.25	9.21	9.91	1.41	30.21	36.72	53.93	236.52	412.53	653.00	264.36	29.56	256.26	288.88	35.62
int	8	36.87	16.54	25.53	10.68	22.73	13.90	16.12	35.01	6.97	1.67	158.65	13.80	122.57	10.79	227.20	209.99	165.20	8.49	2.23	55.68	24.86	9.83	26.33	21.29	9.20	9.94	1.41	30.63	37.60	54.12	235.59	412.78	653.69	264.65	29.59	255.88	288.63	35.72
bin	1024	38.57	17.11	26.07	10.70	24.04	13.83	15.47	37.95	8.78	2.92	159.61	14.70	123.54	12.20	228.88	212.11	166.53	8.80	2.23	55.75	24.82	9.91	26.35	21.14	9.19	9.94	1.42	30.47	38.91	54.33	236.16	412.63	653.17	264.23	29.61	258.72	289.21	35.66
bin	512	37.86	16.88	26.57	10.89	23.29	13.47	15.69	36.45	6.44	2.17	159.91	14.36	123.19	11.34	228.08	211.15	166.16	9.17	2.22	55.46	24.83	9.90	26.40	21.21	9.20	9.96	1.41	30.41	38.23	54.27	236.07	412.73	653.40	264.47	29.60	258.85	288.85	35.64
bin	256	37.18	17.02	26.41	10.74	22.81	13.34	15.40	35.71	7.07	2.09	159.19	14.05	122.83	11.18	227.53	211.13	164.80	8.19	2.22	55.38	24.77	9.86	26.19	21.22	9.24	9.94	1.41	30.31	37.93	54.27	235.63	412.62	653.38	264.56	29.52	258.77	288.83	35.71
bin	128	37.28	16.52	25.95	10.65	22.67	13.32	15.49	35.42	7.55	1.95	158.97	14.01	122.96	11.03	226.70	212.02	164.93	8.13	2.22	55.43	24.79	9.90	26.20	21.29	9.26	9.95	1.41	30.35	37.33	54.21	235.64	412.70	653.61	264.47	29.56	259.46	289.31	35.81
bin	64	36.95	16.61	25.71	10.51	22.62	13.35	15.36	34.76	5.93	1.68	158.74	13.86	122.79	10.91	227.30	210.44	166.02	8.02	2.26	55.33	24.79	9.88	26.23	21.34	9.27	9.91	1.42	30.21	37.71	54.13	235.65	412.59	653.39	264.24	29.65	254.81	289.31	35.77
bin	32	37.14	16.63	25.77	10.56	22.69	12.87	15.04	34.77	5.59	1.77	158.63	13.52	122.54	10.84	225.97	210.59	164.45	7.87	2.25	55.53	24.90	9.87	26.22	21.22	9.24	9.93	1.43	30.21	36.65	53.84	236.22	412.37	652.75	264.47	29.59	253.85	288.50	35.79
bin	16	36.97	16.53	25.56	10.48	22.63	12.81	15.00	34.94	5.91	1.22	158.25	13.49	121.98	10.72	226.34	210.00	166.40	7.78	2.26	55.31	24.90	9.92	26.25	21.29	9.22	9.90	1.46	30.43	36.70	54.17	234.98	412.63	653.82	264.45	29.50	251.74	288.74	35.82
bin	8	36.55	16.22	25.41	10.59	21.99	13.02	15.39	34.65	5.42	1.14	158.86	13.76	122.02	10.35	225.76	211.10	164.72	7.79	2.22	55.30	24.85	9.84	26.28	21.24	9.18	9.96	1.42	30.54	36.30	53.78	235.65	412.13	653.18	264.22	29.55	251.86	288.65	35.03

Table 7: Absolute evaluation time for mxbai-embed-large-v1 on all tasks. Table 2 serves as a legend.

q	m	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	1024	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	512	0.877	0.944	0.985	1.026	0.883	0.585	0.958	0.817	0.692	0.523	0.987	0.990	0.982	0.989	0.993	0.980	0.955	0.863	0.978	0.995	1.001	0.984	0.999	1.003	0.995	1.000	0.997	1.011	0.909	0.987	0.998	0.977	0.985	0.996	1.003	0.976	1.003	1.000
f32	256	0.812	0.930	0.960	1.000	0.590	0.545	0.908	0.732	0.530	0.475	0.980	0.839	0.962	0.785	0.979	0.964	0.927	0.866	1.003	0.994	0.997	0.991	0.996	1.000	1.002	0.997	0.994	1.005	0.910	0.985	0.998	0.976	0.985	0.994	1.005	0.957	1.003	0.999
f32	128	0.626	0.925	0.967	1.009	0.561	0.529	0.921	0.549	0.473	0.280	0.974	0.852	0.955	0.740	0.974	0.949	0.914	0.813	0.965	0.984	1.008	0.987	0.990	1.003	1.007	0.995	1.026	1.001	0.897	0.985	0.995	0.976	0.985	0.996	1.008	0.938	1.002	0.999
f32	64	0.614	0.923	0.970	1.016	0.549	0.517	0.905	0.533	0.455	0.238	0.975	0.791	0.953	0.771	0.977	0.945	0.896	0.721	0.965	0.978	1.010	0.988	0.996	1.000	0.999	0.997	0.995	1.010	0.859	0.986	0.995	0.990	1.000	1.013	1.025	0.963	1.063	1.046
f32	32	0.604	0.934	0.960	1.001	0.537	0.508	0.878	0.516	0.445	0.250	0.971	0.783	0.955	0.735	0.977	0.944	0.895	0.766	0.961	0.982	1.010	0.996	0.992	0.998	1.003	1.002	0.999	1.002	0.860	0.978	1.002	0.998	1.001	1.013	1.033	0.968	1.060	1.048
f32	16	0.588	0.913	0.936	0.997	0.522	0.509	0.870	0.495	0.437	0.209	0.971	0.776	0.952	0.733	0.973	0.941	0.895	0.742	0.978	0.982	1.015	0.998	0.988	0.994	1.008	0.999	1.034	0.999	0.861	0.983	0.994	0.999	1.001	1.011	1.026	0.954	1.057	1.048
f32	8	0.584	0.908	0.933	0.989	0.516	0.512	0.883	0.489	0.394	0.198	0.971	0.778	0.945	0.669	0.969	0.947	0.894	0.686	0.962	0.978	1.012	0.987	0.993	1.002	1.000	1.003	0.994	1.015	0.859	0.975	0.998	0.998	1.000	1.013	1.027	0.944	1.060	1.044
int	1024	1.033	0.978	0.990	1.017	1.069	0.998	0.961	1.027	0.864	0.953	1.004	1.104	1.009	1.058	0.998	1.000	1.001	1.177	0.988	1.010	1.009	0.996	1.008	1.002	0.999	1.003	1.007	1.012	1.072	1.000	1.007	0.879	0.881	0.897	0.897	0.934	0.925	0.916
int	512	0.888	0.960	0.976	1.010	0.889	0.567	0.960	0.835	0.732	0.544	0.985	0.985	0.985	0.975	0.993	0.977	0.945	0.891	0.987	0.999	1.001	0.993	1.000	0.998	1.002	0.999	0.998	1.006	0.926	0.987	0.999	0.878	0.880	0.897	0.901	0.910	0.923	0.915
int	256	0.817	0.923	0.971	1.034	0.593	0.549	0.923	0.740	0.614	0.462	0.976	0.846	0.960	0.771	0.977	0.952	0.909	0.795	0.998	0.992	1.006	0.995	0.997	0.998	1.002	0.998	1.028	1.004	0.917	0.986	0.999	0.878	0.881	0.898	0.897	0.890	0.923	0.913
int	128	0.629	0.944	0.972	1.013	0.568	0.518	0.896	0.557	0.506	0.357	0.976	0.812	0.956	0.750	0.981	0.948	0.899	0.747	0.964	0.980	1.011	0.994	0.988	0.998	1.006	1.002	1.013	1.001	0.898	0.979	0.998	0.878	0.881	0.899	0.900	0.885	0.922	0.913
int	64	0.609	0.923	0.958	1.027	0.548	0.535	0.919	0.525	0.451	0.257	0.973	0.788	0.952	0.709	0.972	0.945	0.890	0.724	0.972	0.981	1.008	0.990	0.989	1.000	1.007	0.999	1.010	1.001	0.868	0.981	0.996	0.879	0.881	0.897	0.898	0.887	0.923	0.911
int	32	0.610	0.931	0.977	1.033	0.541	0.514	0.903	0.525	0.448	0.206	0.973	0.783	0.950	0.701	0.975	0.942	0.889	0.759	0.968	0.977	1.010	0.994	0.988	1.000	1.008	0.998	0.999	0.999	0.879	0.984	0.995	0.879	0.881	0.899	0.897	0.888	0.923	0.916
int	16	0.598	0.933	0.947	1.008	0.540	0.513	0.896	0.512	0.432	0.233	0.970	0.786	0.949	0.692	0.970	0.944	0.889	0.699	0.984	0.976	1.009	0.987	0.987	0.999	1.002	0.997	0.996	0.997	0.854	0.978	0.999	0.878	0.881	0.898	0.896	0.872	0.922	0.911
int	8	0.595	0.906	0.936	1.001	0.534	0.530	0.926	0.510	0.480	0.179	0.969	0.767	0.949	0.678	0.974	0.939	0.890	0.734	0.965	0.983	1.012	0.986	0.993	1.001	1.001	1.000	0.998	1.011	0.875	0.982	0.995	0.878	0.882	0.899	0.897	0.870	0.921	0.914
bin	1024	0.622	0.937	0.956	1.003	0.565	0.527	0.888	0.553	0.604	0.314	0.975	0.817	0.957	0.767	0.981	0.949	0.897	0.761	0.965	0.984	1.010	0.995	0.994	0.993	1.000	1.000	1.004	1.006	0.905	0.985	0.997	0.878	0.881	0.897	0.898	0.880	0.923	0.912
bin	512	0.611	0.924	0.974	1.020	0.547	0.514	0.901	0.531	0.443	0.232	0.977	0.798	0.954	0.713	0.978	0.944	0.895	0.793	0.959	0.979	1.010	0.993	0.996	0.997	1.001	1.002	0.998	1.004	0.890	0.984	0.997	0.878	0.881	0.898	0.897	0.880	0.922	0.912
bin	256	0.600	0.932	0.968	1.007	0.536	0.509	0.884	0.521	0.486	0.224	0.972	0.780	0.951	0.703	0.975	0.944	0.888	0.708	0.958	0.978	1.008	0.990	0.987	0.997	1.006	1.000	0.997	1.000	0.883	0.984	0.995	0.878	0.881	0.899	0.895	0.880	0.922	0.913
bin	128	0.602	0.905	0.951	0.998	0.533	0.508	0.890	0.516	0.520	0.209	0.971	0.778	0.952	0.694	0.972	0.948	0.888	0.703	0.962	0.979	1.009	0.994	0.988	1.000	1.008	1.000	0.998	1.002	0.869	0.983	0.995	0.878	0.882	0.898	0.896	0.882	0.924	0.916
bin	64	0.596	0.910	0.943	0.985	0.532	0.509	0.882	0.507	0.408	0.180	0.970	0.770	0.951	0.686	0.974	0.941	0.894	0.694	0.975	0.977	1.009	0.992	0.989	1.003	1.009	0.997	1.000	0.997	0.877	0.982	0.995	0.878	0.881	0.898	0.899	0.867	0.924	0.915
bin	32	0.599	0.910	0.945	0.990	0.533	0.491	0.864	0.507	0.385	0.190	0.969	0.751	0.949	0.682	0.969	0.942	0.886	0.680	0.973	0.980	1.014	0.990	0.989	0.997	1.006	0.999	1.013	0.997	0.853	0.977	0.997	0.878	0.880	0.898	0.897	0.863	0.921	0.915
bin	16	0.597	0.905	0.937	0.982	0.532	0.489	0.861	0.509	0.406	0.131	0.967	0.749	0.945	0.674	0.970	0.939	0.896	0.673	0.976	0.976	1.013	0.996	0.990	1.000	1.004	0.996	1.029	1.004	0.854	0.983	0.992	0.878	0.882	0.898	0.894	0.856	0.922	0.916
bin	8	0.590	0.888	0.931	0.993	0.517	0.496	0.884	0.505	0.373	0.122	0.970	0.765	0.945	0.650	0.968	0.944	0.887	0.674	0.962	0.976	1.011	0.987	0.991	0.998	0.999	1.002	1.002	1.008	0.845	0.975	0.995	0.877	0.881	0.897	0.896	0.857	0.922	0.896

Table 8: Relative evaluation time for mxbai-embed-large-v1 on all tasks. Table 2 serves as a legend.

q	m	Total Time	Average All Tasks	Average classification	Average clustering	Average sts	Average pairclass	Average retrieval	Average rerank	Average summ
f32	1024	3989.64	104.99	34.12	100.46	19.21	111.65	501.87	213.42	39.10
f32	512	3887.41	102.30	29.38	97.49	19.20	109.95	494.10	211.41	39.10
f32	256	3830.50	100.80	26.19	95.02	19.17	109.96	493.85	209.53	39.04
f32	128	3785.90	99.63	23.03	93.79	19.11	109.52	494.03	207.65	39.06
f32	64	3827.66	100.73	22.67	93.18	19.11	108.98	501.78	216.72	40.91
f32	32	3828.93	100.76	22.28	93.10	19.11	109.37	502.96	216.89	40.97
f32	16	3814.97	100.39	21.75	92.78	19.11	108.88	503.06	215.13	40.96
f32	8	3808.78	100.23	21.63	92.48	19.13	109.01	502.75	214.50	40.83
int	1024	3781.10	99.50	34.82	100.87	19.35	113.24	443.57	198.04	35.82
int	512	3683.75	96.94	29.58	97.29	19.22	110.28	442.99	195.50	35.77
int	256	3623.37	95.35	26.43	94.33	19.18	110.11	443.42	193.48	35.71
int	128	3586.75	94.39	23.13	93.70	19.09	109.65	443.51	192.89	35.68
int	64	3572.91	94.02	22.63	92.82	19.09	109.06	443.40	193.24	35.62
int	32	3573.05	94.03	22.58	92.75	19.07	109.23	443.63	193.29	35.81
int	16	3560.70	93.70	22.23	92.51	19.04	109.06	443.30	191.57	35.62
int	8	3562.39	93.75	22.17	92.53	19.14	109.11	443.71	191.37	35.72
bin	1024	3585.65	94.36	22.97	93.81	19.12	109.80	443.34	192.51	35.66
bin	512	3576.19	94.11	22.64	93.20	19.10	109.52	443.53	192.43	35.64
bin	256	3568.42	93.91	22.33	92.81	19.05	109.28	443.52	192.37	35.71
bin	128	3568.44	93.91	22.16	92.82	19.08	109.06	443.59	192.78	35.81
bin	64	3559.44	93.67	21.98	92.57	19.06	109.16	443.41	191.26	35.77
bin	32	3552.10	93.48	21.93	92.18	19.08	108.91	443.20	190.65	35.79
bin	16	3550.49	93.43	21.86	92.21	19.09	108.62	443.63	189.99	35.82
bin	8	3545.92	93.31	21.73	92.09	19.08	108.58	443.18	190.02	35.03

Table 9: Evaluation time for mxbai-embed-large-v1, averaged per category. Table 2 serves as a legend.

q	m	Average Relative classification	Average Relative clustering	Average Relative sts	Average Relative pairclass	Average Relative retrieval	Average Relative rerank	Average Relative summ	Average Relative all
f32	1024	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	512	0.884	0.895	0.996	0.965	0.986	0.994	1.000	0.943
f32	256	0.809	0.831	0.998	0.964	0.985	0.988	0.999	0.910
f32	128	0.761	0.792	0.997	0.959	0.986	0.983	0.999	0.888
f32	64	0.753	0.772	0.994	0.947	1.001	1.017	1.046	0.885
f32	32	0.742	0.772	0.994	0.946	1.004	1.020	1.048	0.883
f32	16	0.729	0.763	1.000	0.946	1.004	1.012	1.048	0.879
f32	8	0.727	0.745	0.995	0.944	1.004	1.010	1.044	0.872
int	1024	1.009	1.017	1.004	1.026	0.886	0.919	0.916	0.992
int	512	0.885	0.901	0.998	0.971	0.885	0.911	0.915	0.929
int	256	0.819	0.826	1.002	0.967	0.886	0.903	0.913	0.895
int	128	0.762	0.793	0.996	0.958	0.886	0.902	0.913	0.872
int	64	0.756	0.766	0.996	0.948	0.886	0.903	0.911	0.863
int	32	0.754	0.763	0.994	0.953	0.886	0.903	0.916	0.862
int	16	0.743	0.756	0.994	0.944	0.886	0.897	0.911	0.856
int	8	0.742	0.756	0.995	0.950	0.886	0.896	0.914	0.857
bin	1024	0.756	0.802	0.995	0.963	0.886	0.900	0.912	0.873
bin	512	0.753	0.773	0.994	0.957	0.886	0.900	0.912	0.864
bin	256	0.745	0.763	0.992	0.954	0.886	0.899	0.913	0.859
bin	128	0.738	0.764	0.994	0.949	0.886	0.901	0.916	0.858
bin	64	0.733	0.747	0.995	0.951	0.886	0.896	0.915	0.853
bin	32	0.730	0.740	0.996	0.942	0.885	0.894	0.915	0.850
bin	16	0.727	0.735	0.999	0.943	0.886	0.891	0.916	0.848
bin	8	0.726	0.730	0.994	0.938	0.885	0.891	0.896	0.844

Table 10: Relative evaluation time for mxbai-embed-large-v1, averaged per category. Table 2 serves as a legend.

q	m	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	1024	32768	0.878	0.509	0.597	0.750	0.765	0.799	0.940	0.768	0.601	0.603	0.419	0.400	0.367	0.352	0.561	0.553	0.406	0.508	0.861	0.828	0.791	0.898	0.852	0.893	0.868	0.893	0.892	0.690	0.968	0.785	0.862	0.666	0.403	0.377	0.652	0.325	0.552	0.327
f32	512	16384	0.875	0.500	0.588	0.725	0.759	0.792	0.932	0.758	0.609	0.591	0.424	0.395	0.372	0.352	0.539	0.545	0.404	0.524	0.856	0.826	0.788	0.899	0.851	0.892	0.865	0.891	0.885	0.695	0.968	0.784	0.860	0.661	0.396	0.370	0.636	0.324	0.549	0.316
f32	256	8192	0.872	0.483	0.578	0.708	0.743	0.779	0.923	0.741	0.599	0.587	0.410	0.380	0.363	0.343	0.512	0.526	0.405	0.480	0.853	0.823	0.789	0.894	0.848	0.886	0.859	0.889	0.883	0.698	0.965	0.777	0.856	0.632	0.363	0.347	0.640	0.314	0.545	0.308
f32	128	4096	0.862	0.460	0.565	0.672	0.713	0.745	0.905	0.700	0.583	0.582	0.395	0.365	0.353	0.327	0.481	0.493	0.401	0.451	0.852	0.815	0.777	0.885	0.837	0.879	0.844	0.880	0.868	0.697	0.960	0.767	0.851	0.589	0.321	0.300	0.638	0.303	0.536	0.301
f32	64	2048	0.834	0.411	0.520	0.661	0.664	0.681	0.861	0.631	0.579	0.569	0.358	0.320	0.340	0.315	0.367	0.423	0.393	0.373	0.815	0.805	0.765	0.865	0.822	0.865	0.829	0.863	0.824	0.686	0.944	0.746	0.836	0.491	0.262	0.233	0.616	0.281	0.516	0.302
f32	32	1024	0.774	0.374	0.498	0.628	0.576	0.591	0.780	0.502	0.571	0.553	0.312	0.279	0.322	0.298	0.268	0.347	0.391	0.302	0.779	0.787	0.744	0.838	0.792	0.835	0.797	0.828	0.793	0.658	0.915	0.708	0.812	0.326	0.180	0.146	0.598	0.265	0.486	0.285
f32	16	512	0.632	0.332	0.479	0.593	0.431	0.455	0.628	0.344	0.526	0.528	0.253	0.215	0.290	0.270	0.184	0.271	0.376	0.219	0.768	0.751	0.687	0.776	0.728	0.798	0.747	0.787	0.784	0.662	0.823	0.638	0.767	0.151	0.071	0.072	0.557	0.267	0.421	0.278
f32	8	256	0.396	0.313	0.453	0.588	0.289	0.346	0.480	0.211	0.509	0.511	0.190	0.162	0.245	0.228	0.135	0.224	0.370	0.157	0.661	0.663	0.614	0.672	0.628	0.700	0.689	0.705	0.684	0.647	0.538	0.570	0.702	0.043	0.013	0.032	0.517	0.252	0.356	0.265
int	1024	8192	0.870	0.492	0.579	0.648	0.734	0.774	0.935	0.713	0.608	0.593	0.424	0.394	0.370	0.350	0.558	0.549	0.404	0.526	0.856	0.829	0.766	0.897	0.843	0.890	0.861	0.889	0.891	0.691	0.970	0.783	0.863	0.661	0.406	0.373	0.640	0.319	0.550	0.321
int	512	4096	0.867	0.481	0.569	0.631	0.726	0.759	0.926	0.687	0.597	0.585	0.422	0.389	0.369	0.351	0.543	0.540	0.404	0.518	0.852	0.827	0.763	0.895	0.841	0.887	0.858	0.887	0.888	0.688	0.969	0.781	0.862	0.657	0.395	0.361	0.633	0.317	0.546	0.313
int	256	2048	0.859	0.476	0.556	0.631	0.706	0.741	0.911	0.655	0.597	0.582	0.408	0.382	0.362	0.341	0.521	0.523	0.402	0.478	0.860	0.823	0.758	0.891	0.839	0.882	0.852	0.884	0.881	0.693	0.967	0.774	0.858	0.629	0.371	0.338	0.634	0.312	0.542	0.301
int	128	1024	0.841	0.441	0.533	0.569	0.668	0.708	0.887	0.612	0.575	0.572	0.395	0.367	0.352	0.330	0.480	0.490	0.403	0.446	0.851	0.816	0.746	0.882	0.832	0.875	0.843	0.875	0.874	0.697	0.963	0.764	0.852	0.572	0.323	0.301	0.632	0.308	0.533	0.292
int	64	512	0.793	0.378	0.479	0.530	0.604	0.624	0.814	0.509	0.583	0.563	0.342	0.321	0.336	0.315	0.366	0.422	0.391	0.369	0.807	0.807	0.728	0.865	0.811	0.860	0.824	0.863	0.834	0.679	0.946	0.746	0.838	0.471	0.265	0.232	0.612	0.291	0.506	0.295
int	32	256	0.683	0.330	0.451	0.512	0.472	0.506	0.685	0.340	0.574	0.558	0.302	0.272	0.316	0.301	0.259	0.345	0.392	0.293	0.763	0.792	0.690	0.833	0.781	0.834	0.789	0.823	0.802	0.646	0.919	0.708	0.812	0.310	0.168	0.144	0.583	0.277	0.474	0.286
int	16	128	0.459	0.280	0.470	0.550	0.270	0.309	0.478	0.187	0.520	0.520	0.242	0.212	0.289	0.265	0.179	0.270	0.375	0.207	0.697	0.755	0.646	0.782	0.713	0.779	0.732	0.773	0.764	0.638	0.830	0.627	0.767	0.153	0.057	0.066	0.536	0.266	0.413	0.263
int	8	64	0.329	0.271	0.438	0.572	0.214	0.271	0.364	0.167	0.510	0.518	0.179	0.157	0.242	0.229	0.134	0.225	0.370	0.153	0.638	0.685	0.573	0.649	0.592	0.669	0.610	0.710	0.640	0.630	0.573	0.569	0.701	0.041	0.012	0.029	0.490	0.252	0.346	0.255
bin	1024	1024	0.764	0.365	0.482	0.599	0.601	0.622	0.800	0.524	0.551	0.540	0.318	0.283	0.315	0.295	0.367	0.404	0.358	0.350	0.714	0.788	0.735	0.839	0.795	0.833	0.820	0.849	0.846	0.662	0.936	0.728	0.833	0.395	0.182	0.156	0.592	0.283	0.469	0.311
bin	512	512	0.634	0.317	0.443	0.559	0.472	0.500	0.648	0.358	0.517	0.487	0.243	0.210	0.286	0.267	0.240	0.314	0.353	0.236	0.657	0.760	0.708	0.792	0.758	0.788	0.777	0.802	0.797	0.635	0.889	0.684	0.809	0.232	0.104	0.081	0.557	0.268	0.420	0.293
bin	256	256	0.446	0.263	0.411	0.532	0.310	0.365	0.495	0.220	0.494	0.507	0.189	0.170	0.260	0.246	0.151	0.252	0.358	0.184	0.569	0.706	0.654	0.714	0.681	0.710	0.704	0.739	0.751	0.589	0.796	0.608	0.760	0.101	0.039	0.048	0.541	0.261	0.358	0.282
bin	128	128	0.271	0.265	0.427	0.585	0.190	0.224	0.340	0.136	0.492	0.495	0.153	0.140	0.233	0.219	0.120	0.221	0.363	0.146	0.370	0.621	0.545	0.600	0.567	0.592	0.607	0.626	0.608	0.532	0.565	0.521	0.697	0.040	0.015	0.026	0.507	0.261	0.293	0.288
bin	64	64	0.196	0.251	0.406	0.560	0.126	0.186	0.274	0.070	0.491	0.484	0.142	0.123	0.227	0.213	0.107	0.204	0.360	0.127	0.167	0.526	0.435	0.462	0.436	0.473	0.477	0.493	0.485	0.435	0.313	0.433	0.611	0.020	0.014	0.023	0.469	0.247	0.257	0.314
bin	32	32	0.103	0.194	0.385	0.584	0.072	0.147	0.174	0.025	0.484	0.488	0.136	0.118	0.222	0.210	0.101	0.203	0.363	0.120	-0.049	0.439	0.386	0.308	0.334	0.371	0.370	0.390	0.428	0.284	0.193	0.371	0.534	0.011	0.006	0.016	0.446	0.244	0.239	0.281
bin	16	16	0.047	0.181	0.346	0.591	0.041	0.090	0.124	0.009	0.500	0.479	0.132	0.112	0.227	0.205	0.097	0.201	0.365	0.119	0.094	0.363	0.315	0.254	0.270	0.348	0.326	0.316	0.327	0.228	0.133	0.334	0.479	0.010	0.003	0.020	0.416	0.237	0.203	0.262
bin	8	8	0.030	0.191	0.328	0.528	0.034	0.091	0.087	0.006	0.498	0.502	0.118	0.107	0.217	0.205	0.101	0.200	0.344	0.119	-0.015	0.204	0.223	0.122	0.171	0.224	0.173	0.182	0.161	0.223	0.070	0.328	0.426	0.000	0.000	0.002	0.294	0.103	0.052	0.195

Table 11: Absolute performance for mxbai-embed-large-v1 on all tasks. Table 2 serves as a legend.

q	m	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	1024	32768	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	512	16384	0.997	0.984	0.985	0.966	0.992	0.991	0.992	0.986	1.012	0.981	1.012	0.986	1.014	1.000	0.960	0.987	0.995	1.031	0.995	0.997	0.996	1.001	0.999	0.998	0.997	0.998	0.992	1.007	0.999	0.998	0.998	0.993	0.983	0.981	0.977	0.995	0.995	0.968
f32	256	8192	0.993	0.950	0.969	0.943	0.971	0.975	0.982	0.965	0.997	0.973	0.980	0.951	0.990	0.974	0.912	0.953	0.997	0.944	0.991	0.994	0.998	0.996	0.995	0.992	0.990	0.996	0.990	1.012	0.997	0.989	0.992	0.950	0.901	0.919	0.982	0.965	0.987	0.943
f32	128	4096	0.982	0.905	0.947	0.896	0.932	0.932	0.963	0.911	0.970	0.966	0.944	0.913	0.964	0.931	0.858	0.892	0.988	0.886	0.991	0.985	0.982	0.986	0.983	0.984	0.972	0.986	0.973	1.009	0.992	0.976	0.987	0.885	0.798	0.797	0.979	0.931	0.971	0.919
f32	64	2048	0.950	0.808	0.871	0.881	0.868	0.852	0.916	0.821	0.963	0.943	0.855	0.798	0.929	0.895	0.653	0.765	0.967	0.733	0.947	0.973	0.967	0.963	0.965	0.968	0.955	0.967	0.924	0.993	0.975	0.950	0.969	0.738	0.651	0.618	0.945	0.862	0.935	0.924
f32	32	1024	0.882	0.735	0.835	0.838	0.752	0.739	0.830	0.654	0.950	0.917	0.746	0.696	0.879	0.848	0.477	0.628	0.963	0.595	0.906	0.951	0.940	0.933	0.930	0.935	0.919	0.927	0.889	0.953	0.945	0.901	0.942	0.489	0.449	0.388	0.918	0.814	0.881	0.871
f32	16	512	0.720	0.652	0.802	0.790	0.564	0.570	0.669	0.448	0.875	0.876	0.605	0.538	0.790	0.767	0.328	0.490	0.925	0.431	0.892	0.907	0.869	0.865	0.855	0.894	0.861	0.881	0.879	0.959	0.850	0.812	0.890	0.227	0.177	0.191	0.855	0.820	0.762	0.850
f32	8	256	0.450	0.616	0.758	0.783	0.377	0.433	0.511	0.275	0.847	0.848	0.455	0.405	0.668	0.648	0.240	0.405	0.911	0.310	0.768	0.801	0.777	0.748	0.737	0.784	0.794	0.789	0.766	0.937	0.556	0.726	0.814	0.065	0.032	0.084	0.793	0.775	0.645	0.809
int	1024	8192	0.991	0.967	0.970	0.864	0.959	0.969	0.996	0.928	1.012	0.984	1.013	0.985	1.009	0.996	0.994	0.994	0.994	1.036	0.994	1.001	0.969	0.999	0.990	0.996	0.992	0.996	0.998	1.001	1.002	0.997	1.001	0.993	1.007	0.988	0.982	0.981	0.995	0.982
int	512	4096	0.987	0.946	0.953	0.842	0.949	0.950	0.985	0.894	0.994	0.970	1.007	0.973	1.006	0.997	0.968	0.977	0.993	1.020	0.990	0.999	0.964	0.997	0.987	0.993	0.989	0.993	0.995	0.996	1.001	0.995	0.999	0.987	0.981	0.958	0.971	0.974	0.989	0.958
int	256	2048	0.978	0.936	0.931	0.841	0.922	0.927	0.969	0.853	0.993	0.965	0.974	0.954	0.989	0.968	0.928	0.947	0.990	0.940	0.999	0.995	0.959	0.993	0.985	0.987	0.982	0.990	0.988	1.003	0.999	0.985	0.995	0.945	0.921	0.897	0.973	0.959	0.982	0.921
int	128	1024	0.958	0.867	0.893	0.758	0.872	0.886	0.944	0.796	0.956	0.949	0.944	0.917	0.961	0.939	0.855	0.887	0.993	0.877	0.989	0.986	0.943	0.983	0.976	0.980	0.971	0.979	0.980	1.010	0.995	0.972	0.989	0.859	0.801	0.798	0.970	0.947	0.965	0.893
int	64	512	0.903	0.742	0.803	0.707	0.789	0.780	0.866	0.663	0.970	0.934	0.816	0.803	0.916	0.895	0.652	0.764	0.963	0.727	0.938	0.974	0.920	0.964	0.951	0.962	0.950	0.966	0.934	0.983	0.977	0.950	0.972	0.708	0.657	0.616	0.939	0.894	0.916	0.904
int	32	256	0.778	0.649	0.756	0.683	0.617	0.633	0.730	0.443	0.955	0.926	0.720	0.680	0.862	0.857	0.461	0.625	0.965	0.577	0.886	0.956	0.873	0.927	0.916	0.933	0.910	0.922	0.899	0.936	0.950	0.901	0.942	0.465	0.418	0.382	0.895	0.853	0.859	0.875
int	16	128	0.522	0.550	0.788	0.733	0.352	0.387	0.509	0.243	0.865	0.863	0.579	0.530	0.788	0.752	0.319	0.489	0.924	0.407	0.810	0.912	0.818	0.871	0.836	0.872	0.843	0.866	0.857	0.925	0.858	0.798	0.889	0.229	0.140	0.175	0.823	0.818	0.747	0.805
int	8	64	0.375	0.532	0.734	0.762	0.280	0.339	0.387	0.217	0.848	0.859	0.429	0.393	0.661	0.651	0.238	0.406	0.910	0.301	0.742	0.828	0.724	0.723	0.695	0.749	0.703	0.795	0.718	0.912	0.592	0.725	0.813	0.062	0.029	0.076	0.752	0.774	0.627	0.780
bin	1024	1024	0.870	0.718	0.808	0.798	0.785	0.779	0.851	0.682	0.916	0.895	0.760	0.707	0.859	0.838	0.654	0.731	0.881	0.689	0.830	0.953	0.929	0.934	0.933	0.932	0.945	0.950	0.948	0.959	0.967	0.926	0.966	0.594	0.453	0.414	0.909	0.871	0.849	0.952
bin	512	512	0.722	0.623	0.742	0.745	0.617	0.625	0.689	0.467	0.860	0.809	0.582	0.525	0.779	0.758	0.428	0.568	0.869	0.464	0.763	0.918	0.895	0.882	0.889	0.882	0.896	0.898	0.893	0.920	0.919	0.871	0.938	0.349	0.258	0.214	0.855	0.824	0.761	0.897
bin	256	256	0.507	0.516	0.688	0.709	0.405	0.457	0.527	0.286	0.822	0.841	0.452	0.426	0.710	0.699	0.269	0.457	0.882	0.362	0.661	0.853	0.828	0.795	0.799	0.794	0.811	0.828	0.843	0.853	0.823	0.773	0.882	0.151	0.096	0.126	0.831	0.801	0.649	0.863
bin	128	128	0.309	0.521	0.716	0.780	0.248	0.281	0.362	0.177	0.818	0.821	0.365	0.349	0.635	0.623	0.215	0.399	0.893	0.286	0.430	0.750	0.690	0.668	0.665	0.662	0.699	0.701	0.681	0.770	0.584	0.663	0.808	0.060	0.037	0.069	0.777	0.802	0.531	0.882
bin	64	64	0.223	0.492	0.681	0.747	0.165	0.233	0.291	0.092	0.818	0.802	0.339	0.307	0.620	0.606	0.190	0.370	0.886	0.249	0.194	0.635	0.550	0.515	0.512	0.529	0.550	0.552	0.544	0.631	0.323	0.551	0.708	0.030	0.036	0.061	0.719	0.758	0.466	0.960
bin	32	32	0.117	0.381	0.644	0.778	0.095	0.183	0.185	0.033	0.805	0.810	0.324	0.295	0.606	0.598	0.180	0.367	0.894	0.237	-0.056	0.530	0.488	0.343	0.392	0.416	0.426	0.437	0.480	0.411	0.199	0.472	0.620	0.017	0.015	0.043	0.684	0.750	0.433	0.861
bin	16	16	0.054	0.356	0.579	0.787	0.054	0.113	0.132	0.011	0.832	0.795	0.316	0.280	0.618	0.583	0.172	0.363	0.898	0.233	0.109	0.439	0.398	0.283	0.317	0.390	0.376	0.354	0.366	0.331	0.138	0.425	0.555	0.016	0.007	0.053	0.638	0.730	0.368	0.800
bin	8	8	0.034	0.376	0.549	0.704	0.044	0.114	0.092	0.008	0.829	0.833	0.282	0.267	0.593	0.583	0.180	0.362	0.847	0.234	-0.018	0.246	0.281	0.136	0.201	0.251	0.199	0.204	0.180	0.323	0.072	0.417	0.495	0.000	0.000	0.005	0.451	0.318	0.093	0.597

Table 12: Relative performance for mxbai-embed-large-v1 on all tasks. Table 2 serves as a legend.

q	m	req. bits	Average classification	Average clustering	Average sts	Average pairclass	Average retrieval	Average rerank	Average summ	Average all
f32	1024	32768	0.751	0.477	0.847	0.872	0.482	0.510	0.327	0.662
f32	512	16384	0.741	0.475	0.845	0.870	0.476	0.503	0.316	0.658
f32	256	8192	0.728	0.461	0.842	0.866	0.447	0.500	0.308	0.647
f32	128	4096	0.703	0.443	0.834	0.859	0.404	0.492	0.301	0.630
f32	64	2048	0.658	0.404	0.814	0.842	0.329	0.471	0.302	0.596
f32	32	1024	0.590	0.364	0.785	0.811	0.217	0.450	0.285	0.551
f32	16	512	0.487	0.313	0.749	0.743	0.098	0.415	0.278	0.488
f32	8	256	0.384	0.273	0.666	0.604	0.029	0.375	0.265	0.415
int	1024	8192	0.718	0.478	0.841	0.872	0.480	0.503	0.321	0.653
int	512	4096	0.706	0.472	0.839	0.871	0.471	0.498	0.313	0.647
int	256	2048	0.692	0.460	0.836	0.866	0.446	0.496	0.301	0.637
int	128	1024	0.657	0.441	0.829	0.860	0.399	0.491	0.292	0.618
int	64	512	0.591	0.401	0.808	0.843	0.323	0.469	0.295	0.579
int	32	256	0.497	0.361	0.775	0.813	0.207	0.445	0.286	0.527
int	16	128	0.375	0.308	0.728	0.741	0.092	0.405	0.263	0.456
int	8	64	0.328	0.272	0.640	0.614	0.027	0.363	0.255	0.395
bin	1024	1024	0.595	0.378	0.788	0.832	0.245	0.448	0.311	0.561
bin	512	512	0.491	0.315	0.748	0.794	0.139	0.415	0.293	0.497
bin	256	256	0.380	0.281	0.682	0.722	0.062	0.387	0.282	0.433
bin	128	128	0.305	0.258	0.567	0.594	0.027	0.354	0.288	0.366
bin	64	64	0.259	0.248	0.439	0.452	0.019	0.324	0.314	0.306
bin	32	32	0.210	0.245	0.326	0.366	0.011	0.310	0.281	0.256
bin	16	16	0.179	0.244	0.284	0.315	0.011	0.285	0.262	0.232
bin	8	8	0.162	0.241	0.167	0.275	0.001	0.149	0.195	0.180

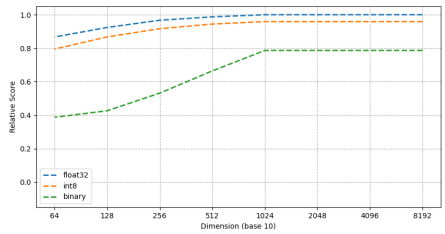
Table 13: Performance for mxbai-embed-large-v1, averaged per category. Table 2 serves as a legend.

q	m	req. bits	Average Relative classification	Average Relative clustering	Average Relative sts	Average Relative pairclass	Average Relative retrieval	Average Relative rerank	Average Relative summ	Average Relative all
f32	1024	32768	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	512	16384	0.987	0.998	0.998	0.998	0.986	0.989	0.968	0.993
f32	256	8192	0.968	0.967	0.995	0.993	0.923	0.978	0.943	0.974
f32	128	4096	0.933	0.931	0.985	0.985	0.827	0.960	0.919	0.944
f32	64	2048	0.871	0.850	0.962	0.965	0.669	0.914	0.924	0.886
f32	32	1024	0.783	0.770	0.928	0.929	0.442	0.871	0.871	0.812
f32	16	512	0.652	0.662	0.886	0.851	0.199	0.812	0.850	0.714
f32	8	256	0.525	0.574	0.790	0.699	0.060	0.738	0.809	0.609
int	1024	8192	0.955	1.002	0.994	1.000	0.996	0.986	0.982	0.988
int	512	4096	0.938	0.991	0.991	0.998	0.975	0.978	0.958	0.977
int	256	2048	0.920	0.965	0.988	0.993	0.921	0.972	0.921	0.960
int	128	1024	0.872	0.928	0.980	0.985	0.820	0.960	0.893	0.927
int	64	512	0.782	0.844	0.954	0.966	0.660	0.916	0.904	0.862
int	32	256	0.661	0.763	0.916	0.931	0.422	0.869	0.875	0.779
int	16	128	0.510	0.652	0.861	0.848	0.182	0.796	0.805	0.671
int	8	64	0.453	0.570	0.759	0.710	0.056	0.718	0.780	0.583
bin	1024	1024	0.786	0.793	0.931	0.953	0.487	0.876	0.952	0.827
bin	512	512	0.654	0.664	0.884	0.909	0.274	0.814	0.897	0.726
bin	256	256	0.512	0.592	0.806	0.826	0.125	0.760	0.863	0.634
bin	128	128	0.424	0.540	0.672	0.685	0.056	0.704	0.882	0.545
bin	64	64	0.365	0.519	0.521	0.527	0.042	0.648	0.960	0.472
bin	32	32	0.302	0.512	0.387	0.430	0.025	0.623	0.861	0.408
bin	16	16	0.261	0.509	0.336	0.373	0.025	0.579	0.800	0.376
bin	8	8	0.240	0.501	0.200	0.328	0.002	0.287	0.597	0.299

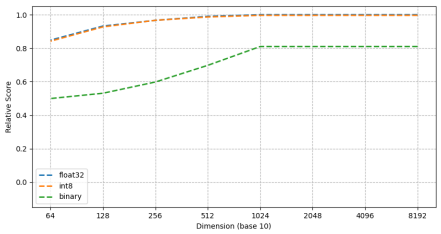
Table 14: Relative performance for mxbai-embed-large-v1, averaged per category. Table 2 serves as a legend.

D. stella_en_400M_v5

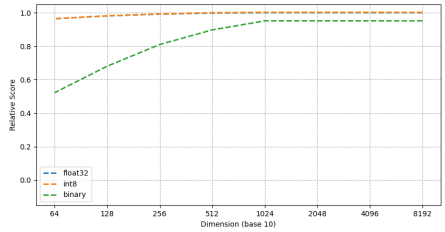
This section contains additional figures and tables for the Stella model.



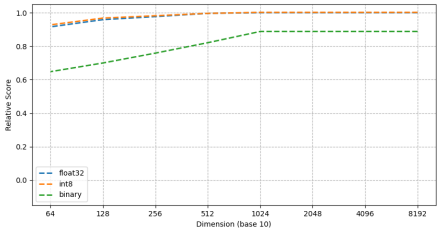
(a) Classification



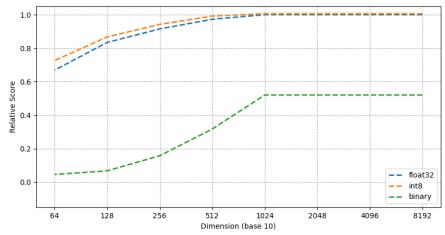
(b) Clustering



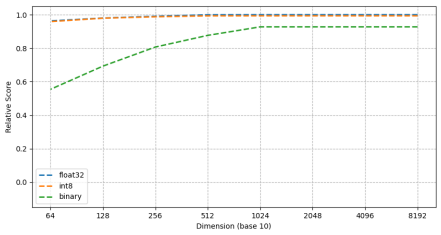
(c) Pair Classification



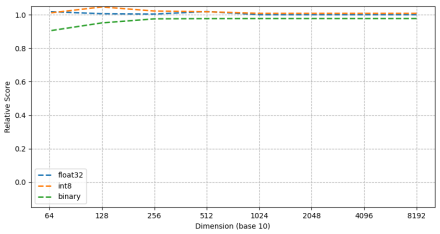
(d) Re-Ranking



(e) Retrieval

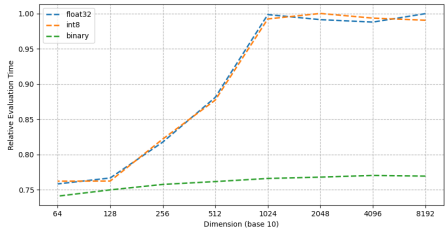


(f) Semantic Textual Similarity

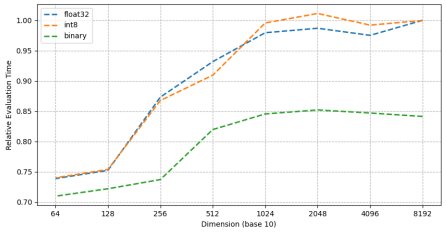


(g) Summarization

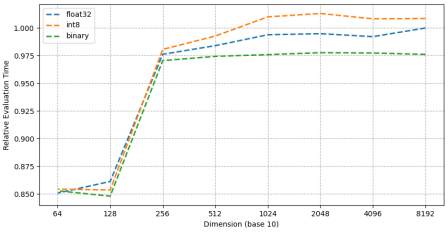
Figure 10: Relative performance of the STELLA model for each task category in the MTEB. The x-axis of each subfigure represents the embedding size while the y-axis represents the relative performance. For each task the reference point is the performance of the float32 embeddings at the full embedding size. Blue: float32, Orange: int8, Green: binary.



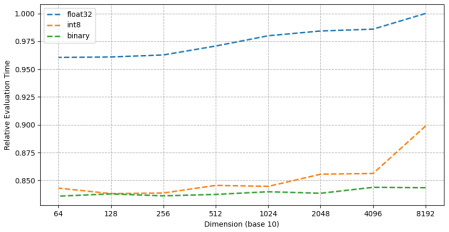
(a) Classification



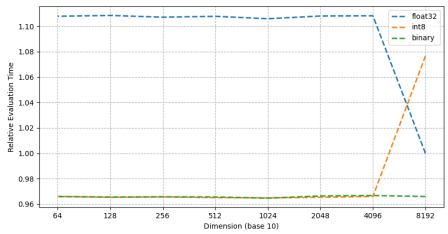
(b) Clustering



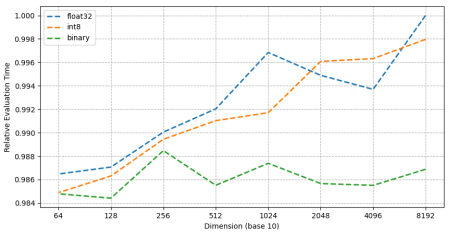
(c) Pair Classification



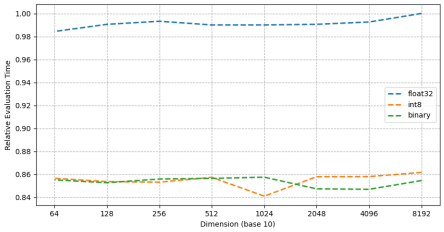
(d) Re-Ranking



(e) Retrieval

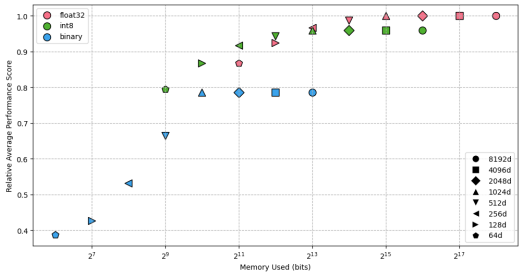


(f) Semantic Textual Similarity

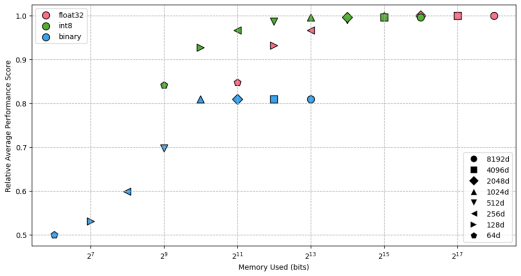


(g) Summarization

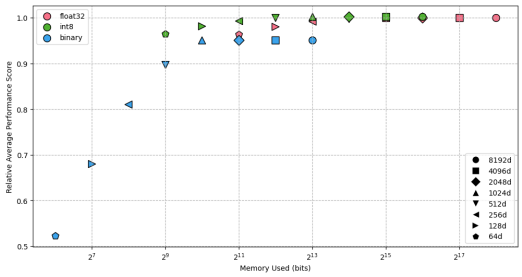
Figure 11: Relative evaluation time of the STELLA model for each task category in the MTEB. The x-axis of each subfigure represents the embedding size while the y-axis represents the relative evaluation time. For each task the reference point is the evaluation time of the float32 embeddings at the full embedding size. Blue: float32, Orange: int8, Green: binary.



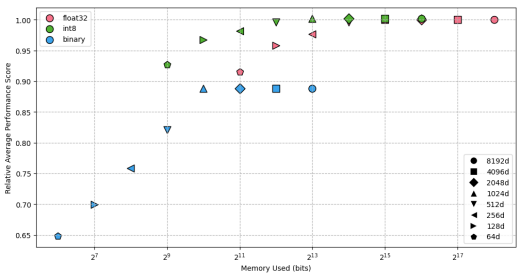
(a) Classification



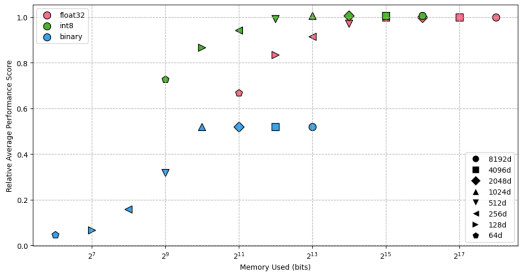
(b) Clustering



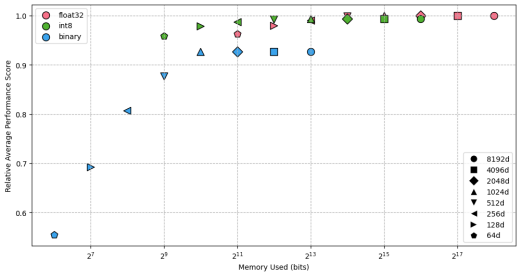
(c) Pair Classification



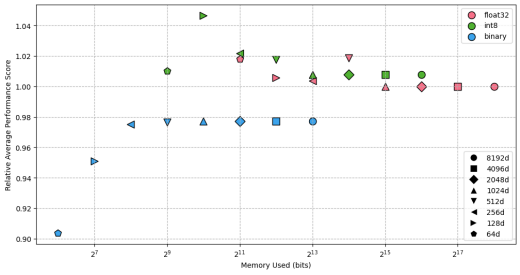
(d) Re-Ranking



(e) Retrieval



(f) Semantic Textual Similarity



(g) Summarization

Figure 12: Relative accuracy compute trade-off of the STELLA model, averaged over all selected MTEB tasks within each category. The x-axis represents memory consumption (log scale) and the y-axis represents relative performance (averaged over tasks within the category). Reference point: float32 embeddings at full size.

q	m	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	8192	68.87	19.61	26.83	10.00	45.49	26.83	19.20	73.21	13.70	9.45	203.77	19.37	163.82	16.57	178.64	171.79	219.06	11.83	2.13	70.07	27.63	11.22	29.00	24.27	10.62	11.36	1.79	38.50	49.42	72.46	296.81	529.52	731.93	312.82	35.43	339.23	350.92	52.36
f32	4096	66.67	19.92	26.79	9.69	43.97	26.94	18.93	72.99	12.40	7.54	203.48	19.04	166.01	16.48	188.83	175.55	217.73	11.67	2.17	70.09	27.56	11.27	29.12	24.01	10.51	11.23	1.71	38.07	48.92	71.51	296.74	586.85	816.29	344.50	34.79	331.49	350.35	51.97
f32	2048	66.89	19.83	26.77	9.72	44.14	26.98	19.45	72.56	11.78	7.55	203.81	19.47	167.52	17.52	189.60	176.49	221.25	12.14	2.19	70.13	27.57	11.25	29.09	23.94	10.49	11.19	1.73	38.06	49.41	71.57	295.93	587.52	816.11	344.02	34.84	329.05	350.70	51.87
f32	1024	68.45	19.89	26.79	9.76	44.95	27.02	19.12	74.36	11.91	7.60	203.35	19.43	167.14	16.88	189.22	175.62	219.14	11.83	2.21	70.14	27.59	11.26	28.97	23.93	10.54	11.29	1.72	38.35	49.38	71.41	295.88	586.12	814.61	343.47	34.84	324.32	351.03	51.84
f32	512	60.00	19.01	26.68	9.85	38.83	16.10	18.30	60.24	10.46	6.57	203.08	18.28	165.17	15.47	187.93	173.07	213.95	11.13	2.17	70.03	27.43	11.24	28.88	23.78	10.44	11.30	1.72	38.22	48.11	71.23	295.48	587.22	816.29	343.89	34.94	313.19	351.92	51.84
f32	256	55.69	18.79	26.42	9.82	26.74	15.12	17.54	54.37	8.74	3.76	201.91	17.88	163.87	14.98	185.89	170.20	210.63	10.48	2.15	69.65	27.44	11.18	29.01	23.97	10.49	11.27	1.69	38.28	47.05	71.17	295.17	587.46	815.04	343.64	34.86	306.45	351.22	52.01
f32	128	44.21	18.53	26.42	9.76	25.42	14.73	17.45	41.76	6.67	2.66	183.28	14.80	149.70	12.37	163.87	149.24	189.31	9.24	2.11	69.20	27.56	11.15	28.84	23.94	10.55	11.30	1.69	38.10	42.30	61.98	258.96	587.22	816.54	344.42	34.95	303.29	351.63	51.86
f32	64	43.37	18.27	26.11	9.96	24.67	14.68	17.21	40.37	7.08	2.16	182.62	14.61	149.90	12.73	161.90	147.39	187.57	8.13	2.07	69.03	27.59	11.08	28.97	24.13	10.53	11.27	1.70	38.21	40.74	62.04	258.66	587.93	816.07	343.61	34.96	304.09	350.32	51.54
int	8192	70.14	19.04	26.74	9.94	46.23	26.53	17.42	75.61	13.40	8.94	204.93	19.17	165.54	16.27	187.74	173.88	214.92	12.32	2.18	71.14	27.57	11.32	29.17	24.00	10.51	11.26	1.73	38.14	50.77	72.22	297.34	566.43	794.73	336.03	34.69	290.25	302.42	45.12
int	4096	68.68	19.26	26.80	9.81	46.28	26.84	17.99	75.77	13.29	8.49	203.59	19.11	165.99	16.21	188.49	173.50	214.11	12.22	2.20	70.88	27.55	11.29	29.08	24.05	10.49	11.26	1.69	38.38	50.59	72.32	297.75	511.30	710.45	300.87	30.28	289.15	302.42	44.93
int	2048	71.36	19.59	26.90	9.70	46.41	26.83	17.97	76.04	13.61	9.02	204.04	19.94	166.15	17.24	188.31	173.71	216.49	12.15	2.18	70.69	27.57	11.29	29.05	24.03	10.53	11.28	1.71	38.23	51.43	72.38	296.75	510.33	710.13	301.00	30.13	289.56	302.80	44.92
int	1024	69.98	19.43	26.40	9.70	45.75	26.85	18.13	75.31	13.28	8.65	203.18	19.42	166.12	16.42	188.59	173.95	215.05	12.03	2.16	70.38	27.50	11.19	29.00	24.00	10.53	11.24	1.69	38.18	51.02	72.23	297.11	509.71	710.16	300.89	30.12	286.58	294.46	44.04
int	512	60.75	18.79	26.31	9.88	39.17	16.07	17.61	60.99	9.13	5.81	202.90	18.37	164.81	15.12	187.49	171.82	210.78	11.10	2.16	70.04	27.41	11.07	28.94	23.99	10.46	11.24	1.73	37.98	48.94	71.60	296.73	511.02	709.68	300.51	30.17	279.57	302.15	44.89
int	256	56.08	18.49	26.25	10.12	27.36	15.32	17.49	54.61	7.45	4.53	201.90	17.63	163.53	14.41	185.39	168.99	208.35	10.79	2.13	69.72	27.53	11.28	29.08	23.88	10.48	11.21	1.69	38.13	47.78	70.98	295.48	511.58	710.03	300.58	30.21	272.72	301.61	44.67
int	128	44.23	18.34	25.96	9.81	25.36	14.72	17.23	41.71	7.77	2.75	182.84	15.72	148.83	12.08	162.21	147.50	188.06	8.39	2.09	69.43	27.50	11.10	29.01	24.07	10.55	11.23	1.68	38.17	41.35	61.81	258.48	511.47	709.65	300.47	30.12	272.48	302.15	44.70
int	64	43.45	18.27	26.19	10.02	24.85	14.92	17.35	40.37	7.28	2.22	183.14	14.62	148.45	11.62	162.38	148.71	187.64	8.79	2.08	69.08	27.46	11.12	28.79	24.06	10.53	11.18	1.71	38.02	41.09	62.25	258.94	511.55	709.91	301.06	30.17	277.02	302.24	44.85
bin	8192	45.49	18.35	26.00	9.95	25.19	15.10	17.38	41.96	7.70	2.80	201.38	17.29	163.27	13.93	185.42	168.54	207.13	10.32	2.10	69.35	27.63	11.12	28.83	24.06	10.53	11.16	1.71	38.06	47.11	71.06	295.18	511.19	710.28	300.98	30.36	276.14	301.55	44.75
bin	4096	44.41	18.38	26.04	10.13	25.26	15.15	17.39	41.84	8.02	2.81	201.88	17.36	163.02	14.16	185.78	169.54	207.45	10.37	2.07	69.31	27.50	11.13	28.90	23.98	10.45	11.16	1.72	38.12	47.03	71.20	296.21	511.67	710.78	301.21	30.22	277.29	302.15	44.35
bin	2048	44.61	18.35	26.10	9.94	25.24	14.88	17.54	41.85	7.98	3.14	201.45	17.48	163.01	14.43	185.82	168.44	207.54	10.43	2.08	69.51	27.45	11.10	28.90	23.97	10.43	11.23	1.71	38.26	47.51	70.81	295.20	511.99	709.70	301.15	30.16	276.48	297.98	44.37
bin	1024	44.16	18.27	26.06	9.91	25.34	15.13	17.28	41.88	8.18	2.80	201.62	17.29	163.39	14.07	184.98	168.17	207.37	10.31	2.09	69.34	27.48	11.10	28.90	23.93	10.45	11.20	1.74	38.28	47.13	71.01	295.08	510.44	709.11	300.70	30.14	274.27	301.86	44.90
bin	512	43.21	18.48	26.36	9.87	24.77	14.59	17.57	40.36	6.06	2.10	201.96	17.26	162.69	13.74	184.97	168.38	207.11	10.25	2.09	69.13	27.54	11.11	28.94	23.93	10.45	11.17	1.71	38.18	46.94	71.16	294.22	511.18	710.02	300.74	30.21	271.81	301.24	44.85
bin	256	42.97	18.39	26.10	9.93	24.50	14.56	17.54	39.50	5.72	1.99	201.02	17.06	149.02	11.40	161.15	146.93	186.74	7.90	2.14	69.00	27.46	11.12	28.90	24.04	10.48	11.22	1.71	38.24	46.57	70.88	294.14	511.52	709.92	300.70	30.11	270.58	302.13	44.82
bin	128	42.50	17.96	25.56	10.09	24.14	14.45	17.24	39.36	6.66	1.97	182.34	14.49	148.03	11.23	161.86	146.45	185.74	8.20	2.08	69.03	27.42	11.16	28.86	24.02	10.49	11.17	1.69	38.33	40.40	61.98	258.54	511.12	709.85	300.77	30.17	271.70	302.19	44.65
bin	64	42.46	18.13	25.69	9.43	24.00	14.05	17.42	38.70	5.98	1.76	181.73	14.21	147.65	11.17	160.92	147.39	187.23	7.81	2.09	69.32	27.49	11.12	28.54	23.91	10.56	11.26	1.69	38.13	41.13	61.88	258.88	511.73	709.66	300.87	30.12	270.03	302.34	44.78

Table 15: Absolute evaluation time for stella_en_400M_v5 on all tasks. Table 2 serves as a legend.

q	m	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	8192	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	4096	0.968	1.015	0.998	0.969	0.967	1.004	0.986	0.997	0.904	0.798	0.999	0.983	1.013	0.995	1.057	1.022	0.994	0.987	1.020	1.000	0.997	1.004	1.004	0.989	0.990	0.988	0.955	0.989	0.990	0.987	1.000	1.108	1.115	1.101	0.982	0.977	0.998	0.993
f32	2048	0.971	1.011	0.998	0.972	0.970	1.006	1.013	0.991	0.860	0.799	1.000	1.005	1.023	1.057	1.061	1.027	1.010	1.027	1.027	1.001	0.998	1.003	1.003	0.986	0.987	0.985	0.970	0.989	1.000	0.988	0.997	1.109	1.115	1.100	0.983	0.970	0.999	0.991
f32	1024	0.994	1.014	0.999	0.976	0.988	1.007	0.996	1.016	0.869	0.804	0.998	1.003	1.020	1.018	1.059	1.022	1.000	1.000	1.037	1.001	0.998	1.003	0.999	0.986	0.993	0.994	0.962	0.996	0.999	0.985	0.997	1.107	1.113	1.098	0.984	0.956	1.000	0.990
f32	512	0.871	0.969	0.995	0.985	0.854	0.600	0.953	0.823	0.763	0.695	0.997	0.944	1.008	0.933	1.052	1.007	0.977	0.941	1.017	0.999	0.993	1.001	0.996	0.980	0.983	0.994	0.965	0.993	0.974	0.983	0.996	1.109	1.115	1.099	0.986	0.923	1.003	0.990
f32	256	0.809	0.958	0.985	0.982	0.588	0.564	0.914	0.743	0.638	0.398	0.991	0.923	1.000	0.904	1.041	0.991	0.962	0.886	1.008	0.994	0.993	0.996	1.001	0.988	0.987	0.992	0.949	0.994	0.952	0.982	0.995	1.109	1.114	1.099	0.984	0.903	1.001	0.993
f32	128	0.642	0.945	0.985	0.976	0.559	0.549	0.909	0.570	0.487	0.281	0.899	0.764	0.914	0.747	0.917	0.869	0.864	0.781	0.989	0.988	0.997	0.994	0.995	0.986	0.993	0.994	0.946	0.990	0.856	0.855	0.873	1.109	1.116	1.101	0.987	0.894	1.002	0.991
f32	64	0.630	0.931	0.973	0.996	0.542	0.547	0.896	0.551	0.517	0.228	0.896	0.754	0.915	0.768	0.906	0.858	0.856	0.687	0.972	0.985	0.998	0.988	0.999	0.994	0.991	0.992	0.953	0.993	0.824	0.856	0.872	1.110	1.115	1.098	0.987	0.896	0.998	0.984
int	8192	1.018	0.971	0.997	0.994	1.016	0.989	0.907	1.033	0.978	0.946	1.006	0.990	1.010	0.982	1.051	1.012	0.981	1.041	1.021	1.015	0.998	1.008	1.006	0.989	0.989	0.991	0.971	0.991	1.027	0.997	1.002	1.070	1.086	1.074	0.979	0.856	0.862	0.862
int	4096	0.997	0.982	0.999	0.982	1.017	1.000	0.937	1.035	0.970	0.898	0.999	0.986	1.013	0.978	1.055	1.010	0.977	1.033	1.033	1.012	0.997	1.006	1.003	0.991	0.987	0.991	0.947	0.997	1.024	0.998	1.003	0.966	0.971	0.962	0.855	0.852	0.862	0.858
int	2048	1.036	0.999	1.003	0.971	1.020	1.000	0.936	1.039	0.993	0.954	1.001	1.029	1.014	1.040	1.054	1.011	0.988	1.028	1.024	1.009	0.998	1.006	1.002	0.990	0.991	0.992	0.957	0.993	1.041	0.999	1.000	0.964	0.970	0.962	0.850	0.854	0.863	0.858
int	1024	1.016	0.991	0.984	0.970	1.006	1.001	0.944	1.029	0.969	0.915	0.997	1.003	1.014	0.991	1.056	1.013	0.982	1.017	1.014	1.004	0.995	0.997	1.000	0.989	0.991	0.989	0.946	0.992	1.032	0.997	1.001	0.963	0.970	0.962	0.850	0.845	0.839	0.841
int	512	0.882	0.958	0.981	0.988	0.861	0.599	0.917	0.833	0.666	0.615	0.996	0.948	1.006	0.912	1.050	1.000	0.962	0.939	1.014	1.000	0.992	0.986	0.998	0.988	0.985	0.989	0.972	0.987	0.990	0.988	1.000	0.965	0.970	0.961	0.852	0.824	0.861	0.857
int	256	0.814	0.943	0.978	1.012	0.602	0.571	0.911	0.746	0.544	0.479	0.991	0.910	0.998	0.870	1.038	0.984	0.951	0.912	1.000	0.995	0.996	1.005	1.003	0.984	0.987	0.987	0.948	0.991	0.967	0.980	0.996	0.966	0.970	0.961	0.853	0.804	0.860	0.853
int	128	0.642	0.935	0.968	0.981	0.557	0.548	0.897	0.570	0.567	0.291	0.897	0.812	0.908	0.729	0.908	0.859	0.859	0.709	0.982	0.991	0.995	0.989	1.000	0.992	0.993	0.989	0.941	0.992	0.837	0.853	0.871	0.966	0.970	0.961	0.850	0.803	0.861	0.854
int	64	0.631	0.931	0.976	1.002	0.546	0.556	0.904	0.551	0.531	0.235	0.899	0.755	0.906	0.702	0.909	0.866	0.857	0.743	0.975	0.986	0.994	0.991	0.993	0.991	0.991	0.984	0.957	0.988	0.831	0.859	0.872	0.966	0.970	0.962	0.851	0.817	0.861	0.857
bin	8192	0.660	0.935	0.969	0.995	0.554	0.563	0.905	0.573	0.562	0.296	0.988	0.892	0.997	0.841	1.038	0.981	0.946	0.872	0.983	0.990	1.000	0.991	0.994	0.991	0.991	0.982	0.958	0.989	0.953	0.981	0.995	0.965	0.970	0.962	0.857	0.814	0.859	0.855
bin	4096	0.645	0.937	0.971	1.013	0.555	0.565	0.906	0.572	0.586	0.297	0.991	0.896	0.995	0.855	1.040	0.987	0.947	0.877	0.972	0.989	0.995	0.992	0.997	0.988	0.983	0.982	0.966	0.990	0.952	0.983	0.998	0.966	0.971	0.963	0.853	0.817	0.861	0.847
bin	2048	0.648	0.935	0.973	0.994	0.555	0.554	0.913	0.572	0.582	0.332	0.989	0.902	0.995	0.871	1.040	0.981	0.947	0.881	0.977	0.992	0.993	0.989	0.997	0.987	0.982	0.988	0.957	0.994	0.961	0.977	0.995	0.967	0.970	0.963	0.851	0.815	0.849	0.847
bin	1024	0.641	0.932	0.972	0.991	0.557	0.564	0.900	0.572	0.597	0.297	0.989	0.892	0.997	0.849	1.036	0.979	0.947	0.871	0.978	0.990	0.995	0.989	0.997	0.986	0.984	0.986	0.975	0.994	0.954	0.980	0.994	0.964	0.969	0.961	0.851	0.808	0.860	0.858
bin	512	0.627	0.942	0.982	0.987	0.544	0.544	0.915	0.551	0.442	0.222	0.991	0.891	0.993	0.829	1.036	0.980	0.946	0.867	0.979	0.987	0.997	0.990	0.998	0.986	0.984	0.983	0.960	0.992	0.950	0.982	0.991	0.965	0.970	0.961	0.853	0.801	0.858	0.857
bin	256	0.624	0.938	0.973	0.993	0.539	0.542	0.913	0.540	0.418	0.210	0.987	0.881	0.910	0.688	0.902	0.855	0.853	0.668	1.003	0.985	0.994	0.991	0.997	0.990	0.986	0.988	0.958	0.994	0.942	0.978	0.991	0.966	0.970	0.961	0.850	0.798	0.861	0.856
bin	128	0.617	0.915	0.953	1.010	0.531	0.538	0.898	0.538	0.486	0.209	0.895	0.748	0.904	0.678	0.906	0.853	0.848	0.693	0.976	0.985	0.992	0.994	0.995	0.989	0.987	0.983	0.946	0.996	0.818	0.855	0.871	0.965	0.970	0.962	0.852	0.801	0.861	0.853
bin	64	0.617	0.924	0.958	0.943	0.527	0.523	0.907	0.529	0.436	0.186	0.892	0.733	0.901	0.674	0.901	0.858	0.855	0.660	0.981	0.989	0.995	0.991	0.984	0.985	0.994	0.991	0.947	0.991	0.832	0.854	0.872	0.966	0.970	0.962	0.850	0.796	0.862	0.855

Table 16: Relative evaluation time for stella_en_400M_v5 on all tasks. Table 2 serves as a legend.

q	m	Total Time	Average All Tasks	Average classification	Average clustering	Average sts	Average pairclass	Average retrieval	Average rerank	Average summ
f32	8192	4295.55	113.04	36.26	100.80	22.66	139.56	524.76	241.86	52.36
f32	4096	4463.78	117.47	35.74	101.87	22.57	139.06	582.55	238.88	51.97
f32	2048	4470.14	117.64	35.79	102.71	22.56	138.97	582.55	238.20	51.87
f32	1024	4461.36	117.40	36.29	102.21	22.60	138.89	581.40	236.73	51.84
f32	512	4393.43	115.62	31.13	100.51	22.52	138.27	582.46	233.35	51.84
f32	256	4342.03	114.26	28.06	98.83	22.51	137.80	582.05	230.84	52.01
f32	128	4156.98	109.39	24.78	88.11	22.44	121.08	582.73	229.96	51.86
f32	64	4143.28	109.03	24.33	87.41	22.46	120.48	582.54	229.79	51.54
int	8192	4325.80	113.84	36.46	101.71	22.70	140.11	565.73	209.12	45.12
int	4096	4143.34	109.04	36.43	101.50	22.69	140.22	507.54	207.28	44.93
int	2048	4151.41	109.25	36.85	102.07	22.65	140.19	507.15	207.49	44.92
int	1024	4130.42	108.70	36.44	101.67	22.59	140.12	506.92	203.72	44.04
int	512	4067.18	107.03	31.19	99.73	22.50	139.09	507.07	203.96	44.89
int	256	4019.45	105.78	28.21	98.30	22.51	138.08	507.40	201.51	44.67
int	128	3831.03	100.82	24.67	87.61	22.48	120.55	507.20	201.58	44.70
int	64	3833.36	100.88	24.43	87.48	22.40	120.76	507.51	203.14	44.85
bin	8192	3990.36	105.01	24.93	97.78	22.45	137.78	507.48	202.69	44.75
bin	4096	3995.47	105.14	24.83	98.04	22.44	138.15	507.88	203.22	44.35
bin	2048	3988.21	104.95	24.81	97.97	22.46	137.84	507.61	201.54	44.37
bin	1024	3985.40	104.88	24.76	97.82	22.45	137.74	506.75	202.09	44.90
bin	512	3976.34	104.64	24.40	97.45	22.43	137.44	507.31	201.09	44.85
bin	256	3888.11	102.32	24.19	88.89	22.43	137.20	507.38	200.94	44.82
bin	128	3813.89	100.37	23.91	86.70	22.42	120.31	507.25	201.36	44.65
bin	64	3811.24	100.30	23.73	86.58	22.41	120.63	507.42	200.83	44.78

Table 17: Evaluation time for stella_en_400M_v5, averaged per category. Table 2 serves as a legend.

q	m	Average Relative classification	Average Relative clustering	Average Relative sts	Average Relative pairclass	Average Relative retrieval	Average Relative rerank	Average Relative summ	Average Relative all
f32	8192	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	4096	0.988	0.975	0.994	0.992	1.108	0.986	0.993	0.996
f32	2048	0.992	0.987	0.995	0.995	1.108	0.984	0.991	1.000
f32	1024	0.999	0.980	0.997	0.994	1.106	0.980	0.990	1.000
f32	512	0.881	0.932	0.992	0.984	1.108	0.971	0.990	0.960
f32	256	0.818	0.873	0.990	0.976	1.107	0.963	0.993	0.929
f32	128	0.767	0.752	0.987	0.861	1.109	0.961	0.991	0.877
f32	64	0.758	0.739	0.987	0.851	1.108	0.961	0.984	0.870
int	8192	0.991	1.000	0.998	1.009	1.077	0.899	0.862	0.993
int	4096	0.994	0.992	0.996	1.008	0.966	0.856	0.858	0.979
int	2048	1.000	1.011	0.996	1.013	0.965	0.856	0.858	0.985
int	1024	0.993	0.996	0.992	1.010	0.965	0.845	0.841	0.977
int	512	0.877	0.909	0.991	0.993	0.965	0.846	0.857	0.929
int	256	0.822	0.868	0.989	0.981	0.966	0.839	0.853	0.904
int	128	0.762	0.754	0.986	0.854	0.965	0.838	0.854	0.851
int	64	0.762	0.740	0.985	0.854	0.966	0.843	0.857	0.847
bin	8192	0.769	0.841	0.987	0.976	0.966	0.844	0.855	0.885
bin	4096	0.770	0.847	0.986	0.977	0.967	0.844	0.847	0.887
bin	2048	0.768	0.852	0.986	0.978	0.966	0.839	0.847	0.887
bin	1024	0.766	0.845	0.987	0.976	0.965	0.840	0.858	0.886
bin	512	0.762	0.820	0.986	0.974	0.966	0.838	0.857	0.877
bin	256	0.758	0.737	0.989	0.971	0.966	0.836	0.856	0.855
bin	128	0.750	0.722	0.984	0.848	0.966	0.838	0.853	0.839
bin	64	0.741	0.710	0.985	0.853	0.966	0.836	0.855	0.834

Table 18: Relative evaluation time for stella_en_400M_v5, averaged per category. Table 2 serves as a legend.

q	m	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	8192	262144	0.876	0.565	0.607	0.742	0.796	0.835	0.965	0.837	0.628	0.614	0.438	0.411	0.395	0.381	0.590	0.589	0.430	0.521	0.868	0.804	0.758	0.843	0.807	0.872	0.850	0.861	0.901	0.698	0.939	0.767	0.871	0.564	0.454	0.412	0.621	0.324	0.509	0.306
f32	4096	131072	0.876	0.565	0.607	0.742	0.796	0.835	0.965	0.837	0.628	0.614	0.438	0.411	0.395	0.381	0.590	0.589	0.430	0.521	0.868	0.804	0.758	0.843	0.807	0.872	0.850	0.861	0.901	0.698	0.939	0.767	0.871	0.564	0.454	0.412	0.621	0.324	0.509	0.306
f32	2048	65536	0.876	0.565	0.607	0.742	0.796	0.835	0.965	0.837	0.628	0.614	0.438	0.411	0.395	0.381	0.590	0.589	0.430	0.521	0.868	0.804	0.758	0.843	0.807	0.872	0.850	0.861	0.901	0.698	0.939	0.767	0.871	0.564	0.454	0.412	0.621	0.324	0.509	0.306
f32	1024	32768	0.876	0.565	0.607	0.742	0.796	0.835	0.965	0.837	0.628	0.614	0.438	0.411	0.395	0.381	0.590	0.589	0.430	0.521	0.868	0.804	0.758	0.843	0.807	0.872	0.850	0.861	0.901	0.698	0.939	0.767	0.871	0.564	0.454	0.412	0.621	0.324	0.509	0.306
f32	512	16384	0.875	0.546	0.594	0.729	0.792	0.828	0.962	0.827	0.625	0.614	0.434	0.406	0.389	0.380	0.570	0.580	0.436	0.512	0.876	0.802	0.754	0.845	0.806	0.869	0.848	0.859	0.899	0.700	0.935	0.766	0.870	0.565	0.433	0.397	0.621	0.322	0.507	0.312
f32	256	8192	0.869	0.521	0.574	0.713	0.779	0.818	0.956	0.809	0.611	0.588	0.417	0.397	0.384	0.375	0.543	0.569	0.433	0.504	0.859	0.797	0.744	0.835	0.801	0.864	0.842	0.852	0.895	0.695	0.928	0.760	0.867	0.543	0.410	0.363	0.617	0.311	0.498	0.307
f32	128	4096	0.856	0.475	0.539	0.672	0.750	0.791	0.937	0.770	0.620	0.599	0.403	0.383	0.380	0.366	0.490	0.524	0.427	0.456	0.836	0.790	0.739	0.828	0.791	0.847	0.841	0.848	0.882	0.693	0.918	0.749	0.861	0.513	0.377	0.314	0.601	0.309	0.485	0.308
f32	64	2048	0.832	0.438	0.504	0.632	0.701	0.737	0.897	0.702	0.601	0.582	0.360	0.337	0.363	0.349	0.394	0.445	0.411	0.376	0.835	0.777	0.726	0.810	0.772	0.836	0.823	0.826	0.860	0.688	0.900	0.734	0.851	0.447	0.297	0.231	0.589	0.283	0.471	0.311
int	8192	65536	0.867	0.552	0.592	0.643	0.765	0.808	0.965	0.782	0.613	0.611	0.430	0.413	0.392	0.376	0.588	0.588	0.436	0.532	0.845	0.802	0.740	0.854	0.811	0.872	0.837	0.855	0.903	0.690	0.943	0.768	0.872	0.581	0.450	0.411	0.619	0.325	0.513	0.308
int	4096	32768	0.867	0.552	0.592	0.643	0.765	0.808	0.965	0.782	0.613	0.611	0.430	0.413	0.392	0.376	0.588	0.588	0.436	0.532	0.845	0.802	0.740	0.854	0.811	0.872	0.837	0.855	0.903	0.690	0.943	0.768	0.872	0.581	0.450	0.411	0.619	0.325	0.513	0.308
int	2048	16384	0.867	0.552	0.592	0.643	0.765	0.808	0.965	0.782	0.613	0.611	0.430	0.413	0.392	0.376	0.588	0.588	0.436	0.532	0.845	0.802	0.740	0.854	0.811	0.872	0.837	0.855	0.903	0.690	0.943	0.768	0.872	0.581	0.450	0.411	0.619	0.325	0.513	0.308
int	1024	8192	0.867	0.552	0.592	0.643	0.765	0.808	0.965	0.782	0.613	0.611	0.430	0.413	0.392	0.376	0.588	0.588	0.436	0.532	0.845	0.802	0.740	0.854	0.811	0.872	0.837	0.855	0.903	0.690	0.943	0.768	0.872	0.581	0.450	0.411	0.619	0.325	0.513	0.308
int	512	4096	0.864	0.532	0.574	0.639	0.758	0.800	0.962	0.763	0.618	0.604	0.425	0.405	0.392	0.380	0.574	0.581	0.438	0.511	0.849	0.800	0.740	0.855	0.808	0.868	0.835	0.852	0.902	0.696	0.940	0.766	0.871	0.583	0.435	0.405	0.611	0.323	0.512	0.311
int	256	2048	0.857	0.494	0.554	0.613	0.742	0.791	0.953	0.732	0.615	0.595	0.414	0.398	0.382	0.375	0.546	0.566	0.432	0.502	0.844	0.797	0.733	0.848	0.800	0.864	0.831	0.849	0.898	0.695	0.930	0.760	0.869	0.568	0.418	0.371	0.606	0.317	0.505	0.313
int	128	1024	0.835	0.424	0.513	0.626	0.702	0.750	0.922	0.677	0.613	0.594	0.404	0.384	0.375	0.368	0.496	0.521	0.421	0.445	0.838	0.792	0.718	0.839	0.794	0.857	0.828	0.844	0.891	0.690	0.918	0.749	0.863	0.539	0.372	0.341	0.604	0.314	0.490	0.320
int	64	512	0.799	0.375	0.472	0.605	0.636	0.680	0.855	0.579	0.582	0.570	0.364	0.336	0.359	0.346	0.392	0.448	0.411	0.372	0.813	0.779	0.710	0.821	0.776	0.840	0.807	0.817	0.863	0.690	0.901	0.733	0.852	0.463	0.299	0.289	0.586	0.295	0.474	0.309
bin	8192	8192	0.775	0.350	0.461	0.604	0.631	0.666	0.880	0.596	0.580	0.564	0.350	0.320	0.344	0.332	0.386	0.399	0.380	0.377	0.737	0.763	0.685	0.796	0.748	0.817	0.811	0.809	0.832	0.659	0.901	0.713	0.841	0.369	0.207	0.186	0.563	0.292	0.436	0.299
bin	4096	4096	0.775	0.350	0.461	0.604	0.631	0.666	0.880	0.596	0.580	0.564	0.350	0.320	0.344	0.332	0.386	0.399	0.380	0.377	0.737	0.763	0.685	0.796	0.748	0.817	0.811	0.809	0.832	0.659	0.901	0.713	0.841	0.369	0.207	0.186	0.563	0.292	0.436	0.299
bin	2048	2048	0.775	0.350	0.461	0.604	0.631	0.666	0.880	0.596	0.580	0.564	0.350	0.320	0.344	0.332	0.386	0.399	0.380	0.377	0.737	0.763	0.685	0.796	0.748	0.817	0.811	0.809	0.832	0.659	0.901	0.713	0.841	0.369	0.207	0.186	0.563	0.292	0.436	0.299
bin	1024	1024	0.775	0.350	0.461	0.604	0.631	0.666	0.880	0.596	0.580	0.564	0.350	0.320	0.344	0.332	0.386	0.399	0.380	0.377	0.737	0.763	0.685	0.796	0.748	0.817	0.811	0.809	0.832	0.659	0.901	0.713	0.841	0.369	0.207	0.186	0.563	0.292	0.436	0.299
bin	512	512	0.658	0.285	0.413	0.574	0.507	0.543	0.750	0.450	0.558	0.548	0.281	0.250	0.312	0.295	0.255	0.304	0.377	0.290	0.696	0.730	0.650	0.749	0.700	0.753	0.759	0.765	0.797	0.638	0.843	0.658	0.816	0.254	0.116	0.101	0.540	0.269	0.389	0.299
bin	256	256	0.470	0.253	0.402	0.547	0.341	0.393	0.586	0.303	0.524	0.528	0.212	0.187	0.289	0.265	0.164	0.256	0.364	0.188	0.672	0.675	0.579	0.699	0.632	0.704	0.700	0.690	0.697	0.609	0.724	0.592	0.774	0.127	0.043	0.064	0.516	0.256	0.334	0.298
bin	128	128	0.266	0.251	0.418	0.542	0.208	0.252	0.417	0.209	0.513	0.506	0.155	0.138	0.249	0.239	0.121	0.228	0.364	0.150	0.439	0.580	0.489	0.614	0.538	0.626	0.618	0.612	0.646	0.551	0.520	0.516	0.710	0.050	0.022	0.027	0.480	0.247	0.287	0.291
bin	64	64	0.218	0.256	0.412	0.530	0.159	0.237	0.345	0.136	0.506	0.500	0.118	0.106	0.244	0.234	0.109	0.216	0.352	0.127	0.367	0.440	0.395	0.447	0.409	0.531	0.504	0.483	0.487	0.496	0.285	0.430	0.613	0.035	0.008	0.024	0.455	0.235	0.247	0.276

Table 19: Absolute performance for stella_en_400M_v5 on all tasks. Table 2 serves as a legend.

q	m	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	8192	262144	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	4096	131072	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	2048	65536	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	1024	32768	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	512	16384	0.999	0.967	0.978	0.983	0.995	0.991	0.996	0.987	0.995	1.000	0.990	0.987	0.983	0.998	0.966	0.985	1.014	0.983	1.010	0.998	0.995	1.002	1.000	0.997	0.997	0.998	0.998	1.002	0.996	0.999	0.999	1.001	0.955	0.963	0.999	0.993	0.996	1.019
f32	256	8192	0.992	0.923	0.945	0.961	0.979	0.979	0.990	0.966	0.973	0.958	0.953	0.964	0.973	0.983	0.920	0.967	1.006	0.969	0.989	0.992	0.982	0.990	0.992	0.992	0.990	0.990	0.994	0.996	0.988	0.991	0.995	0.963	0.903	0.879	0.994	0.958	0.978	1.004
f32	128	4096	0.977	0.840	0.888	0.906	0.943	0.947	0.970	0.919	0.987	0.975	0.920	0.929	0.960	0.961	0.831	0.889	0.993	0.876	0.963	0.984	0.974	0.981	0.981	0.972	0.989	0.985	0.979	0.993	0.977	0.977	0.988	0.909	0.832	0.762	0.967	0.953	0.955	1.006
f32	64	2048	0.950	0.775	0.831	0.852	0.880	0.882	0.929	0.838	0.956	0.948	0.821	0.819	0.918	0.917	0.667	0.756	0.957	0.722	0.963	0.966	0.958	0.961	0.957	0.959	0.968	0.960	0.955	0.985	0.959	0.956	0.977	0.791	0.655	0.561	0.948	0.873	0.926	1.018
int	8192	65536	0.989	0.977	0.975	0.866	0.961	0.968	1.000	0.934	0.976	0.994	0.982	1.003	0.991	0.985	0.996	0.998	1.012	1.022	0.973	0.998	0.976	1.012	1.005	1.000	0.985	0.994	1.002	0.988	1.004	1.001	1.001	1.030	0.991	0.997	0.997	1.000	1.009	1.008
int	4096	32768	0.989	0.977	0.975	0.866	0.961	0.968	1.000	0.934	0.976	0.994	0.982	1.003	0.991	0.985	0.996	0.998	1.012	1.022	0.973	0.998	0.976	1.012	1.005	1.000	0.985	0.994	1.002	0.988	1.004	1.001	1.001	1.030	0.991	0.997	0.997	1.000	1.009	1.008
int	2048	16384	0.989	0.977	0.975	0.866	0.961	0.968	1.000	0.934	0.976	0.994	0.982	1.003	0.991	0.985	0.996	0.998	1.012	1.022	0.973	0.998	0.976	1.012	1.005	1.000	0.985	0.994	1.002	0.988	1.004	1.001	1.001	1.030	0.991	0.997	0.997	1.000	1.009	1.008
int	1024	8192	0.989	0.977	0.975	0.866	0.961	0.968	1.000	0.934	0.976	0.994	0.982	1.003	0.991	0.985	0.996	0.998	1.012	1.022	0.973	0.998	0.976	1.012	1.005	1.000	0.985	0.994	1.002	0.988	1.004	1.001	1.001	1.030	0.991	0.997	0.997	1.000	1.009	1.008
int	512	4096	0.987	0.941	0.945	0.861	0.952	0.958	0.996	0.911	0.983	0.983	0.970	0.983	0.991	0.997	0.973	0.986	1.017	0.982	0.979	0.996	0.976	1.014	1.001	0.996	0.982	0.990	1.002	0.997	1.001	0.999	1.000	1.034	0.960	0.982	0.984	0.997	1.007	1.018
int	256	2048	0.979	0.874	0.912	0.826	0.933	0.948	0.987	0.874	0.979	0.969	0.946	0.967	0.967	0.983	0.926	0.961	1.004	0.964	0.972	0.992	0.967	1.006	0.991	0.992	0.977	0.986	0.997	0.996	0.991	0.991	0.997	1.006	0.921	0.899	0.976	0.976	0.993	1.022
int	128	1024	0.954	0.751	0.846	0.843	0.882	0.898	0.956	0.808	0.976	0.966	0.922	0.932	0.950	0.967	0.840	0.885	0.979	0.855	0.966	0.985	0.947	0.995	0.984	0.983	0.973	0.980	0.989	0.988	0.977	0.977	0.990	0.955	0.820	0.827	0.972	0.967	0.964	1.046
int	64	512	0.912	0.665	0.777	0.815	0.799	0.815	0.886	0.691	0.927	0.929	0.832	0.817	0.909	0.907	0.665	0.761	0.956	0.716	0.936	0.969	0.936	0.974	0.962	0.964	0.949	0.949	0.959	0.988	0.960	0.956	0.978	0.820	0.659	0.700	0.943	0.908	0.931	1.010
bin	8192	8192	0.885	0.619	0.760	0.814	0.793	0.798	0.912	0.712	0.924	0.918	0.798	0.778	0.870	0.870	0.655	0.678	0.884	0.724	0.850	0.949	0.904	0.944	0.927	0.938	0.954	0.940	0.924	0.944	0.959	0.929	0.965	0.654	0.457	0.450	0.906	0.900	0.858	0.977
bin	4096	4096	0.885	0.619	0.760	0.814	0.793	0.798	0.912	0.712	0.924	0.918	0.798	0.778	0.870	0.870	0.655	0.678	0.884	0.724	0.850	0.949	0.904	0.944	0.927	0.938	0.954	0.940	0.924	0.944	0.959	0.929	0.965	0.654	0.457	0.450	0.906	0.900	0.858	0.977
bin	2048	2048	0.885	0.619	0.760	0.814	0.793	0.798	0.912	0.712	0.924	0.918	0.798	0.778	0.870	0.870	0.655	0.678	0.884	0.724	0.850	0.949	0.904	0.944	0.927	0.938	0.954	0.940	0.924	0.944	0.959	0.929	0.965	0.654	0.457	0.450	0.906	0.900	0.858	0.977
bin	1024	1024	0.885	0.619	0.760	0.814	0.793	0.798	0.912	0.712	0.924	0.918	0.798	0.778	0.870	0.870	0.655	0.678	0.884	0.724	0.850	0.949	0.904	0.944	0.927	0.938	0.954	0.940	0.924	0.944	0.959	0.929	0.965	0.654	0.457	0.450	0.906	0.900	0.858	0.977
bin	512	512	0.751	0.505	0.681	0.774	0.638	0.651	0.776	0.538	0.888	0.893	0.641	0.607	0.789	0.775	0.432	0.516	0.876	0.558	0.802	0.908	0.857	0.888	0.868	0.864	0.892	0.888	0.884	0.914	0.898	0.858	0.937	0.450	0.256	0.245	0.870	0.829	0.764	0.977
bin	256	256	0.537	0.448	0.662	0.738	0.429	0.470	0.607	0.362	0.834	0.859	0.484	0.456	0.732	0.696	0.277	0.434	0.847	0.362	0.774	0.840	0.763	0.829	0.784	0.808	0.823	0.801	0.774	0.873	0.770	0.772	0.888	0.225	0.094	0.155	0.832	0.789	0.656	0.975
bin	128	128	0.303	0.445	0.689	0.730	0.261	0.302	0.432	0.250	0.816	0.824	0.355	0.334	0.629	0.627	0.204	0.387	0.846	0.288	0.505	0.721	0.646	0.728	0.667	0.718	0.727	0.711	0.717	0.789	0.554	0.673	0.815	0.089	0.048	0.066	0.774	0.761	0.565	0.951
bin	64	64	0.249	0.453	0.679	0.715	0.200	0.284	0.357	0.163	0.806	0.814	0.270	0.257	0.618	0.615	0.185	0.367	0.819	0.244	0.422	0.548	0.521	0.529	0.507	0.610	0.593	0.561	0.541	0.711	0.304	0.560	0.704	0.062	0.018	0.058	0.732	0.725	0.487	0.904

Table 20: Relative performance for stella_en_400M_v5 on all tasks. Table 2 serves as a legend.

q	m	req. bits	Average classification	Average clustering	Average sts	Average pairclass	Average retrieval	Average rerank	Average summ	Average all
f32	8192	262144	0.778	0.500	0.826	0.859	0.477	0.485	0.306	0.664
f32	4096	131072	0.778	0.500	0.826	0.859	0.477	0.485	0.306	0.664
f32	2048	65536	0.778	0.500	0.826	0.859	0.477	0.485	0.306	0.664
f32	1024	32768	0.778	0.500	0.826	0.859	0.477	0.485	0.306	0.664
f32	512	16384	0.769	0.495	0.826	0.857	0.465	0.483	0.312	0.660
f32	256	8192	0.755	0.482	0.818	0.852	0.439	0.475	0.307	0.649
f32	128	4096	0.724	0.465	0.809	0.843	0.401	0.465	0.308	0.631
f32	64	2048	0.680	0.422	0.795	0.828	0.325	0.448	0.311	0.598
int	8192	65536	0.747	0.498	0.821	0.861	0.481	0.486	0.308	0.657
int	4096	32768	0.747	0.498	0.821	0.861	0.481	0.486	0.308	0.657
int	2048	16384	0.747	0.498	0.821	0.861	0.481	0.486	0.308	0.657
int	1024	8192	0.747	0.498	0.821	0.861	0.481	0.486	0.308	0.657
int	512	4096	0.736	0.493	0.821	0.859	0.474	0.482	0.311	0.652
int	256	2048	0.717	0.482	0.816	0.853	0.452	0.476	0.313	0.641
int	128	1024	0.681	0.462	0.809	0.843	0.417	0.469	0.320	0.623
int	64	512	0.625	0.418	0.792	0.829	0.350	0.451	0.309	0.587
bin	8192	8192	0.620	0.403	0.766	0.818	0.254	0.430	0.299	0.565
bin	4096	4096	0.620	0.403	0.766	0.818	0.254	0.430	0.299	0.565
bin	2048	2048	0.620	0.403	0.766	0.818	0.254	0.430	0.299	0.565
bin	1024	1024	0.620	0.403	0.766	0.818	0.254	0.430	0.299	0.565
bin	512	512	0.523	0.347	0.724	0.772	0.157	0.399	0.299	0.505
bin	256	256	0.412	0.298	0.666	0.696	0.078	0.369	0.298	0.438
bin	128	128	0.320	0.266	0.571	0.582	0.033	0.338	0.291	0.371
bin	64	64	0.287	0.251	0.456	0.443	0.022	0.312	0.276	0.315

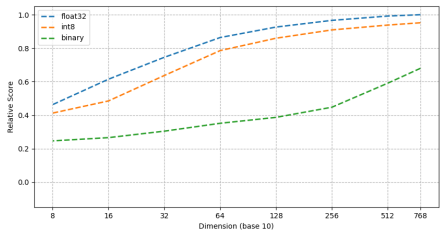
Table 21: Performance for stella_en_400M_v5, averaged per category. Table 2 serves as a legend.

q	m	req. bits	Average Relative classification	Average Relative clustering	Average Relative sts	Average Relative pairclass	Average Relative retrieval	Average Relative rerank	Average Relative summ	Average Relative all
f32	8192	262144	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	4096	131072	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	2048	65536	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	1024	32768	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	512	16384	0.987	0.990	1.000	0.998	0.973	0.996	1.019	0.992
f32	256	8192	0.967	0.966	0.991	0.992	0.915	0.977	1.004	0.973
f32	128	4096	0.924	0.932	0.980	0.981	0.834	0.958	1.006	0.943
f32	64	2048	0.867	0.848	0.963	0.964	0.669	0.916	1.018	0.887
int	8192	65536	0.959	0.996	0.993	1.002	1.006	1.002	1.008	0.990
int	4096	32768	0.959	0.996	0.993	1.002	1.006	1.002	1.008	0.990
int	2048	16384	0.959	0.996	0.993	1.002	1.006	1.002	1.008	0.990
int	1024	8192	0.959	0.996	0.993	1.002	1.006	1.002	1.008	0.990
int	512	4096	0.944	0.987	0.993	1.000	0.992	0.996	1.018	0.982
int	256	2048	0.916	0.967	0.988	0.993	0.942	0.982	1.022	0.964
int	128	1024	0.867	0.927	0.979	0.981	0.867	0.968	1.046	0.934
int	64	512	0.795	0.842	0.959	0.965	0.727	0.927	1.010	0.875
bin	8192	8192	0.786	0.810	0.927	0.951	0.520	0.888	0.977	0.835
bin	4096	4096	0.786	0.810	0.927	0.951	0.520	0.888	0.977	0.835
bin	2048	2048	0.786	0.810	0.927	0.951	0.520	0.888	0.977	0.835
bin	1024	1024	0.786	0.810	0.927	0.951	0.520	0.888	0.977	0.835
bin	512	512	0.664	0.698	0.877	0.897	0.317	0.821	0.977	0.741
bin	256	256	0.532	0.598	0.807	0.810	0.158	0.759	0.975	0.644
bin	128	128	0.426	0.531	0.693	0.681	0.068	0.700	0.951	0.551
bin	64	64	0.387	0.499	0.554	0.523	0.046	0.648	0.904	0.479

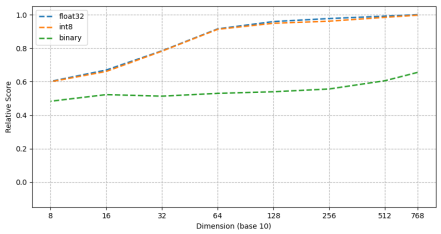
Table 22: Relative performance for stella_en_400M_v5, averaged per category. Table 2 serves as a legend.

E. nomic-embed-text-v1.5

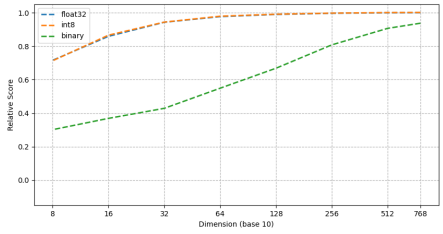
This section contains additional figures and tables for the NOMIC model.



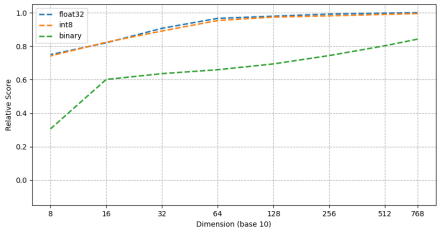
(a) Classification



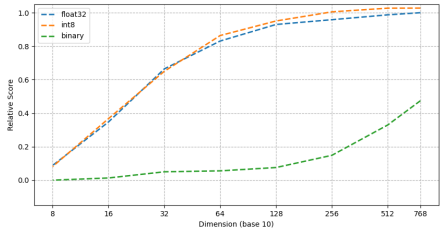
(b) Clustering



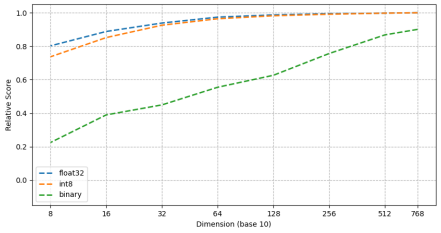
(c) Pair Classification



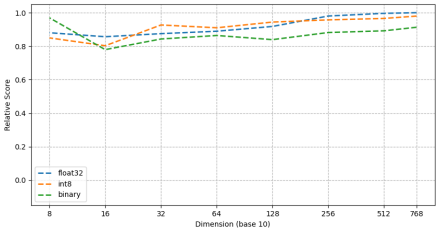
(d) Re-Ranking



(e) Retrieval

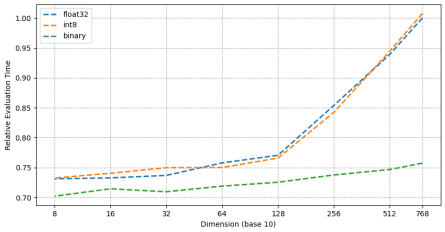


(f) Semantic Textual Similarity

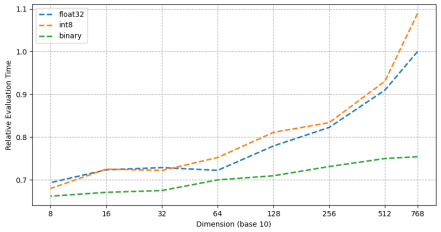


(g) Summarization

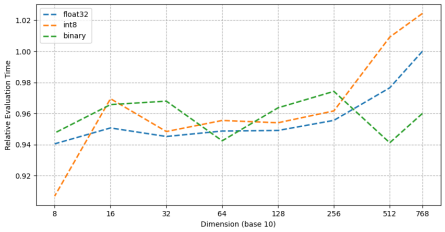
Figure 13: Relative performance of the NOMIC model for each task category in the MTEB. The x-axis of each subfigure represents the embedding size while the y-axis represents the relative performance. For each task the reference point is the performance of the float32 embeddings at the full embedding size. Blue: float32, Orange: int8, Green: binary.



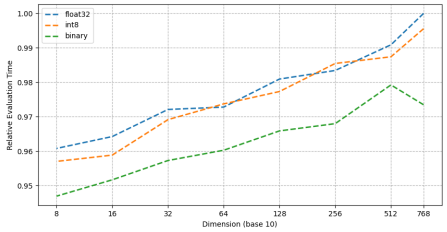
(a) Classification



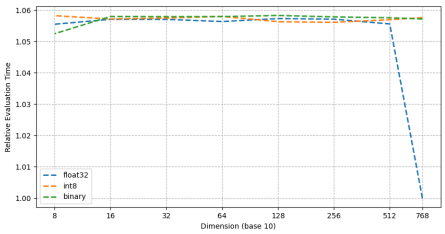
(b) Clustering



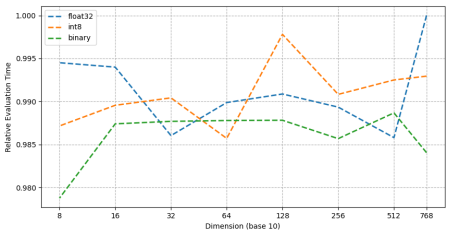
(c) Pair Classification



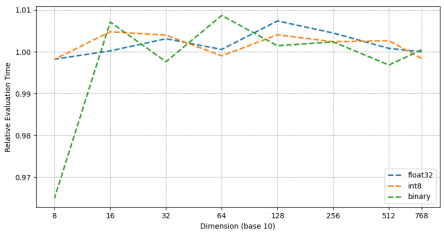
(d) Re-Ranking



(e) Retrieval

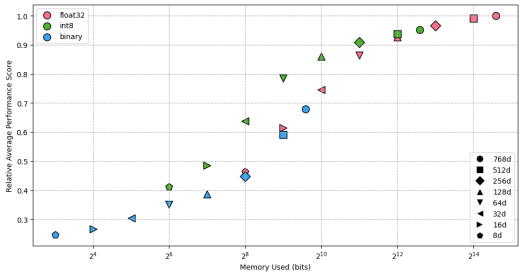


(f) Semantic Textual Similarity

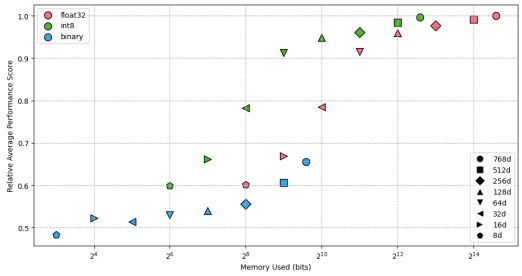


(g) Summarization

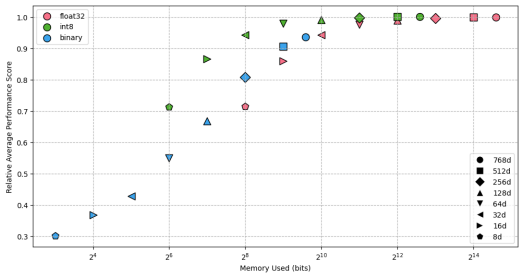
Figure 14: Relative evaluation time of the NOMIC model for each task category in the MTEB. The x-axis of each subfigure represents the embedding size while the y-axis represents the relative evaluation time. For each task the reference point is the evaluation time of the float32 embeddings at the full embedding size. Blue: float32, Orange: int8, Green: binary.



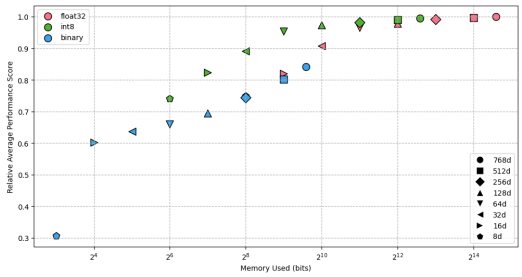
(a) Classification



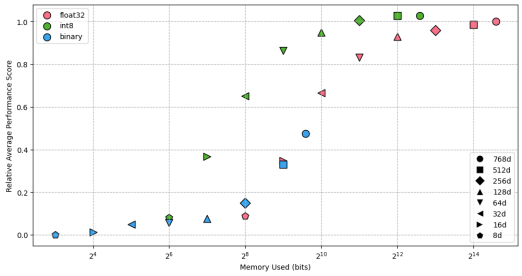
(b) Clustering



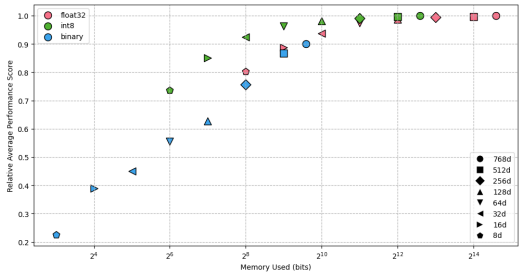
(c) Pair Classification



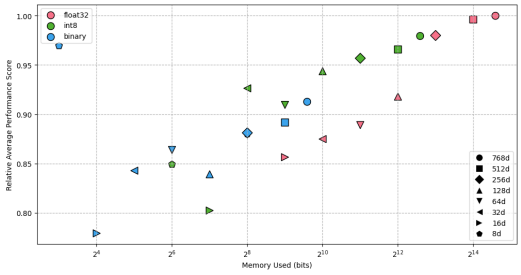
(d) Re-Ranking



(e) Retrieval



(f) Semantic Textual Similarity



(g) Summarization

Figure 15: Relative accuracy compute trade-off of the NOMIC model, averaged over all selected MTEB tasks within each category. The x-axis represents memory consumption (log scale) and the y-axis represents relative performance (averaged over tasks within the category). Reference point: float32 embeddings at full size.

q	m	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	768	38.06	9.66	12.20	4.93	29.61	9.07	8.47	48.01	8.54	6.08	90.87	10.75	NaN	10.95	110.43	108.35	135.25	6.59	1.08	25.43	11.54	4.51	12.03	9.71	4.22	4.59	0.73	NaN	20.74	25.66	109.91	212.30	NaN	NaN	14.33	191.40	136.15	17.58
f32	512	35.98	8.59	13.02	5.16	26.42	7.74	8.16	41.07	7.98	4.71	91.55	8.89	NaN	9.24	108.48	105.97	130.38	5.78	1.08	25.34	11.44	4.50	11.99	9.61	4.14	4.54	0.68	NaN	19.77	25.26	109.05	224.11	NaN	NaN	14.21	187.05	136.67	17.59
f32	256	31.52	8.90	13.38	5.09	14.17	7.73	7.80	33.62	5.69	3.11	89.92	8.99	NaN	7.22	106.85	100.95	125.70	6.00	1.11	25.30	11.51	4.57	11.99	9.65	4.19	4.48	0.67	NaN	18.27	25.66	108.40	224.43	NaN	NaN	14.12	184.30	136.44	17.66
f32	128	21.18	8.48	13.04	5.13	13.04	7.88	7.48	20.54	4.80	3.93	90.19	7.35	NaN	7.06	106.46	101.35	121.23	4.51	1.09	25.46	11.45	4.57	11.96	9.64	4.16	4.49	0.69	NaN	17.88	25.54	108.84	224.48	NaN	NaN	14.11	183.27	136.19	17.71
f32	64	20.09	8.66	13.24	5.09	12.26	7.55	7.48	18.74	4.41	1.95	89.52	7.39	NaN	6.15	106.15	98.89	121.16	4.33	1.09	25.38	11.47	4.55	12.03	9.75	4.17	4.50	0.68	NaN	17.91	25.32	109.45	224.27	NaN	NaN	14.07	179.29	136.08	17.59
f32	32	20.01	8.11	12.67	5.09	11.55	6.65	7.97	18.91	4.83	1.96	89.11	7.86	NaN	5.97	105.86	98.68	121.33	4.25	1.10	25.11	11.34	4.55	11.99	9.73	4.17	4.60	0.65	NaN	17.87	25.31	108.56	224.42	NaN	NaN	14.07	178.99	136.01	17.63
f32	16	19.58	8.22	12.49	5.08	11.84	7.18	7.38	18.27	4.00	2.03	89.79	6.92	NaN	6.36	105.77	100.81	119.32	4.77	1.13	25.44	11.49	4.58	11.97	9.62	4.20	4.48	0.68	NaN	18.42	25.11	108.34	224.43	NaN	NaN	14.15	173.60	135.90	17.58
f32	8	19.28	8.22	12.56	5.01	11.76	7.27	7.40	17.91	3.58	1.80	89.06	6.64	NaN	5.96	105.34	98.63	119.15	4.20	1.13	25.40	11.47	4.57	11.95	9.70	4.21	4.49	0.68	NaN	17.88	25.01	108.25	224.09	NaN	NaN	14.08	172.49	135.92	17.55
int	768	40.46	8.68	13.12	5.22	30.40	8.36	8.13	50.74	10.19	7.95	93.44	12.31	NaN	11.15	111.96	112.86	134.58	6.96	1.09	25.62	11.56	4.56	12.16	9.75	4.21	4.61	0.65	NaN	22.19	25.63	110.39	224.54	NaN	NaN	14.16	190.04	136.93	17.55
int	512	36.54	8.26	12.77	5.52	26.53	7.93	8.00	41.24	9.14	4.70	91.31	9.06	NaN	9.56	108.33	103.99	126.98	6.13	1.14	25.78	11.48	4.56	12.08	9.64	4.23	4.49	0.65	NaN	21.46	25.75	108.76	224.40	NaN	NaN	14.11	186.44	136.60	17.62
int	256	31.76	8.39	13.12	5.16	13.28	7.31	8.08	33.83	6.22	3.35	90.02	8.92	NaN	7.23	107.58	101.45	122.34	6.11	1.10	25.40	11.44	4.53	12.00	9.67	4.17	4.54	0.69	NaN	18.58	25.71	108.50	224.22	NaN	NaN	14.16	184.97	136.38	17.62
int	128	22.14	8.15	12.72	5.52	12.97	6.83	7.76	20.71	6.57	2.81	89.72	7.63	NaN	7.69	107.33	101.50	120.26	5.73	1.14	25.41	11.55	4.60	11.95	9.71	4.22	4.50	0.68	NaN	18.26	25.64	108.01	224.26	NaN	NaN	14.05	181.65	136.47	17.65
int	64	20.58	8.23	12.76	5.27	11.89	6.78	8.02	19.05	5.98	2.11	89.48	7.94	NaN	6.10	106.26	98.85	121.88	4.39	1.10	25.18	11.38	4.52	12.06	9.72	4.19	4.60	0.64	NaN	18.59	25.29	108.26	224.62	NaN	NaN	14.14	178.82	136.13	17.56
int	32	19.85	8.18	12.69	5.61	11.76	6.65	7.89	18.78	4.94	1.68	89.20	7.77	NaN	5.80	105.68	98.20	118.95	4.37	1.07	25.44	11.53	4.52	12.05	9.73	4.14	4.52	0.70	NaN	18.05	25.31	108.65	224.51	NaN	NaN	14.06	177.39	136.11	17.65
int	16	19.74	8.12	12.72	5.17	11.65	7.36	7.53	18.17	4.54	2.25	90.45	7.40	NaN	6.06	106.81	98.11	119.29	4.16	1.08	25.52	11.39	4.55	11.94	9.70	4.29	4.53	0.66	NaN	19.53	25.23	108.10	224.45	NaN	NaN	14.02	172.07	136.04	17.66
int	8	19.77	7.96	12.53	5.00	11.33	7.49	7.53	18.19	3.89	1.80	89.11	6.74	NaN	5.37	101.70	96.58	116.52	3.90	1.07	25.22	11.50	4.49	11.94	9.69	4.15	4.60	0.68	NaN	16.83	23.81	107.86	224.67	NaN	NaN	14.09	170.32	135.85	17.55
bin	768	21.01	8.05	12.70	5.34	12.77	7.41	7.51	19.87	4.74	2.48	89.77	8.07	NaN	6.23	106.35	99.63	119.75	4.93	1.11	25.40	11.42	4.53	11.97	9.67	4.25	4.48	0.63	NaN	18.98	25.26	107.78	224.45	NaN	NaN	14.12	179.07	136.10	17.59
bin	512	20.05	8.34	13.02	5.13	11.90	7.39	7.41	18.50	4.25	2.58	90.12	7.19	NaN	6.79	105.56	99.43	119.13	5.21	1.14	25.38	11.54	4.59	11.91	9.65	4.21	4.47	0.64	NaN	17.83	25.38	107.12	224.53	NaN	NaN	14.15	181.70	136.28	17.52
bin	256	20.23	8.08	12.88	5.09	11.42	7.20	7.40	18.75	4.10	2.88	89.82	6.98	NaN	6.46	105.83	99.68	119.25	4.21	1.09	25.53	11.42	4.53	11.96	9.70	4.19	4.57	0.64	NaN	19.44	25.87	107.41	224.59	NaN	NaN	13.99	177.92	135.87	17.62
bin	128	20.05	7.88	12.33	5.05	11.24	7.14	7.37	18.68	4.21	1.84	89.18	6.97	NaN	6.12	105.53	99.55	118.58	4.27	1.11	25.58	11.53	4.53	11.99	9.67	4.25	4.52	0.64	NaN	18.77	25.86	107.53	224.69	NaN	NaN	14.11	174.65	136.20	17.60
bin	64	19.66	7.80	12.08	4.96	11.21	6.65	7.93	18.36	4.00	1.64	89.15	7.32	NaN	5.60	105.45	97.84	118.35	4.30	1.09	25.29	11.55	4.50	12.00	9.71	4.17	4.57	0.66	NaN	17.90	24.96	108.98	224.61	NaN	NaN	14.04	172.75	135.94	17.73
bin	32	19.83	7.81	12.17	5.03	11.13	6.72	7.06	18.14	3.57	1.27	88.92	6.57	NaN	5.61	105.92	99.53	118.24	3.90	1.09	25.49	11.43	4.53	12.04	9.73	4.27	4.51	0.64	NaN	19.52	25.17	107.92	224.60	NaN	NaN	14.16	169.32	136.00	17.54
bin	16	20.07	7.87	12.40	4.98	11.40	6.96	6.95	18.05	3.32	1.46	89.00	6.58	NaN	5.58	105.93	97.83	117.85	3.77	1.10	25.57	11.41	4.56	12.03	9.72	4.26	4.47	0.64	NaN	19.24	25.18	108.62	224.62	NaN	NaN	14.09	167.04	136.02	17.70
bin	8	19.15	7.64	12.53	5.01	11.16	6.46	6.98	17.56	3.71	1.01	88.69	6.70	NaN	5.04	104.66	97.85	117.51	3.79	1.08	25.02	11.40	4.50	12.01	9.67	4.17	4.51	0.64	NaN	18.44	25.00	107.66	223.45	NaN	NaN	14.08	164.87	135.74	16.96

Table 23: Absolute evaluation time for nomic-embed-text-v1.5 on all tasks. Table 2 serves as a legend.

q	m	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	768	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	nan	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	nan	1.000	1.000	1.000	1.000	nan	nan	1.000	1.000	1.000	1.000
f32	512	0.945	0.890	1.067	1.047	0.892	0.853	0.963	0.856	0.934	0.774	1.008	0.827	nan	0.843	0.982	0.978	0.964	0.877	0.999	0.997	0.992	1.000	0.997	0.990	0.979	0.988	0.931	nan	0.953	0.984	0.992	1.056	nan	nan	0.992	0.977	1.004	1.001
f32	256	0.828	0.921	1.097	1.033	0.479	0.852	0.921	0.700	0.666	0.512	0.990	0.837	nan	0.659	0.968	0.932	0.929	0.911	1.023	0.995	0.998	1.014	0.997	0.994	0.991	0.975	0.917	nan	0.881	1.000	0.986	1.057	nan	nan	0.985	0.963	1.002	1.004
f32	128	0.556	0.878	1.069	1.041	0.441	0.870	0.882	0.428	0.562	0.647	0.993	0.684	nan	0.645	0.964	0.935	0.896	0.685	1.010	1.001	0.992	1.015	0.994	0.993	0.984	0.978	0.951	nan	0.862	0.995	0.990	1.057	nan	nan	0.985	0.958	1.000	1.007
f32	64	0.528	0.897	1.086	1.032	0.414	0.833	0.883	0.390	0.516	0.321	0.985	0.688	nan	0.561	0.961	0.913	0.896	0.658	1.004	0.998	0.994	1.010	1.000	1.003	0.988	0.980	0.932	nan	0.864	0.987	0.996	1.056	nan	nan	0.982	0.937	1.000	1.001
f32	32	0.526	0.840	1.039	1.033	0.390	0.734	0.940	0.394	0.566	0.323	0.981	0.731	nan	0.545	0.959	0.911	0.897	0.645	1.019	0.988	0.983	1.010	0.996	1.001	0.989	1.002	0.886	nan	0.862	0.986	0.988	1.057	nan	nan	0.982	0.935	0.999	1.003
f32	16	0.515	0.851	1.024	1.030	0.400	0.792	0.870	0.381	0.468	0.335	0.988	0.644	nan	0.581	0.958	0.930	0.882	0.725	1.047	1.000	0.996	1.016	0.994	0.990	0.996	0.976	0.931	nan	0.888	0.978	0.986	1.057	nan	nan	0.988	0.907	0.998	1.000
f32	8	0.507	0.851	1.030	1.017	0.397	0.802	0.874	0.373	0.419	0.296	0.980	0.618	nan	0.544	0.954	0.910	0.881	0.637	1.050	0.999	0.994	1.014	0.993	0.998	0.996	0.977	0.929	nan	0.862	0.975	0.985	1.056	nan	nan	0.983	0.901	0.998	0.998
int	768	1.063	0.898	1.075	1.059	1.027	0.923	0.959	1.057	1.193	1.309	1.028	1.145	nan	1.018	1.014	1.042	0.995	1.057	1.009	1.008	1.002	1.012	1.010	1.004	0.997	1.004	0.892	nan	1.070	0.999	1.004	1.058	nan	nan	0.988	0.993	1.006	0.998
int	512	0.960	0.855	1.047	1.121	0.896	0.875	0.944	0.859	1.070	0.774	1.005	0.843	nan	0.873	0.981	0.960	0.939	0.931	1.050	1.014	0.995	1.012	1.004	0.993	1.001	0.978	0.887	nan	1.035	1.003	0.990	1.057	nan	nan	0.985	0.974	1.003	1.003
int	256	0.835	0.868	1.076	1.047	0.449	0.807	0.954	0.705	0.729	0.552	0.991	0.830	nan	0.660	0.974	0.936	0.905	0.927	1.015	0.999	0.992	1.004	0.997	0.996	0.987	0.988	0.940	nan	0.896	1.002	0.987	1.056	nan	nan	0.988	0.966	1.002	1.002
int	128	0.582	0.844	1.043	1.121	0.438	0.753	0.916	0.431	0.769	0.462	0.987	0.710	nan	0.702	0.972	0.937	0.889	0.870	1.056	1.000	1.001	1.020	0.993	0.999	1.000	0.979	0.932	nan	0.880	0.999	0.983	1.056	nan	nan	0.981	0.949	1.002	1.004
int	64	0.541	0.852	1.046	1.070	0.402	0.748	0.946	0.397	0.700	0.346	0.985	0.739	nan	0.557	0.962	0.912	0.901	0.667	1.014	0.991	0.986	1.004	1.003	1.001	0.992	1.002	0.879	nan	0.896	0.985	0.985	1.058	nan	nan	0.987	0.934	1.000	0.999
int	32	0.521	0.846	1.041	1.137	0.397	0.734	0.930	0.391	0.578	0.277	0.982	0.723	nan	0.530	0.957	0.906	0.879	0.664	0.988	1.000	1.000	1.002	1.002	1.002	0.981	0.984	0.954	nan	0.870	0.986	0.989	1.058	nan	nan	0.981	0.927	1.000	1.004
int	16	0.519	0.841	1.043	1.050	0.394	0.812	0.888	0.378	0.532	0.370	0.995	0.688	nan	0.553	0.967	0.905	0.882	0.631	0.999	1.004	0.987	1.010	0.993	0.999	1.017	0.987	0.910	nan	0.942	0.983	0.984	1.057	nan	nan	0.978	0.899	0.999	1.005
int	8	0.519	0.824	1.027	1.015	0.383	0.826	0.888	0.379	0.456	0.296	0.981	0.627	nan	0.490	0.921	0.891	0.862	0.593	0.992	0.992	0.997	0.996	0.992	0.998	0.982	1.001	0.934	nan	0.811	0.928	0.981	1.058	nan	nan	0.983	0.890	0.998	0.998
bin	768	0.552	0.834	1.042	1.083	0.431	0.818	0.886	0.414	0.554	0.409	0.988	0.750	nan	0.569	0.963	0.919	0.885	0.748	1.023	0.999	0.990	1.004	0.995	0.996	1.006	0.975	0.869	nan	0.915	0.984	0.981	1.057	nan	nan	0.985	0.936	1.000	1.000
bin	512	0.527	0.863	1.067	1.040	0.402	0.816	0.875	0.385	0.498	0.424	0.992	0.669	nan	0.620	0.956	0.918	0.881	0.791	1.052	0.998	1.000	1.019	0.990	0.994	0.996	0.974	0.875	nan	0.860	0.989	0.975	1.058	nan	nan	0.987	0.949	1.001	0.997
bin	256	0.532	0.837	1.056	1.032	0.386	0.795	0.873	0.391	0.480	0.473	0.988	0.649	nan	0.590	0.958	0.920	0.882	0.638	1.009	1.004	0.990	1.005	0.994	0.999	0.993	0.995	0.884	nan	0.937	1.008	0.977	1.058	nan	nan	0.977	0.930	0.998	1.002
bin	128	0.527	0.815	1.011	1.025	0.380	0.788	0.870	0.389	0.493	0.302	0.981	0.649	nan	0.559	0.956	0.919	0.877	0.649	1.024	1.006	0.999	1.006	0.997	0.996	1.006	0.985	0.871	nan	0.905	1.008	0.978	1.058	nan	nan	0.985	0.912	1.000	1.001
bin	64	0.517	0.808	0.991	1.007	0.379	0.733	0.935	0.383	0.469	0.270	0.981	0.681	nan	0.511	0.955	0.903	0.875	0.653	1.005	0.995	1.001	0.999	0.997	1.000	0.986	0.996	0.912	nan	0.863	0.973	0.992	1.058	nan	nan	0.980	0.903	0.999	1.009
bin	32	0.521	0.808	0.998	1.021	0.376	0.742	0.834	0.378	0.418	0.209	0.979	0.611	nan	0.512	0.959	0.919	0.874	0.593	1.009	1.003	0.991	1.006	1.001	1.002	1.012	0.983	0.884	nan	0.941	0.981	0.982	1.058	nan	nan	0.988	0.885	0.999	0.998
bin	16	0.527	0.814	1.016	1.010	0.385	0.767	0.820	0.376	0.389	0.240	0.979	0.612	nan	0.510	0.959	0.903	0.871	0.572	1.019	1.006	0.989	1.012	1.000	1.001	1.010	0.973	0.878	nan	0.928	0.981	0.988	1.058	nan	nan	0.983	0.873	0.999	1.007
bin	8	0.503	0.791	1.028	1.018	0.377	0.713	0.823	0.366	0.434	0.166	0.976	0.624	nan	0.460	0.948	0.903	0.869	0.576	1.001	0.984	0.988	0.999	0.998	0.996	0.987	0.983	0.872	nan	0.889	0.974	0.980	1.052	nan	nan	0.982	0.861	0.997	0.965

Table 24: Relative evaluation time for nomic-embed-text-v1.5 on all tasks. Table 2 serves as a legend.

q	m	Total Time	Average All Tasks	Average classification	Average clustering	Average sts	Average pairclass	Average retrieval	Average rerank	Average summ
f32	768	1449.71	42.64	20.00	54.20	8.20	52.10	212.30	113.96	17.58
f32	512	1426.14	41.95	18.27	52.55	8.15	51.36	224.11	112.64	17.59
f32	256	1379.38	40.57	15.28	50.49	8.16	50.78	224.43	111.62	17.66
f32	128	1345.21	39.57	12.10	49.66	8.17	50.75	224.48	111.19	17.71
f32	64	1330.67	39.14	11.64	48.88	8.18	50.90	224.27	109.81	17.59
f32	32	1326.94	39.03	11.37	48.87	8.14	50.58	224.42	109.69	17.63
f32	16	1320.92	38.85	11.25	48.86	8.18	50.62	224.43	107.88	17.58
f32	8	1312.62	38.61	11.18	48.26	8.18	50.38	224.09	107.50	17.55
int	768	1482.13	43.59	20.64	55.71	8.24	52.74	224.54	113.71	17.55
int	512	1425.17	41.92	18.35	52.13	8.23	51.99	224.40	112.39	17.62
int	256	1377.82	40.52	15.12	50.36	8.17	50.93	224.22	111.84	17.62
int	128	1345.79	39.58	12.10	49.92	8.19	50.64	224.26	110.72	17.65
int	64	1332.37	39.19	11.57	49.22	8.16	50.71	224.62	109.70	17.56
int	32	1323.42	38.92	11.43	48.51	8.19	50.67	224.51	109.18	17.65
int	16	1320.31	38.83	11.31	48.79	8.19	50.95	224.45	107.38	17.66
int	8	1299.73	38.23	11.22	47.29	8.15	49.50	224.67	106.75	17.55
bin	768	1333.40	39.22	11.83	49.10	8.16	50.67	224.45	109.76	17.59
bin	512	1330.04	39.12	11.47	48.92	8.17	50.11	224.53	110.71	17.52
bin	256	1326.59	39.02	11.38	48.80	8.18	50.91	224.59	109.26	17.62
bin	128	1319.24	38.80	11.22	48.47	8.20	50.72	224.69	108.32	17.60
bin	64	1312.76	38.61	11.08	48.18	8.17	50.62	224.61	107.58	17.73
bin	32	1309.42	38.51	10.99	48.17	8.19	50.87	224.60	106.49	17.54
bin	16	1306.26	38.42	11.08	47.92	8.20	51.01	224.62	105.72	17.70
bin	8	1294.65	38.08	10.81	47.66	8.11	50.36	223.45	104.90	16.96

Table 25: Evaluation time for nomic-embed-text-v1.5, averaged per category. Table 2 serves as a legend.

q	m	Average Relative classification	Average Relative clustering	Average Relative sts	Average Relative pairclass	Average Relative retrieval	Average Relative rerank	Average Relative summ	Average Relative all
f32	768	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	512	0.939	0.910	0.986	0.977	1.056	0.991	1.001	0.957
f32	256	0.854	0.823	0.989	0.956	1.057	0.983	1.004	0.912
f32	128	0.771	0.779	0.991	0.949	1.057	0.981	1.007	0.881
f32	64	0.758	0.722	0.990	0.949	1.056	0.973	1.001	0.862
f32	32	0.737	0.729	0.986	0.945	1.057	0.972	1.003	0.857
f32	16	0.733	0.723	0.994	0.951	1.057	0.964	1.000	0.857
f32	8	0.731	0.693	0.995	0.941	1.056	0.961	0.998	0.847
int	768	1.008	1.089	0.993	1.024	1.058	0.996	0.998	1.027
int	512	0.945	0.931	0.993	1.009	1.057	0.987	1.003	0.968
int	256	0.842	0.834	0.991	0.962	1.056	0.986	1.002	0.914
int	128	0.766	0.811	0.998	0.954	1.056	0.977	1.004	0.890
int	64	0.750	0.752	0.986	0.956	1.058	0.974	0.999	0.867
int	32	0.750	0.722	0.990	0.948	1.058	0.969	1.004	0.860
int	16	0.741	0.725	0.990	0.970	1.057	0.959	1.005	0.859
int	8	0.733	0.680	0.987	0.907	1.058	0.957	0.998	0.839
bin	768	0.757	0.754	0.984	0.960	1.057	0.974	1.000	0.869
bin	512	0.747	0.750	0.989	0.941	1.058	0.979	0.997	0.866
bin	256	0.738	0.731	0.986	0.974	1.058	0.968	1.002	0.860
bin	128	0.726	0.709	0.988	0.964	1.058	0.966	1.001	0.851
bin	64	0.719	0.700	0.988	0.943	1.058	0.960	1.009	0.845
bin	32	0.710	0.675	0.988	0.968	1.058	0.957	0.998	0.837
bin	16	0.715	0.671	0.987	0.966	1.058	0.952	1.007	0.837
bin	8	0.702	0.662	0.979	0.948	1.052	0.947	0.965	0.826

Table 26: Relative evaluation time for nomic-embed-text-v1.5, averaged per category. Table 2 serves as a legend.

q	m	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	768	24576	0.848	0.488	0.577	0.734	0.720	0.763	0.928	0.735	0.585	0.531	0.413	0.329	NaN	0.327	0.484	0.492	0.394	0.439	0.862	0.785	0.745	0.856	0.804	0.872	0.836	0.843	0.881	NaN	0.956	0.736	0.860	0.536	NaN	NaN	0.617	0.310	0.504	0.325
f32	512	16384	0.848	0.485	0.573	0.721	0.714	0.758	0.923	0.725	0.588	0.520	0.399	0.328	NaN	0.319	0.484	0.498	0.394	0.437	0.856	0.785	0.743	0.854	0.803	0.871	0.834	0.840	0.878	NaN	0.956	0.736	0.859	0.529	NaN	NaN	0.614	0.309	0.504	0.324
f32	256	8192	0.843	0.458	0.554	0.710	0.698	0.741	0.908	0.699	0.573	0.518	0.396	0.314	NaN	0.321	0.465	0.482	0.392	0.445	0.847	0.785	0.745	0.856	0.802	0.868	0.830	0.837	0.870	NaN	0.953	0.734	0.856	0.514	NaN	NaN	0.614	0.306	0.500	0.319
f32	128	4096	0.835	0.428	0.538	0.670	0.664	0.715	0.881	0.657	0.570	0.522	0.383	0.298	NaN	0.319	0.455	0.482	0.387	0.424	0.835	0.783	0.740	0.849	0.795	0.860	0.825	0.832	0.867	NaN	0.947	0.727	0.852	0.498	NaN	NaN	0.609	0.303	0.491	0.298
f32	64	2048	0.810	0.394	0.504	0.636	0.611	0.666	0.835	0.578	0.568	0.518	0.364	0.273	NaN	0.308	0.412	0.460	0.386	0.387	0.799	0.781	0.737	0.831	0.783	0.847	0.818	0.824	0.863	NaN	0.938	0.713	0.843	0.445	NaN	NaN	0.612	0.297	0.479	0.289
f32	32	1024	0.736	0.339	0.474	0.599	0.494	0.547	0.712	0.436	0.543	0.513	0.308	0.219	NaN	0.282	0.278	0.364	0.380	0.272	0.762	0.767	0.717	0.787	0.748	0.811	0.785	0.793	0.845	NaN	0.904	0.682	0.823	0.356	NaN	NaN	0.584	0.278	0.442	0.284
f32	16	512	0.586	0.277	0.456	0.584	0.354	0.406	0.578	0.314	0.532	0.484	0.238	0.164	NaN	0.255	0.175	0.284	0.374	0.203	0.653	0.737	0.708	0.760	0.711	0.760	0.741	0.762	0.801	NaN	0.777	0.628	0.783	0.186	NaN	NaN	0.541	0.251	0.391	0.278
f32	8	256	0.385	0.274	0.436	0.509	0.191	0.260	0.395	0.153	0.505	0.481	0.190	0.138	NaN	0.234	0.127	0.243	0.366	0.161	0.507	0.680	0.677	0.679	0.651	0.699	0.683	0.702	0.705	NaN	0.552	0.543	0.716	0.047	NaN	NaN	0.501	0.244	0.329	0.286
int	768	6144	0.838	0.463	0.563	0.640	0.691	0.731	0.923	0.674	0.582	0.526	0.403	0.335	NaN	0.328	0.482	0.494	0.392	0.439	0.863	0.782	0.746	0.856	0.801	0.871	0.828	0.843	0.893	NaN	0.960	0.735	0.861	0.551	NaN	NaN	0.613	0.307	0.504	0.318
int	512	4096	0.836	0.455	0.557	0.616	0.683	0.725	0.916	0.657	0.575	0.519	0.397	0.321	NaN	0.321	0.483	0.488	0.395	0.436	0.852	0.781	0.747	0.855	0.801	0.871	0.826	0.841	0.891	NaN	0.959	0.735	0.860	0.550	NaN	NaN	0.612	0.305	0.501	0.314
int	256	2048	0.829	0.428	0.539	0.596	0.668	0.709	0.899	0.623	0.563	0.508	0.394	0.312	NaN	0.316	0.452	0.478	0.394	0.423	0.838	0.780	0.743	0.855	0.795	0.865	0.822	0.837	0.881	NaN	0.956	0.733	0.857	0.539	NaN	NaN	0.606	0.304	0.495	0.311
int	128	1024	0.810	0.391	0.517	0.564	0.625	0.675	0.861	0.567	0.560	0.518	0.380	0.295	NaN	0.315	0.448	0.473	0.392	0.416	0.818	0.777	0.733	0.853	0.786	0.858	0.815	0.830	0.879	NaN	0.952	0.726	0.852	0.509	NaN	NaN	0.607	0.304	0.483	0.307
int	64	512	0.763	0.347	0.480	0.574	0.552	0.621	0.795	0.458	0.561	0.527	0.358	0.269	NaN	0.307	0.414	0.455	0.385	0.388	0.782	0.773	0.726	0.828	0.769	0.846	0.801	0.817	0.869	NaN	0.944	0.714	0.843	0.463	NaN	NaN	0.600	0.300	0.465	0.296
int	32	256	0.634	0.272	0.442	0.499	0.412	0.496	0.640	0.317	0.550	0.516	0.304	0.215	NaN	0.281	0.275	0.362	0.379	0.270	0.776	0.753	0.703	0.785	0.723	0.806	0.778	0.781	0.817	NaN	0.910	0.681	0.820	0.348	NaN	NaN	0.569	0.277	0.432	0.301
int	16	128	0.397	0.239	0.443	0.496	0.222	0.302	0.481	0.190	0.519	0.485	0.227	0.154	NaN	0.256	0.175	0.284	0.375	0.202	0.670	0.706	0.666	0.719	0.677	0.740	0.731	0.719	0.735	NaN	0.798	0.627	0.782	0.197	NaN	NaN	0.533	0.264	0.383	0.261
int	8	64	0.290	0.256	0.426	0.503	0.153	0.233	0.302	0.121	0.500	0.479	0.182	0.133	NaN	0.237	0.129	0.242	0.368	0.167	0.533	0.638	0.590	0.632	0.603	0.674	0.632	0.600	0.598	NaN	0.549	0.540	0.714	0.044	NaN	NaN	0.497	0.244	0.319	0.276
bin	768	768	0.673	0.285	0.423	0.620	0.438	0.477	0.657	0.397	0.514	0.485	0.268	0.172	NaN	0.248	0.180	0.251	0.341	0.189	0.721	0.716	0.670	0.773	0.724	0.802	0.768	0.766	0.797	NaN	0.907	0.666	0.823	0.254	NaN	NaN	0.545	0.258	0.408	0.297
bin	512	512	0.576	0.255	0.408	0.564	0.361	0.411	0.548	0.310	0.500	0.478	0.225	0.145	NaN	0.231	0.145	0.225	0.344	0.164	0.727	0.690	0.638	0.734	0.690	0.775	0.745	0.737	0.757	NaN	0.878	0.636	0.805	0.177	NaN	NaN	0.521	0.248	0.385	0.290
bin	256	256	0.386	0.213	0.378	0.512	0.224	0.291	0.383	0.170	0.493	0.493	0.162	0.107	NaN	0.222	0.106	0.205	0.350	0.138	0.612	0.616	0.529	0.635	0.603	0.691	0.665	0.631	0.683	NaN	0.751	0.561	0.754	0.080	NaN	NaN	0.495	0.241	0.329	0.286
bin	128	128	0.250	0.210	0.375	0.546	0.158	0.202	0.286	0.135	0.489	0.501	0.128	0.094	NaN	0.219	0.095	0.199	0.356	0.137	0.475	0.529	0.432	0.521	0.501	0.600	0.570	0.520	0.541	NaN	0.519	0.488	0.687	0.040	NaN	NaN	0.472	0.238	0.277	0.273
bin	64	64	0.184	0.204	0.365	0.518	0.124	0.189	0.277	0.089	0.500	0.480	0.116	0.087	NaN	0.215	0.101	0.198	0.356	0.131	0.551	0.453	0.376	0.449	0.406	0.497	0.491	0.453	0.485	NaN	0.349	0.435	0.594	0.030	NaN	NaN	0.438	0.244	0.243	0.281
bin	32	32	0.107	0.186	0.347	0.546	0.068	0.133	0.235	0.044	0.478	0.488	0.109	0.082	NaN	0.200	0.095	0.193	0.354	0.120	0.409	0.385	0.374	0.326	0.315	0.392	0.403	0.405	0.346	NaN	0.198	0.361	0.507	0.027	NaN	NaN	0.428	0.241	0.220	0.274
bin	16	16	0.054	0.183	0.345	0.535	0.040	0.088	0.152	0.021	0.486	0.463	0.113	0.089	NaN	0.215	0.105	0.203	0.354	0.121	0.246	0.344	0.391	0.288	0.287	0.309	0.371	0.337	0.325	NaN	0.128	0.334	0.447	0.007	NaN	NaN	0.421	0.230	0.193	0.253
bin	8	8	0.025	0.170	0.342	0.532	0.028	0.062	0.129	0.013	0.434	0.472	0.135	0.100	NaN	0.211	0.101	0.201	0.221	0.118	0.218	0.227	0.229	0.135	0.170	0.153	0.170	0.188	0.178	NaN	0.058	0.295	0.381	0.000	NaN	NaN	0.296	0.102	0.055	0.315

Table 27: Absolute performance for nomic-embed-text-v1.5 on all tasks. Table 2 serves as a legend.

q	m	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	768	24576	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	nan	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	nan	1.000	1.000	1.000	1.000	nan	nan	1.000	1.000	1.000	1.000
f32	512	16384	1.000	0.994	0.994	0.982	0.992	0.994	0.995	0.985	1.006	0.979	0.966	0.995	nan	0.974	0.998	1.010	1.000	0.995	0.993	1.000	0.997	0.998	0.999	0.998	0.997	0.996	0.997	nan	1.000	1.000	0.999	0.987	nan	nan	0.995	0.997	1.000	0.996
f32	256	8192	0.995	0.938	0.961	0.967	0.970	0.971	0.979	0.950	0.979	0.976	0.956	0.953	nan	0.982	0.960	0.979	0.996	1.013	0.982	1.000	1.000	0.999	0.997	0.995	0.993	0.994	0.988	nan	0.997	0.997	0.996	0.958	nan	nan	0.995	0.987	0.993	0.980
f32	128	4096	0.985	0.877	0.932	0.912	0.922	0.937	0.949	0.894	0.974	0.984	0.926	0.904	nan	0.975	0.940	0.979	0.984	0.965	0.968	0.997	0.993	0.991	0.988	0.986	0.987	0.987	0.984	nan	0.991	0.988	0.990	0.930	nan	nan	0.987	0.975	0.976	0.918
f32	64	2048	0.955	0.807	0.874	0.866	0.849	0.872	0.900	0.786	0.970	0.976	0.880	0.828	nan	0.941	0.850	0.934	0.979	0.881	0.927	0.994	0.989	0.970	0.974	0.971	0.978	0.978	0.980	nan	0.981	0.969	0.980	0.831	nan	nan	0.992	0.957	0.950	0.889
f32	32	1024	0.868	0.696	0.822	0.816	0.687	0.717	0.767	0.593	0.928	0.967	0.744	0.664	nan	0.861	0.573	0.739	0.964	0.618	0.883	0.978	0.963	0.919	0.930	0.931	0.939	0.942	0.960	nan	0.946	0.927	0.957	0.664	nan	nan	0.947	0.895	0.878	0.875
f32	16	512	0.692	0.569	0.790	0.795	0.492	0.532	0.623	0.427	0.910	0.913	0.576	0.499	nan	0.779	0.361	0.577	0.950	0.462	0.757	0.939	0.951	0.887	0.884	0.872	0.887	0.905	0.909	nan	0.813	0.854	0.911	0.348	nan	nan	0.877	0.807	0.777	0.857
f32	8	256	0.454	0.561	0.756	0.693	0.265	0.340	0.426	0.208	0.863	0.906	0.459	0.419	nan	0.716	0.263	0.493	0.931	0.366	0.588	0.866	0.908	0.793	0.810	0.802	0.817	0.834	0.801	nan	0.577	0.738	0.832	0.088	nan	nan	0.812	0.785	0.652	0.880
int	768	6144	0.989	0.948	0.976	0.872	0.960	0.957	0.995	0.917	0.994	0.991	0.976	1.018	nan	1.002	0.996	1.004	0.996	0.999	1.000	0.996	1.001	1.000	0.995	0.999	0.991	1.000	1.014	nan	1.004	0.999	1.001	1.027	nan	nan	0.993	0.989	1.002	0.980
int	512	4096	0.986	0.933	0.965	0.840	0.950	0.950	0.988	0.893	0.983	0.977	0.961	0.973	nan	0.982	0.996	0.992	1.002	0.993	0.989	0.995	1.002	0.999	0.996	0.999	0.988	0.998	1.012	nan	1.004	0.999	1.000	1.027	nan	nan	0.991	0.984	0.995	0.966
int	256	2048	0.978	0.877	0.934	0.811	0.928	0.929	0.969	0.847	0.962	0.957	0.954	0.948	nan	0.965	0.933	0.970	1.001	0.963	0.971	0.994	0.997	0.998	0.989	0.991	0.983	0.994	1.000	nan	1.000	0.995	0.997	1.005	nan	nan	0.982	0.980	0.982	0.957
int	128	1024	0.956	0.802	0.897	0.768	0.869	0.885	0.928	0.771	0.957	0.976	0.919	0.896	nan	0.963	0.924	0.961	0.996	0.947	0.948	0.990	0.984	0.996	0.977	0.983	0.975	0.985	0.998	nan	0.996	0.986	0.991	0.951	nan	nan	0.984	0.978	0.959	0.944
int	64	512	0.901	0.711	0.832	0.781	0.767	0.813	0.857	0.623	0.959	0.993	0.867	0.818	nan	0.939	0.855	0.924	0.978	0.883	0.907	0.985	0.974	0.967	0.956	0.971	0.958	0.970	0.987	nan	0.988	0.970	0.980	0.864	nan	nan	0.972	0.965	0.922	0.910
int	32	256	0.748	0.557	0.767	0.680	0.573	0.650	0.690	0.431	0.941	0.973	0.735	0.652	nan	0.861	0.568	0.735	0.963	0.615	0.900	0.959	0.944	0.917	0.898	0.925	0.931	0.927	0.927	nan	0.952	0.926	0.954	0.650	nan	nan	0.921	0.892	0.859	0.927
int	16	128	0.468	0.490	0.768	0.675	0.308	0.396	0.518	0.259	0.888	0.914	0.549	0.468	nan	0.784	0.360	0.577	0.952	0.459	0.777	0.900	0.894	0.840	0.842	0.848	0.874	0.853	0.835	nan	0.835	0.853	0.910	0.367	nan	nan	0.863	0.849	0.759	0.803
int	8	64	0.343	0.525	0.738	0.686	0.212	0.305	0.325	0.165	0.855	0.902	0.440	0.404	nan	0.723	0.267	0.491	0.934	0.379	0.618	0.813	0.792	0.738	0.750	0.772	0.756	0.713	0.679	nan	0.575	0.734	0.830	0.081	nan	nan	0.805	0.786	0.633	0.850
bin	768	768	0.794	0.585	0.733	0.845	0.609	0.625	0.709	0.540	0.879	0.914	0.647	0.524	nan	0.758	0.371	0.510	0.865	0.429	0.836	0.912	0.900	0.902	0.900	0.920	0.919	0.910	0.905	nan	0.949	0.905	0.957	0.474	nan	nan	0.884	0.832	0.811	0.913
bin	512	512	0.679	0.522	0.707	0.768	0.501	0.538	0.591	0.422	0.855	0.901	0.544	0.442	nan	0.706	0.299	0.457	0.875	0.374	0.843	0.879	0.857	0.857	0.858	0.888	0.891	0.875	0.859	nan	0.919	0.865	0.936	0.330	nan	nan	0.844	0.800	0.764	0.892
bin	256	256	0.455	0.437	0.654	0.698	0.311	0.382	0.413	0.231	0.843	0.930	0.392	0.327	nan	0.679	0.218	0.416	0.890	0.314	0.710	0.784	0.711	0.742	0.750	0.792	0.795	0.749	0.776	nan	0.786	0.762	0.876	0.148	nan	nan	0.802	0.776	0.654	0.882
bin	128	128	0.295	0.429	0.649	0.744	0.220	0.265	0.308	0.184	0.836	0.945	0.309	0.285	nan	0.669	0.197	0.404	0.904	0.312	0.551	0.674	0.580	0.609	0.623	0.688	0.682	0.618	0.615	nan	0.543	0.664	0.799	0.075	nan	nan	0.765	0.768	0.550	0.839
bin	64	64	0.217	0.419	0.632	0.706	0.172	0.248	0.298	0.121	0.855	0.905	0.279	0.265	nan	0.657	0.207	0.401	0.904	0.299	0.639	0.577	0.505	0.524	0.504	0.570	0.587	0.538	0.550	nan	0.365	0.592	0.691	0.056	nan	nan	0.710	0.785	0.483	0.864
bin	32	32	0.127	0.382	0.601	0.744	0.095	0.175	0.253	0.060	0.817	0.919	0.263	0.249	nan	0.612	0.196	0.392	0.900	0.273	0.474	0.491	0.502	0.381	0.392	0.450	0.482	0.480	0.393	nan	0.207	0.490	0.589	0.050	nan	nan	0.694	0.777	0.436	0.843
bin	16	16	0.064	0.375	0.597	0.729	0.055	0.115	0.164	0.029	0.831	0.871	0.273	0.271	nan	0.656	0.216	0.411	0.899	0.275	0.285	0.438	0.525	0.336	0.357	0.354	0.443	0.400	0.369	nan	0.134	0.453	0.520	0.013	nan	nan	0.682	0.741	0.382	0.779
bin	8	8	0.030	0.349	0.593	0.725	0.038	0.081	0.139	0.018	0.741	0.888	0.326	0.302	nan	0.644	0.209	0.407	0.561	0.268	0.253	0.289	0.307	0.157	0.212	0.175	0.203	0.223	0.202	nan	0.061	0.401	0.443	0.000	nan	nan	0.480	0.328	0.110	0.969

Table 28: Relative performance for nomic-embed-text-v1.5 on all tasks. Table 2 serves as a legend.

q	m	req. bits	Average classification	Average clustering	Average sts	Average pairclass	Average retrieval	Average rerank	Average summ	Average all
f32	768	24576	0.724	0.444	0.832	0.851	0.536	0.477	0.325	0.650
f32	512	16384	0.718	0.441	0.829	0.850	0.529	0.476	0.324	0.647
f32	256	8192	0.701	0.434	0.827	0.848	0.514	0.474	0.319	0.640
f32	128	4096	0.673	0.427	0.821	0.842	0.498	0.468	0.298	0.628
f32	64	2048	0.629	0.408	0.809	0.831	0.445	0.463	0.289	0.606
f32	32	1024	0.542	0.351	0.780	0.803	0.356	0.435	0.284	0.555
f32	16	512	0.444	0.301	0.737	0.729	0.186	0.394	0.278	0.492
f32	8	256	0.325	0.272	0.665	0.603	0.047	0.358	0.286	0.419
int	768	6144	0.690	0.442	0.831	0.852	0.551	0.475	0.318	0.642
int	512	4096	0.681	0.437	0.829	0.851	0.550	0.473	0.314	0.638
int	256	2048	0.661	0.427	0.824	0.849	0.539	0.468	0.311	0.628
int	128	1024	0.626	0.422	0.816	0.843	0.509	0.465	0.307	0.615
int	64	512	0.574	0.407	0.801	0.833	0.463	0.455	0.296	0.591
int	32	256	0.464	0.350	0.769	0.804	0.348	0.426	0.301	0.533
int	16	128	0.346	0.297	0.707	0.736	0.197	0.393	0.261	0.460
int	8	64	0.286	0.271	0.611	0.601	0.044	0.353	0.276	0.394
bin	768	768	0.496	0.294	0.749	0.799	0.254	0.404	0.297	0.515
bin	512	512	0.429	0.273	0.722	0.773	0.177	0.385	0.290	0.480
bin	256	256	0.320	0.253	0.629	0.689	0.080	0.355	0.286	0.412
bin	128	128	0.270	0.246	0.521	0.565	0.040	0.329	0.273	0.355
bin	64	64	0.244	0.243	0.462	0.460	0.030	0.308	0.281	0.321
bin	32	32	0.208	0.235	0.373	0.355	0.027	0.297	0.274	0.276
bin	16	16	0.177	0.239	0.322	0.303	0.007	0.281	0.253	0.249
bin	8	8	0.163	0.221	0.185	0.245	0.000	0.151	0.315	0.190

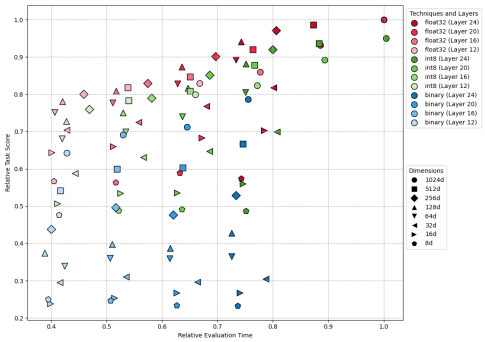
Table 29: Performance for nomic-embed-text-v1.5, averaged per category. Table 2 serves as a legend.

q	m	req. bits	Average Relative classification	Average Relative clustering	Average Relative sts	Average Relative pairclass	Average Relative retrieval	Average Relative rerank	Average Relative summ	Average Relative all
f32	768	24576	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	512	16384	0.992	0.991	0.997	1.000	0.987	0.998	0.996	0.994
f32	256	8192	0.966	0.977	0.994	0.997	0.958	0.992	0.980	0.982
f32	128	4096	0.926	0.959	0.987	0.990	0.930	0.979	0.918	0.961
f32	64	2048	0.864	0.915	0.974	0.977	0.831	0.966	0.889	0.925
f32	32	1024	0.746	0.784	0.938	0.943	0.664	0.907	0.875	0.840
f32	16	512	0.615	0.670	0.888	0.859	0.348	0.820	0.857	0.741
f32	8	256	0.463	0.602	0.802	0.716	0.088	0.750	0.880	0.638
int	768	6144	0.952	0.997	1.000	1.001	1.027	0.995	0.980	0.988
int	512	4096	0.938	0.984	0.997	1.001	1.027	0.990	0.966	0.980
int	256	2048	0.909	0.961	0.991	0.998	1.005	0.981	0.957	0.963
int	128	1024	0.860	0.949	0.982	0.991	0.951	0.974	0.944	0.942
int	64	512	0.786	0.913	0.964	0.979	0.864	0.953	0.910	0.904
int	32	256	0.637	0.782	0.925	0.944	0.650	0.891	0.927	0.810
int	16	128	0.485	0.661	0.851	0.866	0.367	0.824	0.803	0.698
int	8	64	0.412	0.600	0.737	0.713	0.081	0.742	0.850	0.606
bin	768	768	0.680	0.655	0.900	0.937	0.474	0.842	0.913	0.770
bin	512	512	0.591	0.606	0.868	0.907	0.330	0.803	0.892	0.716
bin	256	256	0.448	0.556	0.756	0.808	0.148	0.744	0.882	0.620
bin	128	128	0.387	0.540	0.626	0.669	0.075	0.694	0.839	0.547
bin	64	64	0.352	0.530	0.555	0.549	0.056	0.659	0.864	0.504
bin	32	32	0.304	0.513	0.450	0.429	0.050	0.636	0.843	0.447
bin	16	16	0.266	0.523	0.390	0.369	0.013	0.602	0.779	0.413
bin	8	8	0.247	0.483	0.225	0.301	0.000	0.306	0.969	0.327

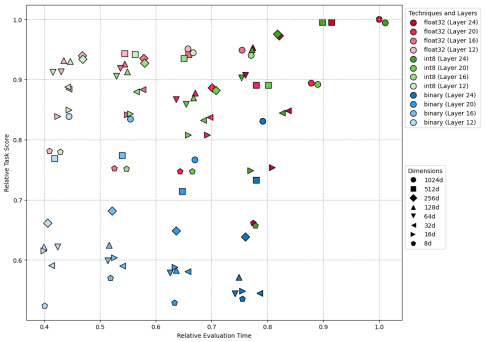
Table 30: Relative performance for nomic-embed-text-v1.5, averaged per category. Table 2 serves as a legend.

F. mxbai-embed-2d-large-v1

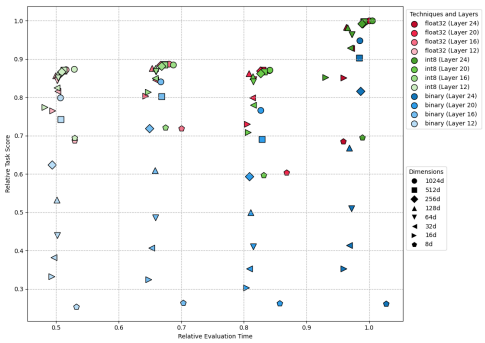
This section contains additional figures and tables for the MBAI-2D model.



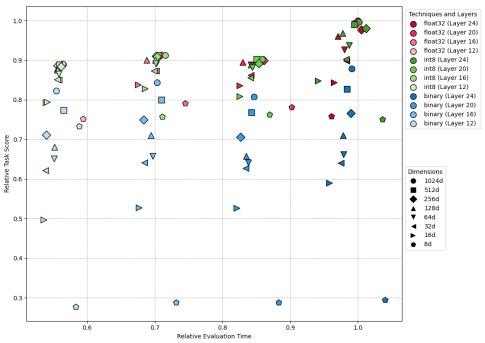
(a) Classification



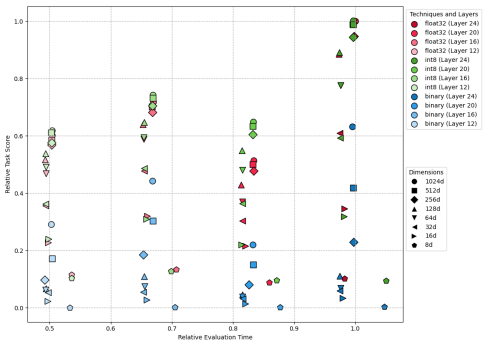
(b) Clustering



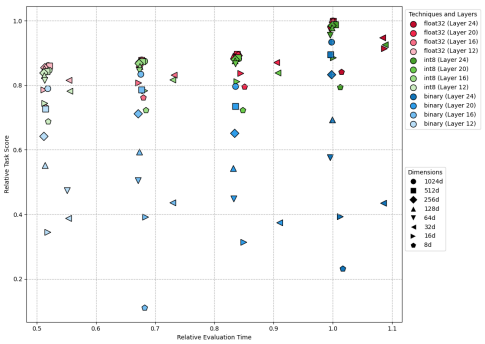
(c) Pair Classification



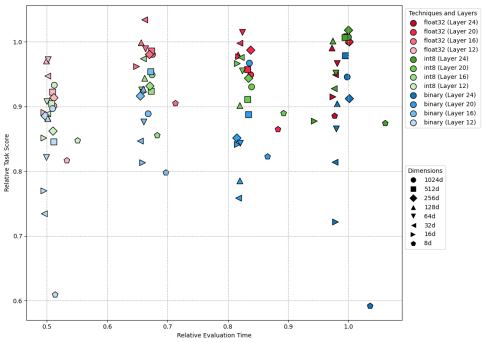
(d) Re-Ranking



(e) Retrieval



(f) Semantic Textual Similarity



(g) Summarization

Figure 16: Trade-off between memory consumption, evaluation time, and performance of MBAI-2D, averaged over all selected MTEB tasks for each task category. The x-axis represents the relative evaluation time while the y-axis represents the relative performance. The color of the markers represent the quantization technique and inference layer, while the shape represents the embedding size of the respective MR-quantization combination.

q	m	l	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	1024	24	68.22	18.40	27.82	11.08	45.70	26.74	18.19	76.19	14.78	8.95	165.95	18.11	132.78	16.55	239.84	230.99	190.74	12.34	2.39	57.73	25.31	10.26	27.41	21.87	9.40	10.23	1.50	31.36	41.51	58.56	257.29	457.45	718.69	288.03	32.70	293.66	311.40	39.45
f32	1024	20	60.65	15.93	23.57	9.43	42.60	24.74	15.10	71.85	13.74	9.22	140.15	15.76	112.17	14.68	201.77	195.59	162.66	10.28	1.97	48.57	21.19	8.46	23.02	18.61	7.96	8.52	1.30	26.07	35.08	49.11	215.32	380.03	600.52	240.81	27.52	261.31	259.91	33.03
f32	1024	16	54.53	13.07	20.12	7.66	39.16	22.78	13.28	65.19	13.22	7.74	112.46	15.33	89.76	12.86	164.83	160.36	133.54	8.90	1.61	39.21	17.11	6.82	18.15	15.00	6.52	6.96	1.03	20.91	29.05	39.12	172.93	305.25	482.23	192.98	22.20	230.51	207.42	26.63
f32	1024	12	48.84	10.35	15.92	6.07	35.51	20.51	10.48	62.86	10.29	9.19	87.63	12.43	71.57	12.78	126.21	125.11	108.46	8.48	1.23	30.21	13.13	5.11	13.78	11.27	4.89	5.38	0.84	16.06	22.59	29.13	130.37	230.40	362.23	145.63	16.76	201.43	154.52	20.18
f32	512	24	57.17	17.95	27.63	11.24	39.01	15.12	17.37	60.07	12.06	7.16	162.60	16.28	127.72	16.04	236.20	225.60	179.20	10.08	2.37	57.69	25.34	10.17	27.38	21.92	9.46	10.24	1.53	31.14	40.31	58.89	256.36	453.93	717.38	287.47	32.73	281.63	320.76	39.33
f32	512	20	52.09	15.44	23.11	9.31	35.97	13.73	14.72	55.37	10.20	5.45	137.60	15.12	107.30	14.25	199.69	190.28	151.56	8.79	1.96	48.84	21.12	8.40	23.03	18.63	7.94	8.52	1.30	26.07	33.98	49.24	213.63	379.63	599.39	239.98	27.51	250.54	267.41	32.83
f32	512	16	46.21	12.92	18.93	7.59	32.26	11.41	12.10	50.17	9.13	5.66	110.26	12.71	88.04	11.56	160.17	154.85	124.74	7.66	1.59	39.26	17.18	6.86	18.14	14.91	6.47	6.97	1.04	21.02	27.80	39.94	171.96	304.23	480.16	193.01	22.20	217.90	215.73	26.56
f32	512	12	39.21	10.06	14.95	6.25	28.80	8.90	9.47	45.63	8.56	5.84	85.64	10.36	67.70	9.44	121.48	117.71	99.50	6.26	1.23	29.81	13.25	5.21	13.75	11.22	4.98	5.24	0.87	16.06	21.42	30.25	129.62	229.59	361.41	145.41	16.72	188.46	163.22	20.09
f32	256	24	52.03	17.66	27.37	11.15	26.14	14.81	16.54	53.17	7.97	3.31	162.01	16.39	127.69	12.98	234.61	220.48	173.64	10.19	2.38	57.93	25.37	10.18	27.29	21.94	9.43	10.24	1.53	30.96	40.05	58.91	256.66	454.51	719.23	287.68	32.87	287.84	320.54	39.50
f32	256	20	46.17	15.32	23.07	9.39	22.87	13.02	14.12	47.54	7.26	3.64	135.48	14.23	105.54	11.09	197.09	186.45	145.09	8.04	2.01	48.40	21.18	8.37	22.89	18.60	7.90	8.57	1.31	25.89	33.53	49.29	213.94	380.36	600.07	240.71	27.57	257.94	268.37	33.05
f32	256	16	40.13	12.14	18.93	7.52	19.06	10.25	11.57	41.86	6.38	3.37	109.54	10.94	86.06	9.39	158.57	151.80	118.68	6.84	1.59	38.88	17.13	6.76	18.12	14.88	6.54	6.95	1.03	20.86	27.32	39.64	171.55	304.48	482.12	192.53	22.17	216.13	215.56	26.41
f32	256	12	33.97	9.59	14.56	5.96	14.99	8.07	8.87	35.89	5.53	3.21	83.79	8.86	67.57	8.31	119.69	113.30	92.62	5.84	1.23	29.52	13.07	5.11	13.76	11.19	4.81	5.30	0.81	15.87	20.93	30.28	129.30	229.80	361.83	145.19	16.75	185.57	163.26	20.16
f32	128	24	41.41	17.10	26.66	10.98	24.45	14.22	16.02	38.84	6.90	2.25	161.36	15.03	126.30	12.54	233.63	218.62	169.01	8.65	2.35	57.17	25.35	10.10	27.39	21.79	9.39	10.24	1.55	31.02	39.11	57.28	249.54	442.76	700.96	280.96	32.01	271.98	313.10	38.35
f32	128	20	35.52	14.87	22.59	9.18	20.92	12.28	13.93	32.87	7.19	2.28	135.41	12.57	105.69	11.17	196.55	185.02	142.55	7.63	1.95	47.78	21.36	8.44	22.74	18.34	7.92	8.48	1.26	26.07	32.92	48.00	208.69	370.34	584.89	234.93	26.85	243.15	261.85	32.22
f32	128	16	29.23	11.94	18.60	7.41	17.23	9.83	11.15	27.68	5.87	2.27	108.90	10.31	84.94	8.48	157.31	148.60	114.90	6.30	1.56	38.36	17.23	6.74	18.19	14.81	6.48	6.88	1.05	20.89	26.77	38.61	168.73	297.33	468.89	188.60	21.64	212.96	210.51	25.90
f32	128	12	23.95	9.87	14.63	5.89	14.18	8.02	9.22	22.86	4.56	2.71	83.18	8.89	65.86	6.71	119.38	112.77	88.24	4.86	1.21	28.95	13.05	5.22	13.76	11.24	4.82	5.19	0.82	16.05	20.77	29.50	126.91	225.19	353.26	142.17	16.41	191.69	159.75	19.68
f32	64	24	40.93	17.04	26.49	10.92	23.53	14.04	15.89	37.24	6.85	1.83	161.08	15.24	125.24	11.60	232.94	218.50	167.95	8.43	2.32	57.17	25.35	10.15	27.23	21.88	9.44	10.18	1.53	31.15	39.76	57.50	251.31	445.29	702.21	281.87	32.10	282.06	310.61	38.71
f32	64	20	34.95	14.64	22.86	9.27	19.98	12.28	13.53	31.84	5.68	1.88	135.06	12.16	104.09	9.61	194.91	181.81	140.59	7.16	1.99	47.67	21.29	8.40	22.74	18.52	7.97	8.49	1.29	25.99	33.60	48.16	209.60	371.82	586.91	235.53	26.98	251.81	262.39	32.51
f32	64	16	28.59	12.04	18.44	7.38	16.55	9.66	11.10	26.61	6.56	1.85	108.58	10.00	84.49	8.09	156.58	148.21	113.62	6.07	1.60	38.42	17.25	6.73	18.27	14.82	6.47	6.92	1.05	20.82	27.36	38.91	168.43	298.68	470.31	188.74	21.77	223.51	211.96	26.17
f32	64	12	23.09	9.20	14.79	5.67	13.39	8.26	8.83	21.12	4.19	2.96	83.23	8.00	64.54	6.41	119.29	112.58	86.85	5.37	1.26	28.94	13.04	5.13	13.79	11.20	4.82	5.31	0.81	15.99	20.99	29.59	127.14	225.87	354.79	142.60	16.43	192.54	160.26	19.81
f32	32	24	43.68	18.38	29.52	12.10	25.64	14.57	17.91	40.10	8.40	2.59	173.30	15.62	132.97	13.08	255.38	237.50	181.80	9.51	2.50	62.80	27.83	11.23	29.87	23.77	10.28	11.14	1.62	33.69	39.67	57.48	251.40	444.59	701.21	281.43	32.13	281.88	313.66	38.58
f32	32	20	38.04	15.77	25.04	10.19	21.55	12.67	14.76	34.46	6.64	1.67	146.30	13.80	111.19	10.40	213.72	198.52	153.78	8.19	2.09	52.48	23.12	9.36	25.06	19.95	8.58	9.27	1.35	28.19	33.34	48.09	209.86	371.33	587.13	235.65	26.94	252.02	262.83	32.34
f32	32	16	30.56	13.22	19.77	8.37	17.77	10.66	12.17	28.34	6.25	2.79	117.07	10.72	90.14	8.42	171.24	159.64	121.94	6.25	1.71	42.49	18.53	7.40	20.04	16.19	6.87	7.45	1.13	22.74	27.47	38.71	168.32	298.66	471.52	188.95	21.72	223.99	211.68	26.10
f32	32	12	23.51	10.19	15.13	6.28	13.66	8.05	9.43	22.59	4.29	1.81	89.53	9.04	68.68	7.50	129.25	121.60	92.67	4.80	1.29	31.97	14.03	5.60	15.10	12.29	5.28	5.63	0.88	16.96	21.12	29.63	126.70	225.65	354.55	142.52	16.45	192.69	160.35	19.77
f32	16	24	43.01	17.91	28.96	11.92	24.94	14.67	17.06	39.73	6.75	1.72	173.16	15.45	132.40	12.04	255.02	239.14	181.41	9.48	2.53	62.69	27.78	11.20	29.93	23.80	10.34	11.21	1.64	33.45	38.78	57.06	249.31	445.18	705.36	286.07	31.92	268.81	311.92	38.43
f32	16	20	36.56	15.50	23.98	10.06	21.35	12.90	14.84	35.16	5.93	2.31	144.94	12.87	112.00	10.74	213.32	198.38	151.24	7.93	2.02	47.58	21.82	8.65	22.93	18.43	7.98	8.62	1.27	26.69	32.64	47.85	208.80	372.42	588.58	238.36	26.86	239.85	261.07	32.25
f32	16	16	26.41	11.84	17.85	7.60	16.55	9.92	11.89	26.51	5.77	1.95	111.98	10.57	86.22	9.01	158.93	150.77	118.56	6.05	1.55	38.35	17.42	6.94	18.27	14.86	6.44	6.94	0.99	21.10	26.69	37.62	165.68	300.07	473.06	191.59	21.29	205.64	210.10	25.56
f32	16	12	22.53	9.41	14.13	5.70	12.68	7.63	9.07	21.09	4.17	1.84	83.59	8.02	65.84	6.70	120.76	114.43	89.74	5.37	1.18	29.94	13.17	5.28	13.80	11.18	4.80	5.22	0.78	16.22	20.52	29.18	126.00	225.22	357.08	145.13	16.47	180.34	155.51	19.51
f32	8	24	40.82	17.53	27.04	11.04	23.71	13.76	16.18	37.81	6.29	2.38	164.55	14.98	127.30	11.57	241.49	225.53	174.13	8.39	2.35	58.39	26.28	10.42	27.78	22.49	9.71	10.54	1.49	31.22	38.69	57.10	249.34	445.23	705.18	285.71	31.81	266.99	311.32	38.54
f32	8	20	36.04	15.08	22.47	9.26	20.15	11.90	13.77	32.19	5.38	1.53	137.91	12.29	106.02	9.77	201.06	190.65	144.97	7.16	1.97	48.98	22.08	8.62	23.28	18.71	8.17	8.77	1.28	26.36	35.22	51.91	223.78	372.87	631.20	254.69	29.45	264.62	281.84	34.82
f32	8	16	28.55	11.97	18.34	7.61	16.45	10.24	11.24	26.95	4.50	1.44	110.97	10.08	85.98	8.15	162.60	152.29	116.89	5.88	1.58	39.24	17.57	6.96	18.61	15.04	6.46	7.11	1.00	21.08	28.50	41.91	179.79	322.14	506.93	205.14	23.16	233.72	227.31	28.12
f32	8	12	22.64	9.58	14.71	5.69	12.96	8.25	8.53	21.18	3.70	1.46	84.29	7.72	65.81	6.36	121.96	115.24	89.39	4.82	1.19	29.87	13.36	5.19	13.95	11.34	4.88	5.32	0.78	15.99	21.77	31.32	136.17	245.84	383.19	154.80	17.96	200.77	171.16	21.02

Table 31: Absolute evaluation time for mxbai-embed-2d-large-v1 on all tasks using float quantization. Table 2 serves as a legend.

q	m	l	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
int	1024	24	70.41	17.84	27.48	11.21	47.66	26.47	17.43	79.14	14.57	8.40	167.29	18.38	130.46	18.16	240.57	232.89	189.46	13.22	2.38	58.17	25.46	10.27	27.51	21.98	9.49	10.32	1.52	31.01	41.93	58.82	257.54	453.30	718.28	288.13	32.92	291.97	312.55	39.43
int	1024	20	63.03	15.65	23.48	9.46	43.40	24.60	15.20	73.94	13.29	8.02	138.75	18.02	111.51	16.88	202.18	196.33	161.51	10.76	1.96	48.87	21.23	8.56	23.17	18.51	7.98	8.59	1.28	26.28	35.18	49.31	214.83	379.57	599.19	240.91	27.56	263.47	260.33	33.09
int	1024	16	56.93	13.04	19.04	7.81	38.72	21.77	12.75	66.77	12.62	8.64	113.48	14.44	92.04	13.77	165.25	160.87	133.28	9.80	1.61	39.38	17.26	6.83	18.35	15.03	6.56	6.97	1.07	20.92	29.12	40.09	173.64	304.78	480.59	193.26	22.28	232.70	209.67	26.59
int	1024	12	48.72	10.24	15.53	6.25	35.90	19.61	10.39	60.64	11.78	10.12	87.68	12.03	71.13	11.99	126.10	124.71	108.09	8.08	1.22	30.12	13.34	5.29	13.88	11.36	4.91	5.27	0.83	16.11	23.11	30.48	130.84	229.92	362.37	145.51	16.79	198.66	156.58	20.20
int	512	24	57.97	18.06	27.77	11.49	39.49	15.02	17.80	60.73	10.78	6.64	163.63	17.06	128.69	14.77	236.50	225.81	177.39	10.19	2.38	57.90	25.49	10.20	27.40	21.87	9.46	10.29	1.57	30.95	40.32	58.94	256.66	453.97	717.60	287.55	32.81	279.94	319.99	39.22
int	512	20	51.51	15.46	23.19	9.46	36.51	13.68	14.74	55.89	10.96	7.25	136.45	15.18	107.23	12.37	198.74	189.43	150.42	10.05	1.97	48.83	21.20	8.41	23.09	18.64	7.94	8.54	1.30	25.88	34.04	49.36	214.49	378.82	599.28	240.30	27.57	248.64	267.87	32.85
int	512	16	45.81	12.84	18.91	7.88	32.27	11.08	12.27	49.44	8.46	5.40	110.47	12.03	89.15	12.04	159.90	154.32	123.19	7.64	1.60	39.25	17.17	6.86	18.15	14.93	6.50	6.96	1.05	21.00	27.79	39.91	171.88	304.17	482.12	193.16	22.12	218.76	215.80	26.53
int	512	12	40.25	10.13	14.93	6.46	29.07	8.42	9.50	44.45	8.74	6.43	85.26	10.95	69.31	10.66	121.56	117.02	97.14	6.37	1.20	29.75	13.22	5.31	13.74	11.27	4.89	5.34	0.81	16.09	21.46	30.25	129.48	229.99	360.78	145.25	16.78	187.95	163.31	20.02
int	256	24	52.41	17.36	27.14	11.28	25.78	14.09	16.80	51.54	7.99	4.48	161.17	15.23	125.29	13.35	234.87	219.93	170.15	9.25	2.39	57.56	25.27	10.14	27.31	21.85	9.44	10.19	1.52	31.18	40.01	58.86	256.26	452.55	718.46	288.12	32.76	294.53	320.92	39.44
int	256	20	45.16	14.75	22.73	9.49	21.97	12.58	14.35	46.22	7.53	4.31	134.93	13.36	105.55	10.70	196.59	184.21	145.34	8.82	1.99	48.33	21.17	8.39	22.80	18.51	7.99	8.53	1.26	25.97	33.54	49.30	213.82	378.74	600.07	240.21	27.43	252.55	268.23	32.91
int	256	16	39.99	12.63	18.61	7.81	18.47	10.41	12.12	41.25	6.30	3.36	109.32	10.94	85.50	9.15	158.91	149.27	118.82	7.49	1.57	39.16	17.18	6.69	18.12	14.78	6.42	6.91	1.05	20.90	27.33	39.72	172.29	304.35	481.80	192.98	22.14	217.38	215.72	26.43
int	256	12	33.96	10.03	14.71	6.46	15.27	7.86	9.10	35.55	6.04	3.92	84.18	8.88	65.31	7.43	120.02	114.31	90.64	5.43	1.20	29.27	13.14	5.11	13.67	11.20	4.94	5.25	0.81	15.82	20.93	30.42	129.29	229.85	361.08	145.37	16.76	190.36	163.01	20.12
int	128	24	41.34	17.45	26.91	11.29	24.49	13.98	16.35	38.72	7.36	2.20	161.93	15.33	125.55	11.72	234.32	218.35	167.61	8.65	2.38	57.07	25.36	10.12	27.37	21.80	9.41	10.23	1.54	30.94	39.17	57.38	250.29	443.92	701.69	280.71	32.08	277.61	313.42	38.43
int	128	20	35.17	15.34	22.90	9.60	21.03	12.09	14.40	32.82	7.36	2.43	135.19	12.67	105.22	10.11	196.18	184.27	141.75	7.76	1.98	47.80	21.27	8.39	22.73	18.44	7.95	8.51	1.28	25.95	33.57	47.93	210.02	371.37	586.03	235.26	26.94	252.51	262.34	32.34
int	128	16	30.21	12.48	18.91	7.71	17.38	9.74	11.51	27.69	5.38	2.32	109.28	10.78	84.88	8.52	157.26	147.53	115.05	6.93	1.64	38.21	17.11	6.81	18.23	14.82	6.40	6.90	1.04	20.85	27.46	38.73	168.41	298.58	470.61	188.96	21.73	223.09	211.20	26.11
int	128	12	23.91	9.64	15.32	6.06	14.11	8.10	9.70	22.69	4.73	3.47	83.57	8.08	64.87	6.77	119.69	112.75	87.56	5.74	1.25	28.97	13.05	5.22	13.84	11.23	4.84	5.22	0.79	15.88	21.19	29.60	126.78	225.57	353.71	142.49	16.45	191.84	159.82	19.77
int	64	24	41.03	17.90	27.33	11.38	23.70	13.68	16.39	37.06	6.64	2.00	161.12	14.41	124.91	11.33	233.69	217.19	167.37	8.41	2.36	56.96	25.37	10.09	27.22	21.88	9.41	10.22	1.52	31.17	39.72	57.45	252.00	444.49	703.52	281.51	32.24	283.81	314.20	38.63
int	64	20	34.18	15.09	22.74	9.56	20.19	11.72	14.47	31.76	6.60	2.13	134.88	12.36	104.56	9.71	195.20	183.14	141.18	7.92	1.97	47.94	21.24	8.38	22.80	18.49	7.87	8.46	1.31	26.02	33.43	48.23	209.82	371.53	586.39	235.56	26.90	254.43	262.70	32.51
int	64	16	29.48	12.85	18.74	8.04	16.23	10.05	11.98	26.76	5.16	2.01	108.59	10.39	84.33	8.20	156.70	146.37	114.54	5.89	1.60	38.36	17.15	6.83	18.18	14.88	6.40	6.97	1.03	20.97	27.27	39.08	168.41	297.90	470.37	189.13	21.72	223.00	211.72	25.92
int	64	12	22.75	9.74	15.30	6.02	13.36	8.14	9.54	21.27	4.73	1.77	82.55	8.15	65.23	6.44	118.47	111.30	86.47	4.67	1.23	28.87	12.96	5.16	13.76	11.24	4.86	5.35	0.80	15.99	21.08	29.68	127.63	226.05	354.56	142.64	16.43	192.96	160.37	19.71
int	32	24	43.39	18.81	30.01	12.30	25.08	14.92	17.73	40.54	6.97	2.87	173.70	15.47	133.02	12.11	254.87	238.70	181.43	9.77	2.55	62.39	28.11	11.10	29.77	23.84	10.34	11.29	1.63	33.55	39.54	57.40	251.43	444.66	702.51	281.16	32.06	281.77	313.62	38.49
int	32	20	37.29	16.07	24.95	10.36	21.48	12.97	15.02	34.38	5.96	1.78	145.38	13.79	111.23	10.40	212.94	198.74	150.52	7.57	2.09	52.85	23.09	9.27	25.18	19.94	8.64	9.33	1.36	28.27	33.60	47.99	209.69	371.68	587.20	235.16	26.92	251.86	262.38	32.41
int	32	16	29.90	13.58	20.11	8.51	17.64	10.71	12.81	28.37	5.13	1.81	117.22	11.32	89.92	9.01	171.70	159.55	121.57	6.25	1.69	42.72	18.68	7.38	20.06	16.20	6.85	7.47	1.10	22.64	27.29	38.83	168.29	298.85	470.46	188.81	21.81	220.34	211.31	26.02
int	32	12	23.73	10.46	16.43	6.42	13.73	8.48	9.83	22.72	5.21	1.62	88.96	8.33	68.51	7.21	129.11	120.89	94.19	4.75	1.32	31.37	14.16	5.64	14.85	12.15	5.30	5.62	0.89	17.24	20.95	29.63	127.11	225.61	354.76	142.45	16.39	191.49	160.09	19.78
int	16	24	40.98	17.22	26.62	11.33	23.45	14.23	16.64	37.09	6.64	2.03	163.08	14.74	126.99	11.51	235.78	220.22	171.48	9.11	2.33	57.93	25.78	10.26	27.21	22.06	9.52	10.35	1.46	31.56	37.44	54.98	244.70	444.94	704.61	285.50	31.45	265.16	298.42	37.20
int	16	20	33.99	14.60	22.52	9.73	19.89	11.66	13.63	31.51	5.90	2.04	139.56	12.57	107.28	10.26	200.62	187.74	143.17	7.29	1.94	48.32	21.55	8.58	22.72	18.48	8.01	8.56	1.26	26.27	32.72	48.03	208.71	368.10	581.86	238.52	26.92	239.52	260.82	32.17
int	16	16	28.37	12.14	18.73	7.80	16.50	9.92	11.91	27.28	5.47	2.16	113.08	10.91	86.92	8.28	163.28	152.28	115.96	6.72	1.54	38.78	17.66	7.00	18.75	15.08	6.50	7.05	1.02	21.47	26.60	38.33	166.07	298.11	472.06	191.99	21.72	211.63	209.00	26.02
int	16	12	22.82	9.54	14.49	6.01	12.78	7.73	9.44	21.59	4.69	2.47	85.09	8.86	65.71	7.27	122.66	114.69	88.74	5.41	1.19	29.86	13.37	5.20	13.98	11.38	4.92	5.35	0.74	16.16	20.04	28.29	123.35	226.71	356.20	145.05	16.37	180.30	158.33	19.51
int	8	24	41.06	17.91	27.36	11.22	23.76	13.85	16.46	37.88	7.23	2.01	165.06	15.31	126.36	11.59	240.11	225.19	170.54	8.74	2.35	58.23	26.22	10.40	27.69	22.45	9.74	10.50	1.49	31.17	39.35	58.62	261.76	478.50	754.09	303.83	34.63	297.58	322.71	41.83
int	8	20	34.56	15.32	22.76	9.52	20.03	11.78	13.95	32.16	6.51	1.99	138.83	12.80	107.33	10.46	201.34	188.50	144.54	7.34	1.96	48.95	22.09	8.66	23.16	18.74	8.12	8.79	1.25	26.25	33.28	49.35	218.84	393.95	630.96	252.00	28.31	258.73	268.35	35.18
int	8	16	29.42	12.47	19.18	7.58	16.03	9.69	11.37	26.86	6.04	1.91	111.86	10.62	86.67	8.20	161.79	151.64	116.21	6.21	1.56	39.07	17.71	6.95	18.55	15.11	6.58	7.02	1.05	21.40	27.08	39.70	178.41	319.22	507.11	199.75	22.09	226.43	213.56	26.94
int	8	12	24.13	9.43	14.58	6.23	13.48	7.67	8.94	21.62	4.80	1.81	85.43	8.06	66.68	6.45	122.70	114.82	88.89	5.41	1.22	30.35	13.45	5.30	14.00	11.36	4.89	5.32	0.78	16.11	21.62	31.69	135.81	245.38	381.88	155.68	17.78	198.07	170.37	21.73

Table 32: Absolute evaluation time for mxbai-embed-2d-large-v1 on all tasks using int quantization. Table 2 serves as a legend.

q	m	l	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
bin	1024	24	42.32	17.36	26.99	11.15	24.80	14.15	16.49	39.63	7.98	3.41	161.67	14.93	124.98	11.96	234.13	219.25	168.71	8.93	2.35	57.32	25.39	10.22	27.29	21.98	9.41	10.20	1.52	31.10	39.67	58.87	255.71	453.09	717.00	287.55	32.81	276.44	320.05	39.34
bin	1024	20	37.17	15.09	22.56	9.28	21.41	12.06	14.27	33.89	6.41	2.56	134.87	13.17	104.98	10.41	197.61	183.26	143.57	8.01	1.98	47.95	21.34	8.42	22.82	18.62	7.97	8.51	1.26	25.92	33.44	49.23	214.58	378.97	599.38	240.34	27.37	247.62	267.62	32.95
bin	1024	16	29.70	12.21	18.98	7.52	17.80	10.09	11.48	28.57	5.56	2.63	109.26	10.87	84.87	9.00	158.26	148.10	114.94	6.51	1.59	38.54	17.25	6.86	18.22	14.88	6.52	6.90	1.04	20.93	27.22	39.72	171.80	304.47	480.95	192.82	22.13	217.53	215.57	26.34
bin	1024	12	24.60	9.53	14.98	5.86	14.53	8.44	9.66	23.73	4.68	2.37	83.36	9.13	65.68	6.94	120.03	114.43	88.12	6.02	1.23	29.26	13.13	5.26	13.84	11.30	4.89	5.34	0.84	15.89	20.84	30.16	129.47	229.74	361.03	145.19	16.72	184.28	163.32	20.09
bin	512	24	41.16	17.29	26.79	11.19	24.01	14.38	16.12	37.96	7.52	2.03	161.56	15.20	124.45	12.39	233.50	217.90	169.25	9.39	2.36	57.23	25.41	10.14	27.21	21.89	9.39	10.18	1.51	31.20	39.49	58.76	256.48	452.80	717.60	288.12	32.69	271.31	320.12	39.24
bin	512	20	35.52	14.72	23.10	9.51	20.29	12.37	13.86	31.94	6.01	2.19	134.79	12.44	104.18	9.97	195.89	182.78	140.37	7.29	1.99	47.82	21.41	8.44	22.71	18.55	7.92	8.55	1.26	26.02	33.81	49.23	213.80	379.90	599.31	240.51	27.53	242.56	268.04	32.90
bin	512	16	28.76	12.06	18.79	7.46	17.11	9.86	11.26	27.26	5.48	1.93	109.09	10.61	84.74	9.02	157.76	147.11	113.60	6.13	1.59	38.69	17.28	6.84	18.32	14.86	6.43	6.88	1.08	20.75	27.27	39.87	171.96	305.37	480.38	193.30	22.21	222.12	215.85	26.51
bin	512	12	22.82	9.68	14.51	6.29	13.73	7.76	9.07	21.93	4.26	1.86	83.77	8.50	64.73	6.60	119.28	111.99	88.76	4.76	1.23	29.23	13.11	5.23	13.83	11.28	4.84	5.38	0.79	16.09	20.97	30.24	129.17	230.17	361.86	145.70	16.76	192.87	163.68	20.17
bin	256	24	39.66	17.09	26.68	11.15	23.22	13.48	16.29	36.76	7.58	1.59	160.77	14.60	124.88	11.38	232.78	218.17	168.66	8.74	2.38	57.05	25.40	10.07	27.25	21.91	9.46	10.19	1.53	31.12	39.77	58.64	257.04	454.56	718.01	288.01	32.62	276.95	320.19	39.49
bin	256	20	34.26	14.41	22.41	9.33	19.84	11.78	13.37	31.53	6.30	1.59	135.47	12.14	104.53	9.53	195.42	181.81	141.83	7.02	1.96	47.58	21.31	8.43	22.68	18.32	7.97	8.55	1.29	26.00	32.97	47.89	209.44	379.50	599.87	234.67	26.96	239.40	261.88	32.15
bin	256	16	28.27	12.37	18.23	7.66	15.88	9.82	11.51	26.64	5.13	1.61	108.55	9.97	83.82	8.31	156.55	146.16	114.81	5.75	1.57	38.31	17.18	6.84	18.16	14.85	6.38	6.97	1.02	20.97	26.47	38.61	167.42	298.11	469.80	188.37	21.68	208.99	210.50	25.81
bin	256	12	22.05	9.23	14.57	5.75	13.00	7.99	8.59	21.16	3.84	1.52	82.32	8.10	64.47	6.08	118.91	111.46	85.92	5.03	1.22	28.91	13.12	5.17	13.83	11.19	4.83	5.31	0.80	15.88	20.33	29.36	125.96	224.82	352.84	142.08	16.33	178.72	159.23	19.60
bin	128	24	39.58	16.92	26.30	11.14	23.08	13.43	15.72	36.72	6.11	1.55	160.55	14.50	124.65	11.61	232.66	218.19	168.18	8.58	2.38	56.78	25.52	10.10	27.16	21.75	9.46	10.23	1.55	31.13	39.49	57.28	251.05	444.32	702.13	281.10	32.01	278.34	313.59	38.71
bin	128	20	33.95	14.14	22.07	9.38	19.65	11.76	13.34	31.06	6.45	1.47	134.93	12.00	104.32	9.39	195.16	182.25	141.22	7.34	1.96	47.74	21.26	8.44	22.71	18.35	7.93	8.53	1.25	26.03	33.25	47.89	209.58	372.25	586.34	234.97	26.89	246.89	262.10	32.32
bin	128	16	28.85	11.92	17.79	7.57	15.61	9.76	11.51	26.28	4.67	1.40	108.39	9.79	83.95	8.17	156.22	145.77	113.57	6.15	1.59	38.24	17.21	6.84	18.14	14.83	6.33	6.94	1.05	20.92	27.25	38.74	168.65	298.45	470.87	189.22	21.75	219.58	208.68	26.03
bin	128	12	21.62	8.91	13.84	5.74	12.39	7.41	8.46	20.86	3.80	1.34	82.17	8.01	64.38	5.95	118.19	110.93	85.66	4.74	1.24	29.03	13.01	5.20	13.78	11.25	4.85	5.21	0.82	15.98	21.02	29.58	126.86	225.83	354.58	142.59	16.42	190.54	157.06	19.77
bin	64	24	39.78	17.16	26.40	10.90	22.87	13.57	15.79	36.44	6.18	1.54	160.98	14.32	124.57	11.40	232.25	216.57	166.81	8.12	2.35	57.05	25.43	10.07	27.15	21.78	9.45	10.24	1.53	31.05	39.70	57.41	251.47	445.38	702.74	281.42	32.08	278.21	313.72	38.64
bin	64	20	33.94	14.51	21.92	9.04	19.45	11.59	13.66	31.17	5.91	1.41	134.91	11.93	103.84	9.21	194.46	180.86	139.30	6.97	1.95	47.72	21.23	8.47	22.68	18.31	7.93	8.53	1.27	25.96	33.55	48.12	209.72	371.67	587.46	235.40	26.95	248.75	262.71	32.37
bin	64	16	28.57	11.68	17.73	7.33	15.92	10.17	11.39	26.12	4.73	1.41	108.56	9.76	83.97	7.65	156.26	146.27	112.41	6.21	1.57	38.31	17.16	6.80	18.15	14.78	6.38	6.90	1.05	20.83	27.35	38.81	168.40	299.07	471.15	189.06	21.79	218.61	211.80	26.04
bin	64	12	24.29	9.97	15.26	6.00	13.55	8.04	9.48	21.92	4.44	1.30	82.44	7.98	70.01	6.36	129.85	121.73	91.90	4.95	1.30	31.72	14.02	5.57	14.88	12.46	5.30	5.61	0.82	17.20	21.00	29.65	126.77	225.83	354.35	142.53	16.38	187.13	160.41	19.69
bin	32	24	43.31	18.31	29.08	11.80	25.02	14.21	17.39	39.73	6.36	1.12	172.97	15.47	132.02	11.47	253.98	236.81	178.78	8.67	2.53	63.05	27.83	11.15	29.63	23.89	10.35	11.23	1.61	33.58	39.46	57.30	250.91	444.16	701.06	281.49	32.00	275.86	313.64	38.55
bin	32	20	36.66	15.40	24.56	9.88	21.24	12.25	14.50	33.35	5.08	1.03	144.77	12.89	110.68	9.69	212.17	197.77	149.55	7.21	2.13	52.36	23.17	9.36	25.16	19.98	8.59	9.31	1.39	28.02	33.02	47.93	208.92	371.91	587.32	235.49	26.83	246.56	262.52	32.26
bin	32	16	29.88	12.41	19.61	8.04	17.45	9.71	11.48	27.56	4.85	1.05	117.37	10.55	88.96	7.96	170.51	160.22	121.27	5.88	1.73	42.09	18.63	7.39	19.98	16.29	6.90	7.51	1.09	22.64	26.73	38.65	167.91	298.02	470.39	188.70	21.65	209.86	210.81	25.82
bin	32	12	23.43	9.86	14.69	6.14	13.36	7.62	9.11	21.79	3.96	0.90	88.78	8.11	68.18	6.07	129.47	120.20	91.59	4.49	1.28	31.48	14.24	5.63	15.03	12.30	5.30	5.71	0.85	17.08	20.26	29.98	126.19	225.11	357.97	145.08	16.59	175.74	158.92	19.54
bin	16	24	42.02	17.66	26.88	11.01	23.74	13.37	15.80	37.81	6.26	1.32	163.76	14.40	126.96	11.57	240.58	224.23	170.05	8.34	2.36	58.39	26.13	10.39	27.61	22.44	9.74	10.49	1.49	31.11	38.76	57.15	249.21	445.46	701.85	284.68	31.79	264.80	311.07	38.60
bin	16	20	35.03	14.94	22.68	9.38	19.93	11.36	13.44	32.14	5.05	1.39	137.86	12.18	105.63	9.36	201.18	187.82	142.87	7.19	1.96	48.60	22.12	8.66	23.28	18.70	8.12	8.74	1.28	26.27	32.51	47.90	208.39	372.35	588.50	238.53	26.79	236.92	260.42	32.17
bin	16	16	29.24	12.30	18.46	7.60	16.44	9.52	10.82	26.81	4.44	1.75	111.40	10.19	86.24	7.72	161.83	152.16	117.15	5.75	1.58	39.30	17.56	6.92	18.68	15.05	6.48	7.11	1.04	21.25	26.38	38.65	166.91	299.63	472.75	191.66	21.48	205.48	209.53	25.98
bin	16	12	23.05	9.48	14.45	5.80	13.23	7.12	8.28	21.19	3.52	1.19	85.25	7.60	66.03	5.87	122.14	115.03	88.79	4.48	1.21	30.24	13.50	5.30	13.99	11.30	4.87	5.27	0.79	16.27	19.98	29.81	125.65	226.18	356.15	144.24	16.36	175.21	158.92	19.51
bin	8	24	40.99	17.64	26.81	10.98	23.28	13.57	15.83	37.38	6.27	1.37	164.47	14.34	126.81	11.59	239.52	223.58	170.43	8.31	2.37	58.61	26.33	10.31	27.81	22.51	9.73	10.48	1.52	31.27	41.19	61.19	268.37	476.17	753.13	303.56	34.42	289.56	336.57	40.85
bin	8	20	34.50	14.69	23.03	9.32	19.64	11.62	13.70	31.98	4.87	1.22	134.79	12.60	105.94	9.83	200.46	189.49	142.67	6.98	1.95	48.74	21.96	8.60	23.13	18.73	8.11	8.79	1.26	26.56	34.37	51.15	223.66	400.71	628.14	253.44	28.70	255.39	280.61	34.16
bin	8	16	28.06	12.14	18.34	7.55	16.18	9.34	10.76	26.50	4.25	1.59	109.04	10.07	86.09	7.67	161.42	151.24	117.16	5.64	1.57	39.26	17.67	6.96	18.61	15.11	6.53	7.06	1.02	21.21	29.31	41.36	179.20	320.83	505.65	204.40	23.25	223.10	225.26	27.51
bin	8	12	22.58	9.16	14.11	5.91	12.85	7.06	8.49	20.85	3.92	1.31	81.72	7.60	65.38	5.99	122.04	114.37	87.72	4.59	1.18	29.80	13.51	5.26	14.15	11.34	4.93	5.31	0.74	16.21	22.18	31.48	135.18	243.99	381.09	154.61	17.76	191.76	172.58	20.25

Table 33: Absolute evaluation time for mxbai-embed-2d-large-v1 on all tasks using binary quantization. Table 2 serves as a legend.

q	m	l	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	1024	24	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	1024	20	0.889	0.866	0.847	0.851	0.932	0.925	0.830	0.943	0.929	1.030	0.845	0.870	0.845	0.887	0.841	0.847	0.853	0.833	0.823	0.841	0.837	0.824	0.840	0.851	0.846	0.832	0.870	0.831	0.845	0.839	0.837	0.831	0.836	0.836	0.842	0.890	0.835	0.837
f32	1024	16	0.799	0.711	0.723	0.692	0.857	0.852	0.730	0.856	0.895	0.865	0.678	0.846	0.676	0.777	0.687	0.694	0.700	0.721	0.673	0.679	0.676	0.665	0.662	0.686	0.693	0.680	0.688	0.667	0.700	0.668	0.672	0.667	0.671	0.670	0.679	0.785	0.666	0.675
f32	1024	12	0.716	0.562	0.572	0.548	0.777	0.767	0.576	0.825	0.697	1.027	0.528	0.687	0.539	0.772	0.526	0.542	0.569	0.687	0.515	0.523	0.519	0.498	0.503	0.515	0.519	0.526	0.560	0.512	0.544	0.497	0.507	0.504	0.504	0.506	0.512	0.686	0.496	0.512
f32	512	24	0.838	0.976	0.993	1.015	0.854	0.566	0.955	0.788	0.816	0.800	0.980	0.899	0.962	0.970	0.985	0.977	0.940	0.817	0.989	0.999	1.001	0.991	0.999	1.002	1.006	1.000	1.023	0.993	0.971	1.006	0.996	0.992	0.998	0.998	1.001	0.959	1.030	0.997
f32	512	20	0.764	0.839	0.831	0.841	0.787	0.514	0.809	0.727	0.690	0.610	0.829	0.835	0.808	0.861	0.833	0.824	0.795	0.713	0.820	0.846	0.835	0.819	0.840	0.852	0.844	0.832	0.868	0.831	0.819	0.841	0.830	0.830	0.834	0.833	0.842	0.853	0.859	0.832
f32	512	16	0.677	0.702	0.680	0.685	0.706	0.427	0.665	0.658	0.618	0.632	0.664	0.702	0.663	0.699	0.668	0.670	0.654	0.621	0.664	0.680	0.679	0.669	0.662	0.682	0.689	0.681	0.694	0.670	0.670	0.682	0.668	0.665	0.668	0.670	0.679	0.742	0.693	0.673
f32	512	12	0.575	0.547	0.537	0.564	0.630	0.333	0.521	0.599	0.579	0.653	0.516	0.572	0.510	0.571	0.506	0.510	0.522	0.507	0.514	0.516	0.524	0.508	0.502	0.513	0.529	0.512	0.582	0.512	0.516	0.516	0.504	0.502	0.503	0.505	0.511	0.642	0.524	0.509
f32	256	24	0.763	0.960	0.984	1.007	0.572	0.554	0.910	0.698	0.539	0.370	0.976	0.905	0.962	0.784	0.978	0.955	0.910	0.826	0.994	1.004	1.002	0.992	0.996	1.004	1.003	1.000	1.024	0.987	0.965	1.006	0.998	0.994	1.001	0.999	1.005	0.980	1.029	1.001
f32	256	20	0.677	0.833	0.829	0.848	0.500	0.487	0.776	0.624	0.491	0.407	0.816	0.786	0.795	0.670	0.822	0.807	0.761	0.652	0.842	0.838	0.837	0.816	0.835	0.850	0.840	0.837	0.876	0.826	0.808	0.842	0.832	0.832	0.835	0.836	0.843	0.878	0.862	0.838
f32	256	16	0.588	0.660	0.680	0.678	0.417	0.383	0.636	0.549	0.432	0.377	0.660	0.604	0.648	0.568	0.661	0.657	0.622	0.554	0.667	0.673	0.677	0.659	0.661	0.680	0.695	0.679	0.687	0.665	0.658	0.677	0.667	0.666	0.671	0.668	0.678	0.736	0.692	0.669
f32	256	12	0.498	0.521	0.523	0.538	0.328	0.302	0.488	0.471	0.374	0.359	0.505	0.489	0.509	0.502	0.499	0.490	0.486	0.473	0.516	0.511	0.516	0.498	0.502	0.511	0.512	0.518	0.541	0.506	0.504	0.517	0.502	0.502	0.503	0.504	0.512	0.632	0.524	0.511
f32	128	24	0.607	0.930	0.958	0.991	0.535	0.532	0.881	0.510	0.467	0.251	0.972	0.830	0.951	0.758	0.974	0.946	0.886	0.701	0.981	0.990	1.002	0.984	0.999	0.997	0.998	1.000	1.036	0.989	0.942	0.978	0.970	0.968	0.975	0.976	0.979	0.926	1.006	0.972
f32	128	20	0.521	0.808	0.812	0.829	0.458	0.459	0.766	0.431	0.486	0.255	0.816	0.694	0.796	0.675	0.820	0.801	0.747	0.619	0.816	0.828	0.844	0.822	0.830	0.839	0.842	0.829	0.838	0.831	0.793	0.820	0.811	0.810	0.814	0.816	0.821	0.828	0.841	0.817
f32	128	16	0.428	0.649	0.669	0.669	0.377	0.367	0.613	0.363	0.397	0.253	0.656	0.569	0.640	0.513	0.656	0.643	0.602	0.511	0.652	0.664	0.681	0.656	0.663	0.677	0.689	0.672	0.704	0.666	0.645	0.659	0.656	0.650	0.652	0.655	0.662	0.725	0.676	0.656
f32	128	12	0.351	0.536	0.526	0.531	0.310	0.300	0.507	0.300	0.309	0.303	0.501	0.491	0.496	0.405	0.498	0.488	0.463	0.394	0.507	0.501	0.516	0.509	0.502	0.514	0.512	0.507	0.545	0.512	0.500	0.504	0.493	0.492	0.491	0.494	0.502	0.653	0.513	0.499
f32	64	24	0.600	0.926	0.952	0.986	0.515	0.525	0.874	0.489	0.463	0.204	0.971	0.842	0.943	0.701	0.971	0.946	0.880	0.683	0.969	0.990	1.002	0.989	0.994	1.000	1.004	0.995	1.021	0.993	0.958	0.982	0.977	0.973	0.977	0.979	0.982	0.961	0.998	0.981
f32	64	20	0.512	0.796	0.822	0.837	0.437	0.459	0.744	0.418	0.385	0.211	0.814	0.672	0.784	0.581	0.813	0.787	0.737	0.580	0.830	0.826	0.841	0.819	0.830	0.847	0.848	0.830	0.863	0.829	0.809	0.822	0.815	0.813	0.817	0.818	0.825	0.858	0.843	0.824
f32	64	16	0.419	0.654	0.663	0.667	0.362	0.361	0.611	0.349	0.444	0.207	0.654	0.552	0.636	0.489	0.653	0.642	0.596	0.491	0.671	0.666	0.681	0.656	0.667	0.678	0.688	0.676	0.704	0.664	0.659	0.664	0.655	0.653	0.654	0.655	0.666	0.761	0.681	0.663
f32	64	12	0.338	0.500	0.532	0.512	0.293	0.309	0.486	0.277	0.284	0.330	0.501	0.442	0.486	0.388	0.497	0.487	0.455	0.435	0.527	0.501	0.515	0.500	0.503	0.512	0.512	0.519	0.538	0.510	0.506	0.505	0.494	0.494	0.494	0.495	0.502	0.656	0.515	0.502
f32	32	24	0.640	0.999	1.061	1.093	0.561	0.545	0.985	0.526	0.568	0.289	1.044	0.863	1.001	0.790	1.065	1.028	0.953	0.771	1.046	1.088	1.100	1.094	1.090	1.087	1.093	1.089	1.080	1.074	0.956	0.982	0.977	0.972	0.976	0.977	0.983	0.960	1.007	0.978
f32	32	20	0.558	0.857	0.900	0.920	0.471	0.474	0.811	0.452	0.450	0.187	0.882	0.762	0.837	0.629	0.891	0.859	0.806	0.664	0.873	0.909	0.913	0.912	0.914	0.912	0.912	0.905	0.901	0.899	0.803	0.821	0.816	0.812	0.817	0.818	0.824	0.858	0.844	0.820
f32	32	16	0.448	0.719	0.711	0.755	0.389	0.399	0.669	0.372	0.423	0.311	0.706	0.592	0.679	0.509	0.714	0.691	0.639	0.507	0.714	0.736	0.732	0.721	0.731	0.740	0.731	0.728	0.756	0.725	0.662	0.661	0.654	0.653	0.656	0.656	0.664	0.763	0.680	0.662
f32	32	12	0.345	0.554	0.544	0.567	0.299	0.301	0.519	0.296	0.290	0.202	0.539	0.499	0.517	0.453	0.539	0.526	0.486	0.389	0.539	0.554	0.554	0.546	0.551	0.562	0.561	0.550	0.587	0.541	0.509	0.506	0.492	0.493	0.493	0.495	0.503	0.656	0.515	0.501
f32	16	24	0.630	0.974	1.041	1.076	0.546	0.549	0.938	0.521	0.457	0.192	1.044	0.853	0.997	0.728	1.063	1.035	0.951	0.768	1.058	1.086	1.098	1.092	1.092	1.089	1.099	1.095	1.094	1.067	0.934	0.974	0.969	0.973	0.981	0.993	0.976	0.915	1.002	0.974
f32	16	20	0.536	0.843	0.862	0.908	0.467	0.482	0.816	0.462	0.401	0.259	0.873	0.711	0.844	0.649	0.889	0.859	0.793	0.642	0.842	0.824	0.862	0.843	0.837	0.843	0.849	0.842	0.850	0.851	0.786	0.817	0.811	0.814	0.819	0.828	0.822	0.817	0.838	0.818
f32	16	16	0.387	0.644	0.641	0.686	0.362	0.371	0.653	0.348	0.390	0.218	0.675	0.584	0.649	0.545	0.663	0.653	0.622	0.490	0.647	0.664	0.688	0.676	0.667	0.679	0.685	0.678	0.661	0.673	0.643	0.642	0.644	0.656	0.658	0.665	0.651	0.700	0.675	0.648
f32	16	12	0.330	0.512	0.508	0.514	0.277	0.285	0.499	0.277	0.282	0.206	0.504	0.443	0.496	0.405	0.503	0.495	0.470	0.435	0.493	0.519	0.520	0.515	0.504	0.511	0.511	0.510	0.523	0.517	0.494	0.498	0.490	0.492	0.497	0.504	0.504	0.614	0.499	0.495
f32	8	24	0.598	0.953	0.972	0.997	0.519	0.514	0.890	0.496	0.425	0.266	0.992	0.827	0.959	0.699	1.007	0.976	0.913	0.680	0.983	1.011	1.038	1.015	1.014	1.029	1.033	1.030	0.993	0.995	0.932	0.975	0.969	0.973	0.981	0.992	0.973	0.909	1.000	0.977
f32	8	20	0.528	0.820	0.808	0.836	0.441	0.445	0.757	0.422	0.364	0.171	0.831	0.679	0.798	0.590	0.838	0.825	0.760	0.580	0.823	0.849	0.872	0.840	0.849	0.856	0.868	0.857	0.857	0.841	0.849	0.886	0.870	0.815	0.878	0.884	0.901	0.901	0.905	0.883
f32	8	16	0.418	0.650	0.659	0.687	0.360	0.383	0.618	0.354	0.304	0.161	0.669	0.556	0.648	0.492	0.678	0.659	0.613	0.476	0.659	0.680	0.694	0.678	0.679	0.688	0.687	0.695	0.668	0.672	0.687	0.716	0.699	0.704	0.705	0.712	0.708	0.796	0.730	0.713
f32	8	12	0.332	0.521	0.529	0.514	0.283	0.308	0.469	0.278	0.250	0.163	0.508	0.426	0.496	0.384	0.508	0.499	0.469	0.391	0.499	0.517	0.528	0.506	0.509	0.519	0.519	0.520	0.523	0.510	0.524	0.535	0.529	0.537	0.533	0.537	0.549	0.684	0.550	0.533

Table 34: Relative evaluation time for mxbai-embed-2d-large-v1 on all tasks using float quantization. Table 2 serves as a legend.

q	m	l	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
int	1024	24	1.032	0.970	0.988	1.012	1.043	0.990	0.958	1.039	0.986	0.939	1.008	1.015	0.983	1.098	1.003	1.008	0.993	1.071	0.995	1.008	1.006	1.001	1.004	1.005	1.009	1.008	1.013	0.989	1.010	1.004	1.001	0.991	0.999	1.000	1.007	0.994	1.004	1.000
int	1024	20	0.924	0.851	0.844	0.854	0.950	0.920	0.836	0.971	0.899	0.896	0.836	0.995	0.840	1.020	0.843	0.850	0.847	0.872	0.820	0.847	0.839	0.834	0.845	0.847	0.849	0.839	0.852	0.838	0.848	0.842	0.835	0.830	0.834	0.836	0.843	0.897	0.836	0.839
int	1024	16	0.835	0.709	0.684	0.705	0.847	0.814	0.701	0.876	0.854	0.965	0.684	0.797	0.693	0.832	0.689	0.696	0.699	0.794	0.674	0.682	0.682	0.665	0.670	0.687	0.698	0.681	0.712	0.667	0.702	0.685	0.675	0.666	0.669	0.671	0.681	0.792	0.673	0.674
int	1024	12	0.714	0.557	0.558	0.564	0.785	0.733	0.571	0.796	0.797	1.131	0.528	0.664	0.536	0.725	0.526	0.540	0.567	0.654	0.510	0.522	0.527	0.516	0.506	0.519	0.522	0.515	0.557	0.514	0.557	0.520	0.509	0.503	0.504	0.505	0.513	0.676	0.503	0.512
int	512	24	0.850	0.982	0.998	1.038	0.864	0.562	0.979	0.797	0.730	0.742	0.986	0.942	0.969	0.892	0.986	0.978	0.930	0.826	0.994	1.003	1.007	0.994	1.000	1.000	1.006	1.005	1.046	0.987	0.971	1.006	0.998	0.992	0.999	0.998	1.004	0.953	1.028	0.994
int	512	20	0.755	0.841	0.833	0.854	0.799	0.512	0.810	0.734	0.742	0.810	0.822	0.838	0.808	0.747	0.829	0.820	0.789	0.814	0.822	0.846	0.838	0.820	0.842	0.852	0.844	0.834	0.871	0.825	0.820	0.843	0.834	0.828	0.834	0.834	0.843	0.847	0.860	0.833
int	512	16	0.672	0.698	0.680	0.712	0.706	0.414	0.674	0.649	0.573	0.603	0.666	0.664	0.671	0.728	0.667	0.668	0.646	0.619	0.669	0.680	0.678	0.668	0.662	0.683	0.691	0.681	0.704	0.670	0.670	0.681	0.668	0.665	0.671	0.671	0.677	0.745	0.693	0.673
int	512	12	0.590	0.551	0.537	0.583	0.636	0.315	0.522	0.583	0.591	0.719	0.514	0.605	0.522	0.644	0.507	0.507	0.509	0.516	0.501	0.515	0.522	0.517	0.501	0.515	0.520	0.522	0.541	0.513	0.517	0.516	0.503	0.503	0.502	0.504	0.513	0.640	0.524	0.508
int	256	24	0.768	0.943	0.976	1.018	0.564	0.527	0.924	0.677	0.541	0.500	0.971	0.841	0.944	0.807	0.979	0.952	0.892	0.750	1.000	0.997	0.999	0.988	0.997	0.999	1.004	0.996	1.012	0.994	0.964	1.005	0.996	0.989	1.000	1.000	1.002	1.003	1.031	1.000
int	256	20	0.662	0.802	0.817	0.857	0.481	0.470	0.789	0.607	0.509	0.482	0.813	0.738	0.795	0.646	0.820	0.797	0.762	0.714	0.832	0.837	0.837	0.817	0.832	0.847	0.850	0.834	0.838	0.828	0.808	0.842	0.831	0.828	0.835	0.834	0.839	0.860	0.861	0.834
int	256	16	0.586	0.686	0.669	0.705	0.404	0.389	0.666	0.541	0.426	0.376	0.659	0.604	0.644	0.553	0.663	0.646	0.623	0.607	0.656	0.678	0.679	0.652	0.661	0.676	0.683	0.675	0.702	0.666	0.658	0.678	0.670	0.665	0.670	0.670	0.677	0.740	0.693	0.670
int	256	12	0.498	0.545	0.529	0.583	0.334	0.294	0.500	0.467	0.408	0.438	0.507	0.491	0.492	0.449	0.500	0.495	0.475	0.440	0.500	0.507	0.519	0.498	0.499	0.512	0.525	0.513	0.538	0.504	0.504	0.519	0.502	0.502	0.502	0.505	0.513	0.648	0.523	0.510
int	128	24	0.606	0.949	0.967	1.019	0.536	0.523	0.899	0.508	0.498	0.245	0.976	0.847	0.946	0.708	0.977	0.945	0.879	0.701	0.994	0.989	1.002	0.986	0.999	0.997	1.000	1.000	1.028	0.987	0.944	0.980	0.973	0.970	0.976	0.975	0.981	0.945	1.006	0.974
int	128	20	0.515	0.834	0.823	0.867	0.460	0.452	0.792	0.431	0.498	0.272	0.815	0.700	0.792	0.611	0.818	0.798	0.743	0.629	0.829	0.828	0.840	0.818	0.829	0.843	0.846	0.831	0.852	0.828	0.809	0.818	0.816	0.812	0.815	0.817	0.824	0.860	0.843	0.820
int	128	16	0.443	0.679	0.680	0.696	0.380	0.364	0.633	0.363	0.364	0.259	0.658	0.595	0.639	0.515	0.656	0.639	0.603	0.561	0.684	0.662	0.676	0.663	0.665	0.678	0.680	0.674	0.694	0.665	0.661	0.661	0.655	0.653	0.655	0.656	0.665	0.760	0.678	0.662
int	128	12	0.350	0.524	0.551	0.547	0.309	0.303	0.533	0.298	0.320	0.388	0.504	0.446	0.488	0.409	0.499	0.488	0.459	0.465	0.521	0.502	0.515	0.509	0.505	0.513	0.514	0.510	0.530	0.506	0.510	0.505	0.493	0.493	0.492	0.495	0.503	0.653	0.513	0.501
int	64	24	0.601	0.973	0.982	1.028	0.519	0.511	0.901	0.486	0.450	0.223	0.971	0.796	0.941	0.685	0.974	0.940	0.877	0.681	0.985	0.987	1.002	0.983	0.993	1.000	1.000	0.999	1.013	0.994	0.957	0.981	0.980	0.972	0.979	0.977	0.986	0.966	1.009	0.979
int	64	20	0.501	0.820	0.817	0.863	0.442	0.439	0.795	0.417	0.447	0.237	0.813	0.682	0.787	0.587	0.814	0.793	0.740	0.642	0.823	0.830	0.839	0.817	0.832	0.846	0.837	0.827	0.874	0.830	0.805	0.824	0.816	0.812	0.816	0.818	0.823	0.866	0.844	0.824
int	64	16	0.432	0.699	0.674	0.726	0.355	0.376	0.659	0.351	0.349	0.225	0.654	0.574	0.635	0.495	0.653	0.634	0.601	0.477	0.668	0.665	0.678	0.665	0.663	0.680	0.681	0.681	0.686	0.669	0.657	0.667	0.655	0.651	0.654	0.657	0.664	0.759	0.680	0.657
int	64	12	0.334	0.529	0.550	0.543	0.292	0.304	0.525	0.279	0.320	0.198	0.497	0.450	0.491	0.389	0.494	0.482	0.453	0.379	0.516	0.500	0.512	0.503	0.502	0.514	0.517	0.523	0.534	0.510	0.508	0.507	0.496	0.494	0.493	0.495	0.502	0.657	0.515	0.500
int	32	24	0.636	1.022	1.079	1.111	0.549	0.558	0.975	0.532	0.472	0.321	1.047	0.854	1.002	0.732	1.063	1.033	0.951	0.792	1.066	1.081	1.111	1.082	1.086	1.090	1.100	1.104	1.091	1.070	0.953	0.980	0.977	0.972	0.978	0.976	0.981	0.960	1.007	0.976
int	32	20	0.547	0.873	0.897	0.936	0.470	0.485	0.826	0.451	0.403	0.199	0.876	0.761	0.838	0.628	0.888	0.860	0.789	0.614	0.872	0.915	0.912	0.904	0.919	0.912	0.919	0.912	0.905	0.902	0.809	0.820	0.815	0.812	0.817	0.816	0.823	0.858	0.843	0.822
int	32	16	0.438	0.738	0.723	0.768	0.386	0.400	0.704	0.372	0.347	0.203	0.706	0.625	0.677	0.545	0.716	0.691	0.637	0.507	0.707	0.740	0.738	0.719	0.732	0.741	0.729	0.730	0.737	0.722	0.657	0.663	0.654	0.653	0.655	0.655	0.667	0.750	0.679	0.660
int	32	12	0.348	0.569	0.590	0.580	0.300	0.317	0.541	0.298	0.352	0.182	0.536	0.460	0.516	0.436	0.538	0.523	0.494	0.385	0.552	0.543	0.559	0.550	0.542	0.556	0.563	0.549	0.593	0.550	0.505	0.506	0.494	0.493	0.494	0.495	0.501	0.652	0.514	0.501
int	16	24	0.601	0.936	0.957	1.023	0.513	0.532	0.915	0.487	0.450	0.227	0.983	0.814	0.956	0.696	0.983	0.953	0.899	0.738	0.974	1.004	1.019	1.000	0.993	1.009	1.012	1.011	0.977	1.006	0.902	0.939	0.951	0.973	0.980	0.991	0.962	0.903	0.958	0.943
int	16	20	0.498	0.794	0.809	0.879	0.435	0.436	0.749	0.414	0.399	0.227	0.841	0.694	0.808	0.620	0.837	0.813	0.751	0.591	0.810	0.837	0.852	0.836	0.829	0.845	0.852	0.836	0.842	0.838	0.788	0.820	0.811	0.805	0.810	0.828	0.823	0.816	0.838	0.816
int	16	16	0.416	0.660	0.673	0.704	0.361	0.371	0.655	0.358	0.370	0.241	0.681	0.603	0.655	0.501	0.681	0.659	0.608	0.544	0.645	0.672	0.698	0.682	0.684	0.689	0.691	0.689	0.684	0.685	0.641	0.654	0.645	0.652	0.657	0.667	0.664	0.721	0.671	0.660
int	16	12	0.335	0.519	0.521	0.542	0.280	0.289	0.519	0.283	0.318	0.276	0.513	0.489	0.495	0.439	0.511	0.496	0.465	0.439	0.499	0.517	0.528	0.506	0.510	0.521	0.523	0.522	0.494	0.515	0.483	0.483	0.479	0.496	0.496	0.504	0.501	0.614	0.508	0.494
int	8	24	0.602	0.974	0.983	1.013	0.520	0.518	0.905	0.497	0.489	0.224	0.995	0.846	0.952	0.700	1.001	0.975	0.894	0.708	0.981	1.009	1.036	1.014	1.010	1.027	1.036	1.026	0.993	0.994	0.948	1.001	1.017	1.046	1.049	1.055	1.059	1.013	1.036	1.060
int	8	20	0.507	0.833	0.818	0.860	0.438	0.441	0.767	0.422	0.440	0.222	0.837	0.707	0.808	0.632	0.840	0.816	0.758	0.595	0.819	0.848	0.873	0.844	0.845	0.857	0.863	0.859	0.835	0.837	0.802	0.843	0.851	0.861	0.878	0.875	0.866	0.881	0.862	0.892
int	8	16	0.431	0.678	0.690	0.684	0.351	0.362	0.625	0.353	0.409	0.213	0.674	0.586	0.653	0.495	0.675	0.656	0.609	0.503	0.652	0.677	0.700	0.677	0.677	0.691	0.700	0.686	0.699	0.682	0.652	0.678	0.693	0.698	0.706	0.694	0.676	0.771	0.686	0.683
int	8	12	0.354	0.513	0.524	0.563	0.295	0.287	0.492	0.284	0.325	0.203	0.515	0.445	0.502	0.390	0.512	0.497	0.466	0.438	0.509	0.526	0.532	0.517	0.511	0.519	0.520	0.520	0.524	0.514	0.521	0.541	0.528	0.536	0.531	0.540	0.544	0.674	0.547	0.551

Table 35: Relative evaluation time for mxbai-embed-2d-large-v1 on all tasks using int quantization. Table 2 serves as a legend.

q	m	l	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
bin	1024	24	0.620	0.944	0.970	1.007	0.542	0.529	0.907	0.520	0.540	0.381	0.974	0.825	0.941	0.723	0.976	0.949	0.884	0.724	0.981	0.993	1.003	0.996	0.996	1.005	1.001	0.997	1.014	0.992	0.956	1.005	0.994	0.991	0.998	0.998	1.004	0.941	1.028	0.997
bin	1024	20	0.545	0.820	0.811	0.838	0.469	0.451	0.784	0.445	0.433	0.286	0.813	0.727	0.791	0.629	0.824	0.793	0.753	0.649	0.830	0.831	0.843	0.820	0.833	0.851	0.848	0.832	0.840	0.827	0.806	0.841	0.834	0.828	0.834	0.834	0.837	0.843	0.859	0.835
bin	1024	16	0.435	0.664	0.682	0.679	0.390	0.377	0.631	0.375	0.376	0.294	0.658	0.600	0.639	0.544	0.660	0.641	0.603	0.528	0.664	0.668	0.681	0.669	0.665	0.680	0.694	0.675	0.696	0.667	0.656	0.678	0.668	0.666	0.669	0.669	0.677	0.741	0.692	0.668
bin	1024	12	0.360	0.518	0.539	0.529	0.318	0.316	0.531	0.311	0.316	0.265	0.502	0.504	0.495	0.419	0.500	0.495	0.462	0.488	0.514	0.507	0.519	0.513	0.505	0.517	0.520	0.522	0.560	0.507	0.502	0.515	0.503	0.502	0.502	0.504	0.511	0.627	0.524	0.509
bin	512	24	0.603	0.940	0.963	1.010	0.525	0.538	0.886	0.498	0.509	0.227	0.974	0.840	0.937	0.749	0.974	0.943	0.887	0.761	0.986	0.991	1.004	0.988	0.993	1.001	0.998	0.995	1.007	0.995	0.951	1.003	0.997	0.990	0.999	1.000	1.000	0.924	1.028	0.995
bin	512	20	0.521	0.800	0.830	0.858	0.444	0.462	0.762	0.419	0.406	0.244	0.812	0.687	0.785	0.602	0.817	0.791	0.736	0.590	0.833	0.828	0.846	0.823	0.829	0.848	0.843	0.835	0.843	0.830	0.814	0.841	0.831	0.831	0.834	0.835	0.842	0.826	0.861	0.834
bin	512	16	0.422	0.656	0.675	0.674	0.374	0.369	0.619	0.358	0.371	0.215	0.657	0.586	0.638	0.545	0.658	0.637	0.596	0.497	0.665	0.670	0.683	0.667	0.669	0.680	0.684	0.672	0.721	0.662	0.657	0.681	0.668	0.667	0.668	0.671	0.679	0.756	0.693	0.672
bin	512	12	0.335	0.526	0.522	0.568	0.300	0.290	0.499	0.288	0.288	0.208	0.505	0.469	0.487	0.399	0.497	0.485	0.465	0.385	0.513	0.506	0.518	0.510	0.505	0.516	0.515	0.526	0.529	0.513	0.505	0.516	0.502	0.503	0.503	0.506	0.513	0.657	0.526	0.511
bin	256	24	0.581	0.929	0.959	1.006	0.508	0.504	0.896	0.482	0.513	0.177	0.969	0.806	0.941	0.687	0.971	0.945	0.884	0.708	0.995	0.988	1.004	0.981	0.994	1.002	1.006	0.996	1.020	0.992	0.958	1.001	0.999	0.994	0.999	1.000	0.998	0.943	1.028	1.001
bin	256	20	0.502	0.783	0.805	0.843	0.434	0.441	0.735	0.414	0.426	0.177	0.816	0.670	0.787	0.576	0.815	0.787	0.744	0.569	0.818	0.824	0.842	0.821	0.827	0.838	0.848	0.836	0.860	0.829	0.794	0.818	0.814	0.830	0.835	0.815	0.825	0.815	0.841	0.815
bin	256	16	0.414	0.672	0.655	0.692	0.347	0.367	0.633	0.350	0.347	0.179	0.654	0.551	0.631	0.502	0.653	0.633	0.602	0.466	0.657	0.664	0.679	0.666	0.663	0.679	0.678	0.681	0.678	0.669	0.638	0.659	0.651	0.652	0.654	0.654	0.663	0.712	0.676	0.654
bin	256	12	0.323	0.502	0.524	0.519	0.284	0.299	0.472	0.278	0.260	0.170	0.496	0.447	0.485	0.368	0.496	0.482	0.451	0.408	0.511	0.501	0.518	0.504	0.505	0.511	0.513	0.519	0.533	0.506	0.490	0.501	0.490	0.491	0.491	0.493	0.499	0.609	0.511	0.497
bin	128	24	0.580	0.920	0.945	1.005	0.505	0.502	0.864	0.482	0.413	0.173	0.968	0.801	0.939	0.702	0.970	0.945	0.882	0.695	0.994	0.984	1.008	0.984	0.991	0.995	1.006	1.000	1.036	0.993	0.951	0.978	0.976	0.971	0.977	0.976	0.979	0.948	1.007	0.981
bin	128	20	0.498	0.769	0.793	0.847	0.430	0.440	0.734	0.408	0.436	0.164	0.813	0.663	0.786	0.567	0.814	0.789	0.740	0.595	0.821	0.827	0.840	0.823	0.829	0.839	0.844	0.834	0.834	0.830	0.801	0.818	0.815	0.814	0.816	0.816	0.823	0.841	0.842	0.820
bin	128	16	0.423	0.648	0.639	0.683	0.342	0.365	0.633	0.345	0.316	0.156	0.653	0.540	0.632	0.494	0.651	0.631	0.595	0.498	0.664	0.662	0.680	0.666	0.662	0.678	0.673	0.679	0.699	0.667	0.657	0.661	0.655	0.652	0.655	0.657	0.665	0.748	0.670	0.660
bin	128	12	0.317	0.484	0.497	0.518	0.271	0.277	0.465	0.274	0.257	0.149	0.495	0.442	0.485	0.360	0.493	0.480	0.449	0.384	0.519	0.503	0.514	0.506	0.503	0.514	0.515	0.509	0.548	0.510	0.506	0.505	0.493	0.494	0.493	0.495	0.502	0.649	0.504	0.501
bin	64	24	0.583	0.933	0.949	0.984	0.500	0.507	0.868	0.478	0.418	0.173	0.970	0.791	0.938	0.689	0.968	0.938	0.875	0.658	0.981	0.988	1.005	0.982	0.991	0.996	1.005	1.000	1.019	0.990	0.957	0.980	0.977	0.974	0.978	0.977	0.981	0.947	1.008	0.980
bin	64	20	0.497	0.789	0.788	0.816	0.425	0.433	0.751	0.409	0.400	0.157	0.813	0.659	0.782	0.557	0.811	0.783	0.730	0.565	0.816	0.827	0.839	0.826	0.827	0.837	0.843	0.833	0.850	0.828	0.808	0.822	0.815	0.812	0.817	0.817	0.824	0.847	0.844	0.821
bin	64	16	0.419	0.635	0.637	0.662	0.348	0.380	0.626	0.343	0.320	0.158	0.654	0.539	0.632	0.463	0.651	0.633	0.589	0.503	0.655	0.664	0.678	0.663	0.662	0.676	0.678	0.674	0.700	0.664	0.659	0.663	0.654	0.654	0.656	0.656	0.667	0.744	0.680	0.660
bin	64	12	0.356	0.542	0.549	0.541	0.296	0.301	0.521	0.288	0.300	0.145	0.497	0.441	0.527	0.384	0.541	0.527	0.482	0.401	0.545	0.549	0.554	0.543	0.543	0.570	0.563	0.548	0.548	0.548	0.506	0.506	0.493	0.494	0.493	0.495	0.501	0.637	0.515	0.499
bin	32	24	0.635	0.995	1.045	1.065	0.547	0.531	0.956	0.521	0.430	0.126	1.042	0.854	0.994	0.693	1.059	1.025	0.937	0.703	1.058	1.092	1.100	1.086	1.081	1.092	1.101	1.097	1.077	1.071	0.951	0.978	0.975	0.971	0.976	0.977	0.979	0.939	1.007	0.977
bin	32	20	0.537	0.837	0.883	0.892	0.465	0.458	0.797	0.438	0.344	0.115	0.872	0.712	0.834	0.586	0.885	0.856	0.784	0.584	0.891	0.907	0.916	0.913	0.918	0.914	0.914	0.909	0.930	0.893	0.796	0.818	0.812	0.813	0.817	0.818	0.821	0.840	0.843	0.818
bin	32	16	0.438	0.674	0.705	0.726	0.382	0.363	0.631	0.362	0.328	0.117	0.707	0.583	0.670	0.481	0.711	0.694	0.636	0.476	0.722	0.729	0.736	0.720	0.729	0.745	0.734	0.734	0.725	0.722	0.644	0.660	0.653	0.651	0.654	0.655	0.662	0.715	0.677	0.655
bin	32	12	0.343	0.536	0.528	0.554	0.292	0.285	0.501	0.286	0.268	0.100	0.535	0.448	0.513	0.367	0.540	0.520	0.480	0.364	0.536	0.545	0.563	0.549	0.548	0.562	0.563	0.558	0.568	0.544	0.488	0.512	0.490	0.492	0.498	0.504	0.507	0.598	0.510	0.495
bin	16	24	0.616	0.960	0.966	0.994	0.519	0.500	0.869	0.496	0.424	0.147	0.987	0.795	0.956	0.699	1.003	0.971	0.891	0.675	0.986	1.012	1.033	1.013	1.008	1.026	1.036	1.026	0.996	0.992	0.934	0.976	0.969	0.974	0.977	0.988	0.972	0.902	0.999	0.979
bin	16	20	0.513	0.812	0.815	0.847	0.436	0.425	0.739	0.422	0.342	0.156	0.831	0.672	0.795	0.566	0.839	0.813	0.749	0.583	0.821	0.842	0.874	0.844	0.849	0.855	0.863	0.854	0.856	0.838	0.783	0.818	0.810	0.814	0.819	0.828	0.819	0.807	0.836	0.816
bin	16	16	0.429	0.668	0.663	0.686	0.360	0.356	0.595	0.352	0.300	0.196	0.671	0.563	0.649	0.467	0.675	0.659	0.614	0.466	0.661	0.681	0.694	0.674	0.682	0.688	0.690	0.695	0.694	0.677	0.635	0.660	0.649	0.655	0.658	0.665	0.657	0.700	0.673	0.659
bin	16	12	0.338	0.515	0.519	0.523	0.289	0.266	0.456	0.278	0.238	0.133	0.514	0.419	0.497	0.355	0.509	0.498	0.466	0.363	0.504	0.524	0.533	0.516	0.511	0.517	0.517	0.515	0.526	0.519	0.481	0.509	0.488	0.494	0.496	0.501	0.500	0.597	0.510	0.495
bin	8	24	0.601	0.959	0.964	0.991	0.509	0.507	0.870	0.491	0.424	0.153	0.991	0.792	0.955	0.701	0.999	0.968	0.893	0.673	0.992	1.015	1.040	1.005	1.015	1.029	1.035	1.024	1.015	0.997	0.992	1.045	1.043	1.041	1.048	1.054	1.053	0.986	1.081	1.036
bin	8	20	0.506	0.798	0.828	0.842	0.430	0.434	0.753	0.420	0.329	0.136	0.812	0.696	0.798	0.594	0.836	0.820	0.748	0.566	0.816	0.844	0.868	0.838	0.844	0.857	0.862	0.859	0.844	0.847	0.828	0.873	0.869	0.876	0.874	0.880	0.878	0.870	0.901	0.866
bin	8	16	0.411	0.660	0.659	0.682	0.354	0.349	0.591	0.348	0.288	0.177	0.657	0.556	0.648	0.464	0.673	0.655	0.614	0.457	0.655	0.680	0.698	0.678	0.679	0.691	0.694	0.690	0.678	0.676	0.706	0.706	0.697	0.701	0.704	0.710	0.711	0.760	0.723	0.697
bin	8	12	0.331	0.498	0.507	0.533	0.281	0.264	0.467	0.274	0.265	0.146	0.492	0.419	0.492	0.362	0.509	0.495	0.460	0.372	0.495	0.516	0.534	0.512	0.516	0.519	0.524	0.519	0.493	0.517	0.534	0.537	0.525	0.533	0.530	0.537	0.543	0.653	0.554	0.513

Table 36: Relative evaluation time for mxbai-embed-2d-large-v1 on all tasks using binary quantization. Table 2 serves as a legend.

q	m	l	Average All Tasks	Average classification	Average clustering	Average sts	Average pairclass	Average retrieval	Average rerank	Average summ
f32	1024	24	105.78	36.54	103.10	19.75	119.12	488.06	212.59	39.45
f32	1024	20	89.69	32.98	87.60	16.57	99.84	407.12	182.91	33.03
f32	1024	16	73.59	29.47	71.90	13.33	80.37	326.82	153.38	26.63
f32	1024	12	57.84	26.32	57.21	10.19	60.70	246.09	124.24	20.18
f32	512	24	103.28	30.70	99.29	19.72	118.52	486.26	211.71	39.33
f32	512	20	87.37	27.47	84.02	16.58	98.95	406.33	181.82	32.83
f32	512	16	71.30	23.95	68.48	13.35	79.90	325.80	151.94	26.56
f32	512	12	55.36	20.41	53.25	10.16	60.43	245.47	122.80	20.09
f32	256	24	102.19	27.36	96.93	19.73	118.54	487.14	213.75	39.50
f32	256	20	86.19	23.94	81.39	16.51	98.92	407.05	184.63	33.05
f32	256	16	69.83	20.18	66.16	13.27	79.50	326.38	151.29	26.41
f32	256	12	53.80	16.49	50.87	10.07	60.17	245.61	121.86	20.16
f32	128	24	99.12	23.71	95.43	19.64	115.31	474.89	205.70	38.35
f32	128	20	83.59	20.27	80.61	16.43	96.54	396.72	177.28	32.22
f32	128	16	67.71	16.63	64.79	13.22	78.04	318.27	148.37	25.90
f32	128	12	52.41	13.58	49.72	10.03	59.06	240.21	122.62	19.68
f32	64	24	99.30	23.26	94.97	19.64	116.19	476.46	208.26	38.71
f32	64	20	83.58	19.92	79.30	16.44	97.12	398.09	180.39	32.51
f32	64	16	67.96	16.30	64.41	13.23	78.23	319.24	152.41	26.17
f32	64	12	52.32	13.04	49.34	10.03	59.24	241.09	123.08	19.81
f32	32	24	102.34	25.24	103.01	21.47	116.18	475.74	209.22	38.58
f32	32	20	86.20	21.56	86.42	17.95	97.10	398.04	180.60	32.34
f32	32	16	69.92	17.61	69.45	14.46	78.17	319.71	152.46	26.10
f32	32	12	53.59	13.61	52.92	10.90	59.15	240.91	123.16	19.77
f32	16	24	101.90	24.77	102.66	21.46	115.05	478.87	204.22	38.43
f32	16	20	85.39	21.29	85.97	16.60	96.43	399.79	175.93	32.25
f32	16	16	67.86	16.07	65.98	13.29	76.66	321.57	145.68	25.56
f32	16	12	52.08	12.78	50.05	10.16	58.57	242.48	117.44	19.51
f32	8	24	99.87	23.49	97.66	20.07	115.04	478.71	203.37	38.54
f32	8	20	87.53	20.11	81.67	16.82	103.64	419.59	191.97	34.82
f32	8	16	71.62	16.42	65.88	13.46	83.40	344.74	161.40	28.12
f32	8	12	55.00	12.94	50.08	10.19	63.09	261.28	129.96	21.02
int	1024	24	105.90	37.20	103.34	19.81	119.43	486.57	212.48	39.43
int	1024	20	89.89	33.59	87.72	16.64	99.77	406.56	183.79	33.09
int	1024	16	73.89	29.60	72.42	13.40	80.95	326.21	154.88	26.59
int	1024	12	57.78	25.91	57.17	10.23	61.48	245.93	124.01	20.20
int	512	24	103.27	31.04	99.15	19.75	118.64	486.37	210.91	39.22
int	512	20	87.30	27.55	83.81	16.58	99.30	406.13	181.36	32.85
int	512	16	71.28	23.81	68.26	13.35	79.86	326.48	152.23	26.53
int	512	12	55.36	20.40	53.34	10.16	60.40	245.34	122.68	20.02
int	256	24	102.02	27.05	96.17	19.69	118.38	486.38	216.07	39.44
int	256	20	85.80	23.41	81.13	16.49	98.89	406.34	182.74	32.91
int	256	16	69.82	20.16	65.91	13.28	79.78	326.38	151.75	26.43
int	256	12	53.86	16.62	50.62	10.04	60.21	245.43	123.38	20.12
int	128	24	99.33	23.82	95.30	19.62	115.61	475.44	207.70	38.43
int	128	20	83.92	20.42	80.29	16.43	97.17	397.55	180.60	32.34
int	128	16	68.17	16.95	64.79	13.20	78.20	319.38	152.01	26.11
int	128	12	52.48	13.69	49.72	10.03	59.19	240.59	122.70	19.77
int	64	24	99.46	23.56	94.71	19.62	116.39	476.51	210.08	38.63
int	64	20	83.77	19.96	79.77	16.45	97.16	397.83	181.34	32.51
int	64	16	67.98	16.77	64.22	13.24	78.25	319.13	152.15	25.92
int	64	12	52.30	13.26	48.98	10.02	59.46	241.08	123.25	19.71
int	32	24	102.34	25.35	102.89	21.46	116.12	476.11	209.15	38.49
int	32	20	86.05	21.57	85.83	18.00	97.09	398.01	180.39	32.41
int	32	16	69.79	17.70	69.35	14.48	78.14	319.37	151.15	26.02
int	32	12	53.62	13.97	52.88	10.85	59.23	240.94	122.66	19.78
int	16	24	98.74	23.44	96.16	19.85	112.37	478.35	198.34	37.20
int	16	20	83.61	19.69	81.64	16.57	96.49	396.16	175.75	32.17
int	16	16	68.27	16.58	66.51	13.48	77.00	320.72	147.45	26.02
int	16	12	52.27	13.05	50.56	10.21	57.23	242.65	118.33	19.51
int	8	24	104.07	23.69	97.21	20.02	119.91	512.14	218.31	41.83
int	8	20	87.28	20.01	81.96	16.80	100.49	425.64	185.13	35.18
int	8	16	70.76	16.58	66.12	13.50	81.73	342.03	154.03	26.94
int	8	12	55.10	13.26	50.51	10.28	63.04	260.98	128.74	21.73
bin	1024	24	100.69	24.11	95.59	19.68	118.08	485.88	209.77	39.34
bin	1024	20	84.92	20.72	80.48	16.48	99.08	406.23	180.87	32.95
bin	1024	16	68.89	17.04	65.00	13.27	79.58	326.08	151.74	26.34
bin	1024	12	53.00	13.92	50.08	10.10	60.16	245.32	121.44	20.09
bin	512	24	100.40	23.61	95.32	19.65	118.24	486.17	208.04	39.24
bin	512	20	84.46	20.16	79.59	16.47	98.95	406.57	179.38	32.90
bin	512	16	68.83	16.57	64.55	13.27	79.70	326.35	153.39	26.51
bin	512	12	52.97	13.22	49.45	10.10	60.13	245.91	124.44	20.17
bin	256	24	100.40	23.04	94.91	19.64	118.48	486.86	209.92	39.49
bin	256	20	83.72	19.62	79.56	16.41	96.77	404.68	176.08	32.15
bin	256	16	67.34	16.30	64.07	13.22	77.50	318.76	147.06	25.81
bin	256	12	51.57	12.79	48.77	10.03	58.55	239.91	118.09	19.60
bin	128	24	99.04	22.86	94.66	19.61	115.94	475.85	207.98	38.71
bin	128	20	83.33	19.42	79.45	16.42	96.91	397.85	178.63	32.32
bin	128	16	67.60	16.16	63.81	13.21	78.21	319.51	150.00	26.03
bin	128	12	51.82	12.40	48.52	10.04	59.15	241.00	121.34	19.77
bin	64	24	99.01	22.86	94.27	19.61	116.19	476.51	208.00	38.64
bin	64	20	83.28	19.41	78.88	16.41	97.13	398.18	179.47	32.37
bin	64	16	67.64	16.11	63.72	13.19	78.19	319.76	150.73	26.04
bin	64	12	53.21	13.56	52.10	10.89	59.14	240.90	121.31	19.69
bin	32	24	101.73	24.86	101.77	21.48	115.89	475.57	207.17	38.55
bin	32	20	85.55	20.98	85.08	17.95	96.62	398.24	178.64	32.26
bin	32	16	69.15	17.02	68.86	14.43	77.76	319.04	147.44	25.82
bin	32	12	52.95	13.25	52.17	10.89	58.81	242.72	117.08	19.54
bin	16	24	99.45	23.54	96.75	20.02	115.04	477.33	202.55	38.60
bin	16	20	83.73	19.86	81.05	16.77	96.27	399.79	174.71	32.17
bin	16	16	67.98	16.40	65.86	13.50	77.31	321.35	145.50	25.98
bin	16	12	52.03	12.83	49.99	10.27	58.48	242.19	116.83	19.51
bin	8	24	104.19	23.31	96.67	20.09	123.58	510.95	220.18	40.85
bin	8	20	87.51	19.81	80.88	16.78	103.06	427.43	188.23	34.16
bin	8	16	71.00	16.11	65.42	13.50	83.29	343.63	157.20	27.51
bin	8	12	54.45	12.63	49.46	10.24	62.95	259.90	127.37	20.25

Table 37: Evaluation time for mxbai-embed-2d-large-v1, averaged per category. Table 2 serves as a legend.

q	m	l	Average Relative classification	Average Relative clustering	Average Relative sts	Average Relative pairclass	Average Relative retrieval	Average Relative rerank	Average Relative summ	Average Relative all
f32	1024	24	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	1024	20	0.885	0.878	0.840	0.840	0.834	0.855	0.837	0.860
f32	1024	16	0.777	0.754	0.677	0.680	0.669	0.710	0.675	0.721
f32	1024	12	0.668	0.657	0.519	0.516	0.504	0.565	0.512	0.589
f32	512	24	0.873	0.914	1.000	0.991	0.996	0.997	0.997	0.950
f32	512	20	0.764	0.780	0.839	0.830	0.832	0.851	0.832	0.807
f32	512	16	0.650	0.659	0.677	0.673	0.668	0.705	0.673	0.668
f32	512	12	0.538	0.545	0.521	0.512	0.503	0.559	0.509	0.531
f32	256	24	0.806	0.821	1.000	0.989	0.998	1.005	1.001	0.911
f32	256	20	0.697	0.701	0.840	0.827	0.834	0.861	0.838	0.773
f32	256	16	0.574	0.578	0.674	0.667	0.668	0.702	0.669	0.629
f32	256	12	0.459	0.469	0.513	0.508	0.503	0.556	0.511	0.492
f32	128	24	0.743	0.774	0.998	0.963	0.973	0.970	0.972	0.878
f32	128	20	0.635	0.671	0.832	0.808	0.813	0.830	0.817	0.744
f32	128	16	0.517	0.544	0.672	0.653	0.652	0.688	0.656	0.604
f32	128	12	0.420	0.435	0.513	0.499	0.492	0.556	0.499	0.473
f32	64	24	0.733	0.760	0.996	0.972	0.976	0.980	0.981	0.874
f32	64	20	0.628	0.636	0.836	0.816	0.816	0.842	0.824	0.737
f32	64	16	0.511	0.536	0.675	0.659	0.654	0.703	0.663	0.603
f32	64	12	0.406	0.431	0.514	0.502	0.494	0.557	0.502	0.470
f32	32	24	0.801	0.837	1.084	0.971	0.975	0.983	0.978	0.931
f32	32	20	0.680	0.697	0.905	0.813	0.816	0.842	0.820	0.781
f32	32	16	0.558	0.577	0.732	0.659	0.655	0.702	0.662	0.638
f32	32	12	0.428	0.444	0.554	0.502	0.494	0.558	0.501	0.489
f32	16	24	0.784	0.809	1.087	0.959	0.983	0.965	0.974	0.919
f32	16	20	0.672	0.692	0.844	0.805	0.820	0.826	0.818	0.761
f32	16	16	0.512	0.549	0.672	0.643	0.660	0.675	0.648	0.602
f32	16	12	0.400	0.424	0.512	0.494	0.498	0.539	0.495	0.465
f32	8	24	0.742	0.774	1.014	0.959	0.982	0.961	0.977	0.882
f32	8	20	0.632	0.644	0.851	0.868	0.859	0.902	0.883	0.757
f32	8	16	0.516	0.526	0.680	0.700	0.707	0.745	0.713	0.615
f32	8	12	0.404	0.409	0.515	0.529	0.536	0.594	0.533	0.473
int	1024	24	1.004	1.010	1.004	1.005	0.997	1.002	1.000	1.005
int	1024	20	0.893	0.890	0.841	0.842	0.833	0.859	0.839	0.866
int	1024	16	0.771	0.770	0.682	0.687	0.669	0.716	0.674	0.726
int	1024	12	0.660	0.667	0.521	0.529	0.504	0.564	0.512	0.591
int	512	24	0.884	0.898	1.004	0.992	0.996	0.995	0.994	0.948
int	512	20	0.767	0.802	0.840	0.832	0.832	0.850	0.833	0.814
int	512	16	0.651	0.650	0.678	0.673	0.669	0.705	0.673	0.666
int	512	12	0.540	0.563	0.517	0.512	0.503	0.559	0.508	0.535
int	256	24	0.800	0.818	0.998	0.988	0.996	1.012	1.000	0.909
int	256	20	0.686	0.708	0.835	0.827	0.832	0.853	0.834	0.771
int	256	16	0.581	0.580	0.673	0.669	0.669	0.703	0.670	0.631
int	256	12	0.469	0.469	0.512	0.509	0.503	0.561	0.510	0.494
int	128	24	0.751	0.772	0.998	0.966	0.974	0.978	0.974	0.880
int	128	20	0.647	0.668	0.835	0.815	0.815	0.842	0.820	0.748
int	128	16	0.530	0.549	0.674	0.659	0.654	0.701	0.662	0.610
int	128	12	0.427	0.447	0.513	0.503	0.493	0.556	0.501	0.478
int	64	24	0.750	0.754	0.996	0.973	0.976	0.987	0.979	0.876
int	64	20	0.637	0.654	0.836	0.815	0.815	0.844	0.824	0.743
int	64	16	0.534	0.530	0.674	0.660	0.654	0.701	0.657	0.605
int	64	12	0.420	0.415	0.513	0.503	0.494	0.558	0.500	0.469
int	32	24	0.808	0.827	1.088	0.970	0.975	0.982	0.976	0.931
int	32	20	0.686	0.686	0.907	0.815	0.815	0.841	0.822	0.780
int	32	16	0.566	0.565	0.730	0.658	0.654	0.699	0.660	0.636
int	32	12	0.443	0.442	0.556	0.502	0.494	0.556	0.501	0.491
int	16	24	0.746	0.770	1.000	0.931	0.981	0.941	0.943	0.873
int	16	20	0.627	0.658	0.838	0.807	0.814	0.826	0.816	0.740
int	16	16	0.525	0.554	0.682	0.647	0.658	0.685	0.660	0.610
int	16	12	0.411	0.444	0.514	0.482	0.498	0.541	0.494	0.472
int	8	24	0.751	0.778	1.013	0.989	1.050	1.036	1.060	0.900
int	8	20	0.636	0.665	0.848	0.832	0.871	0.870	0.892	0.759
int	8	16	0.522	0.547	0.684	0.674	0.699	0.711	0.683	0.616
int	8	12	0.414	0.429	0.519	0.530	0.536	0.589	0.551	0.482
bin	1024	24	0.755	0.792	0.998	0.985	0.996	0.991	0.997	0.891
bin	1024	20	0.645	0.670	0.835	0.827	0.832	0.847	0.835	0.752
bin	1024	16	0.529	0.554	0.676	0.667	0.668	0.703	0.668	0.614
bin	1024	12	0.428	0.445	0.518	0.507	0.503	0.554	0.509	0.480
bin	512	24	0.746	0.780	0.996	0.984	0.996	0.984	0.995	0.884
bin	512	20	0.637	0.647	0.836	0.829	0.833	0.843	0.834	0.744
bin	512	16	0.518	0.540	0.677	0.669	0.669	0.710	0.672	0.609
bin	512	12	0.416	0.419	0.515	0.508	0.504	0.565	0.511	0.471
bin	256	24	0.733	0.760	0.998	0.986	0.998	0.990	1.001	0.878
bin	256	20	0.620	0.637	0.834	0.809	0.826	0.827	0.815	0.733
bin	256	16	0.516	0.522	0.671	0.649	0.653	0.684	0.654	0.597
bin	256	12	0.400	0.406	0.512	0.493	0.492	0.540	0.497	0.459
bin	128	24	0.726	0.749	0.999	0.968	0.975	0.978	0.981	0.869
bin	128	20	0.615	0.637	0.832	0.811	0.815	0.835	0.820	0.732
bin	128	16	0.510	0.517	0.673	0.658	0.655	0.694	0.660	0.596
bin	128	12	0.388	0.400	0.514	0.502	0.494	0.552	0.501	0.458
bin	64	24	0.725	0.742	0.996	0.971	0.976	0.979	0.980	0.867
bin	64	20	0.614	0.626	0.833	0.815	0.816	0.838	0.821	0.730
bin	64	16	0.506	0.514	0.671	0.659	0.655	0.697	0.660	0.595
bin	64	12	0.424	0.424	0.551	0.502	0.494	0.551	0.499	0.481
bin	32	24	0.787	0.786	1.085	0.968	0.975	0.975	0.977	0.914
bin	32	20	0.663	0.657	0.910	0.809	0.816	0.834	0.818	0.768
bin	32	16	0.535	0.540	0.730	0.652	0.654	0.685	0.655	0.621
bin	32	12	0.416	0.413	0.554	0.497	0.498	0.539	0.495	0.476
bin	16	24	0.740	0.755	1.013	0.959	0.980	0.958	0.979	0.875
bin	16	20	0.626	0.634	0.850	0.804	0.820	0.821	0.816	0.737
bin	16	16	0.514	0.526	0.683	0.648	0.659	0.676	0.659	0.600
bin	16	12	0.398	0.399	0.518	0.493	0.497	0.536	0.495	0.459
bin	8	24	0.737	0.755	1.017	1.027	1.048	1.040	1.036	0.894
bin	8	20	0.626	0.634	0.848	0.857	0.877	0.883	0.866	0.751
bin	8	16	0.507	0.519	0.682	0.703	0.705	0.731	0.697	0.610
bin	8	12	0.394	0.401	0.515	0.532	0.533	0.583	0.513	0.468

Table 38: Relative evaluation time for mxbai-embed-2d-large-v1, averaged per category. Table 2 serves as a legend.

q	m	l	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	1024	24	32768	0.866	0.492	0.592	0.747	0.740	0.781	0.931	0.733	0.605	0.603	0.420	0.397	0.372	0.356	0.546	0.575	0.400	0.511	0.875	0.810	0.796	0.896	0.850	0.881	0.867	0.887	0.886	0.697	0.961	0.761	0.855	0.630	0.379	0.370	0.638	0.319	0.532	0.316
f32	1024	20	32768	0.811	0.407	0.586	0.680	0.692	0.745	0.871	0.706	0.605	0.595	0.328	0.316	0.342	0.335	0.416	0.542	0.383	0.438	0.782	0.736	0.708	0.769	0.714	0.792	0.808	0.804	0.815	0.630	0.758	0.669	0.805	0.383	0.198	0.151	0.577	0.300	0.453	0.300
f32	1024	16	32768	0.785	0.396	0.517	0.592	0.631	0.717	0.846	0.604	0.605	0.583	0.380	0.353	0.372	0.363	0.459	0.539	0.400	0.477	0.809	0.700	0.680	0.765	0.685	0.746	0.766	0.776	0.829	0.664	0.911	0.587	0.805	0.471	0.240	0.285	0.572	0.317	0.452	0.310
f32	1024	12	32768	0.754	0.347	0.488	0.593	0.613	0.704	0.844	0.589	0.610	0.586	0.394	0.348	0.377	0.352	0.468	0.551	0.390	0.471	0.745	0.680	0.690	0.754	0.680	0.766	0.757	0.755	0.801	0.637	0.904	0.563	0.799	0.425	0.190	0.233	0.548	0.323	0.422	0.285
f32	512	24	16384	0.864	0.484	0.585	0.720	0.735	0.773	0.920	0.722	0.603	0.598	0.421	0.392	0.371	0.359	0.536	0.574	0.400	0.502	0.874	0.807	0.796	0.897	0.848	0.878	0.865	0.886	0.879	0.702	0.960	0.759	0.854	0.627	0.371	0.366	0.629	0.318	0.530	0.319
f32	512	20	16384	0.804	0.406	0.580	0.667	0.684	0.737	0.864	0.692	0.593	0.592	0.325	0.313	0.344	0.335	0.412	0.539	0.384	0.441	0.778	0.737	0.711	0.774	0.711	0.785	0.809	0.803	0.817	0.640	0.759	0.673	0.803	0.391	0.185	0.145	0.580	0.302	0.449	0.303
f32	512	16	16384	0.779	0.381	0.512	0.589	0.626	0.706	0.836	0.590	0.599	0.584	0.376	0.345	0.374	0.361	0.455	0.535	0.400	0.464	0.804	0.700	0.681	0.763	0.684	0.737	0.764	0.774	0.827	0.664	0.909	0.588	0.803	0.467	0.236	0.274	0.570	0.316	0.450	0.312
f32	512	12	16384	0.745	0.338	0.488	0.595	0.603	0.693	0.834	0.574	0.611	0.586	0.382	0.346	0.374	0.349	0.464	0.547	0.391	0.464	0.748	0.682	0.693	0.751	0.678	0.761	0.755	0.753	0.796	0.635	0.899	0.562	0.797	0.420	0.184	0.224	0.548	0.321	0.420	0.292
f32	256	24	8192	0.861	0.471	0.577	0.708	0.723	0.766	0.909	0.706	0.602	0.597	0.403	0.385	0.365	0.345	0.514	0.560	0.398	0.486	0.873	0.807	0.796	0.894	0.845	0.874	0.863	0.885	0.877	0.697	0.957	0.751	0.850	0.611	0.351	0.349	0.625	0.309	0.522	0.316
f32	256	20	8192	0.794	0.390	0.574	0.649	0.675	0.727	0.853	0.666	0.595	0.590	0.319	0.311	0.340	0.335	0.406	0.533	0.388	0.441	0.768	0.741	0.707	0.774	0.709	0.778	0.807	0.800	0.817	0.636	0.764	0.669	0.798	0.393	0.170	0.133	0.578	0.302	0.447	0.312
f32	256	16	8192	0.768	0.370	0.506	0.574	0.612	0.692	0.825	0.568	0.609	0.579	0.369	0.345	0.367	0.361	0.448	0.535	0.396	0.458	0.804	0.701	0.676	0.757	0.681	0.727	0.762	0.771	0.824	0.653	0.907	0.587	0.800	0.463	0.226	0.263	0.570	0.317	0.447	0.310
f32	256	12	8192	0.732	0.324	0.487	0.580	0.590	0.678	0.821	0.551	0.606	0.584	0.385	0.340	0.373	0.356	0.451	0.545	0.391	0.461	0.747	0.684	0.677	0.740	0.673	0.747	0.752	0.754	0.802	0.631	0.893	0.560	0.793	0.405	0.182	0.214	0.547	0.322	0.420	0.289
f32	128	24	4096	0.854	0.454	0.563	0.674	0.701	0.741	0.893	0.669	0.585	0.585	0.395	0.375	0.358	0.347	0.494	0.543	0.396	0.478	0.873	0.803	0.791	0.886	0.836	0.869	0.854	0.877	0.863	0.694	0.948	0.742	0.845	0.586	0.318	0.328	0.621	0.300	0.513	0.313
f32	128	20	4096	0.777	0.374	0.562	0.627	0.653	0.709	0.832	0.630	0.591	0.584	0.309	0.306	0.341	0.334	0.395	0.524	0.388	0.443	0.764	0.740	0.718	0.771	0.707	0.774	0.802	0.796	0.804	0.641	0.770	0.656	0.790	0.376	0.143	0.115	0.575	0.305	0.441	0.310
f32	128	16	4096	0.755	0.355	0.493	0.572	0.594	0.675	0.806	0.546	0.592	0.573	0.372	0.339	0.367	0.359	0.434	0.526	0.398	0.452	0.800	0.696	0.677	0.748	0.672	0.717	0.756	0.764	0.815	0.651	0.898	0.582	0.793	0.454	0.201	0.246	0.567	0.315	0.440	0.316
f32	128	12	4096	0.709	0.319	0.476	0.566	0.581	0.669	0.806	0.523	0.610	0.570	0.380	0.337	0.372	0.356	0.442	0.536	0.392	0.455	0.768	0.682	0.675	0.733	0.665	0.745	0.747	0.750	0.792	0.630	0.882	0.556	0.786	0.392	0.155	0.193	0.545	0.318	0.415	0.307
f32	64	24	2048	0.827	0.420	0.536	0.647	0.666	0.702	0.859	0.614	0.571	0.572	0.367	0.349	0.353	0.337	0.439	0.509	0.394	0.439	0.850	0.796	0.786	0.869	0.828	0.858	0.845	0.864	0.828	0.686	0.927	0.726	0.832	0.549	0.271	0.274	0.606	0.285	0.497	0.306
f32	64	20	2048	0.745	0.361	0.534	0.594	0.611	0.677	0.803	0.574	0.587	0.582	0.299	0.296	0.340	0.332	0.385	0.515	0.392	0.431	0.759	0.744	0.721	0.765	0.702	0.761	0.794	0.790	0.793	0.646	0.762	0.636	0.779	0.347	0.117	0.093	0.567	0.305	0.435	0.321
f32	64	16	2048	0.727	0.341	0.485	0.557	0.571	0.651	0.776	0.501	0.584	0.581	0.363	0.335	0.367	0.359	0.424	0.519	0.399	0.448	0.781	0.694	0.675	0.736	0.665	0.701	0.748	0.755	0.800	0.650	0.892	0.575	0.783	0.427	0.180	0.226	0.566	0.313	0.430	0.313
f32	64	12	2048	0.678	0.302	0.480	0.553	0.556	0.651	0.777	0.477	0.589	0.575	0.367	0.326	0.370	0.350	0.423	0.529	0.394	0.436	0.750	0.680	0.669	0.722	0.659	0.727	0.735	0.739	0.772	0.633	0.869	0.552	0.773	0.366	0.136	0.172	0.539	0.317	0.407	0.307
f32	32	24	1024	0.768	0.387	0.519	0.617	0.591	0.647	0.798	0.503	0.577	0.561	0.322	0.300	0.333	0.323	0.383	0.474	0.394	0.392	0.829	0.779	0.774	0.841	0.799	0.833	0.827	0.837	0.803	0.674	0.891	0.691	0.812	0.461	0.209	0.202	0.592	0.280	0.472	0.300
f32	32	20	1024	0.685	0.333	0.508	0.586	0.552	0.626	0.755	0.496	0.582	0.575	0.275	0.273	0.322	0.321	0.377	0.497	0.393	0.412	0.754	0.729	0.712	0.743	0.682	0.750	0.781	0.778	0.774	0.640	0.679	0.609	0.760	0.305	0.091	0.069	0.551	0.301	0.413	0.316
f32	32	16	1024	0.676	0.313	0.463	0.545	0.527	0.607	0.735	0.437	0.587	0.580	0.327	0.307	0.354	0.348	0.403	0.500	0.396	0.422	0.753	0.685	0.657	0.707	0.640	0.684	0.731	0.736	0.770	0.644	0.862	0.565	0.767	0.370	0.134	0.181	0.552	0.310	0.415	0.327
f32	32	12	1024	0.602	0.287	0.465	0.546	0.517	0.615	0.728	0.421	0.590	0.581	0.339	0.309	0.356	0.338	0.401	0.510	0.394	0.411	0.690	0.673	0.641	0.694	0.636	0.710	0.718	0.721	0.752	0.640	0.829	0.538	0.751	0.312	0.102	0.113	0.532	0.311	0.393	0.299
f32	16	24	512	0.636	0.351	0.493	0.566	0.479	0.537	0.695	0.367	0.537	0.548	0.258	0.244	0.299	0.288	0.307	0.418	0.386	0.330	0.784	0.757	0.753	0.798	0.754	0.807	0.789	0.805	0.781	0.674	0.807	0.616	0.773	0.308	0.091	0.114	0.563	0.279	0.412	0.289
f32	16	20	512	0.587	0.303	0.477	0.563	0.466	0.548	0.694	0.397	0.569	0.568	0.261	0.254	0.309	0.309	0.358	0.471	0.390	0.400	0.700	0.708	0.687	0.715	0.649	0.722	0.759	0.753	0.730	0.636	0.581	0.555	0.731	0.232	0.056	0.048	0.542	0.296	0.387	0.309
f32	16	16	512	0.581	0.283	0.443	0.538	0.463	0.558	0.683	0.359	0.595	0.586	0.303	0.277	0.336	0.328	0.360	0.468	0.391	0.393	0.711	0.674	0.652	0.687	0.625	0.664	0.721	0.714	0.718	0.637	0.807	0.535	0.743	0.285	0.076	0.114	0.527	0.304	0.390	0.304
f32	16	12	512	0.490	0.270	0.449	0.543	0.460	0.557	0.672	0.361	0.581	0.579	0.311	0.271	0.339	0.320	0.367	0.482	0.387	0.387	0.662	0.657	0.639	0.661	0.614	0.674	0.701	0.700	0.694	0.620	0.746	0.511	0.723	0.224	0.049	0.073	0.502	0.291	0.361	0.282
f32	8	24	256	0.425	0.308	0.434	0.562	0.356	0.429	0.579	0.248	0.524	0.521	0.193	0.183	0.261	0.249	0.243	0.352	0.379	0.284	0.711	0.706	0.729	0.711	0.677	0.713	0.746	0.747	0.673	0.668	0.501	0.529	0.717	0.112	0.011	0.037	0.519	0.261	0.341	0.280
f32	8	20	256	0.412	0.284	0.457	0.564	0.354	0.449	0.615	0.306	0.571	0.558	0.222	0.206	0.293	0.283	0.302	0.427	0.385	0.364	0.666	0.678	0.671	0.666	0.609	0.665	0.725	0.718	0.717	0.593	0.352	0.490	0.685	0.110	0.010	0.024	0.516	0.285	0.340	0.273
f32	8	16	256	0.411	0.249	0.398	0.529	0.351	0.470	0.626	0.290	0.583	0.575	0.222	0.209	0.299	0.292	0.304	0.429	0.386	0.337	0.615	0.656	0.626	0.657	0.591	0.609	0.696	0.683	0.670	0.611	0.676	0.480	0.703	0.134	0.025	0.045	0.518	0.290	0.346	0.286
f32	8	12	256	0.353	0.241	0.428	0.552	0.367	0.478	0.619	0.296	0.582	0.577	0.252	0.231	0.314	0.299	0.319	0.445	0.387	0.358	0.491	0.633	0.598	0.628	0.583	0.620	0.672	0.665	0.667	0.604	0.623	0.465	0.685	0.116	0.022	0.039	0.483	0.285	0.322	0.258

Table 39: Absolute performance for mxbai-embed-2d-large-v1 on all tasks using float quantization. Table 2 serves as a legend.

q	m	l	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
int	1024	24	8192	0.859	0.473	0.568	0.649	0.706	0.754	0.928	0.666	0.608	0.602	0.413	0.385	0.374	0.356	0.543	0.574	0.400	0.504	0.869	0.815	0.768	0.896	0.842	0.876	0.855	0.884	0.885	0.683	0.964	0.758	0.855	0.614	0.386	0.374	0.634	0.315	0.535	0.319
int	1024	20	8192	0.807	0.392	0.575	0.599	0.669	0.710	0.865	0.657	0.600	0.581	0.331	0.320	0.344	0.332	0.412	0.541	0.383	0.439	0.755	0.748	0.686	0.762	0.697	0.782	0.804	0.790	0.830	0.610	0.755	0.669	0.812	0.421	0.249	0.229	0.582	0.304	0.448	0.294
int	1024	16	8192	0.779	0.366	0.510	0.535	0.608	0.684	0.841	0.562	0.594	0.585	0.379	0.349	0.369	0.361	0.453	0.538	0.399	0.458	0.808	0.706	0.650	0.772	0.682	0.760	0.756	0.772	0.842	0.638	0.910	0.582	0.806	0.472	0.260	0.293	0.569	0.317	0.452	0.300
int	1024	12	8192	0.751	0.324	0.473	0.543	0.593	0.685	0.851	0.552	0.609	0.577	0.390	0.345	0.371	0.352	0.465	0.552	0.389	0.470	0.727	0.682	0.647	0.765	0.673	0.756	0.746	0.750	0.817	0.596	0.901	0.568	0.801	0.420	0.202	0.243	0.550	0.323	0.423	0.295
int	512	24	4096	0.855	0.468	0.559	0.642	0.699	0.742	0.917	0.640	0.603	0.593	0.421	0.391	0.372	0.358	0.541	0.570	0.399	0.510	0.864	0.813	0.766	0.895	0.839	0.873	0.853	0.883	0.876	0.685	0.963	0.754	0.854	0.616	0.378	0.364	0.630	0.314	0.530	0.318
int	512	20	4096	0.800	0.383	0.565	0.582	0.660	0.709	0.858	0.638	0.597	0.595	0.328	0.314	0.340	0.335	0.409	0.541	0.386	0.433	0.757	0.749	0.691	0.765	0.701	0.783	0.806	0.793	0.831	0.610	0.747	0.667	0.810	0.415	0.240	0.224	0.584	0.304	0.446	0.288
int	512	16	4096	0.773	0.348	0.499	0.527	0.597	0.675	0.835	0.551	0.605	0.587	0.370	0.341	0.371	0.360	0.448	0.533	0.397	0.454	0.808	0.705	0.653	0.769	0.682	0.752	0.758	0.772	0.838	0.638	0.908	0.583	0.805	0.466	0.255	0.290	0.570	0.319	0.448	0.292
int	512	12	4096	0.742	0.301	0.471	0.550	0.577	0.670	0.842	0.532	0.607	0.586	0.390	0.342	0.370	0.351	0.463	0.547	0.388	0.460	0.719	0.684	0.646	0.760	0.674	0.752	0.744	0.748	0.808	0.590	0.896	0.569	0.800	0.413	0.202	0.238	0.551	0.323	0.420	0.286
int	256	24	2048	0.850	0.457	0.555	0.647	0.682	0.730	0.895	0.608	0.610	0.591	0.404	0.388	0.365	0.349	0.520	0.562	0.399	0.480	0.866	0.811	0.763	0.892	0.837	0.867	0.850	0.880	0.872	0.685	0.961	0.747	0.850	0.593	0.357	0.350	0.623	0.312	0.523	0.322
int	256	20	2048	0.785	0.347	0.552	0.588	0.642	0.690	0.840	0.606	0.605	0.595	0.323	0.308	0.338	0.329	0.398	0.534	0.385	0.424	0.730	0.751	0.682	0.766	0.706	0.777	0.803	0.791	0.833	0.603	0.743	0.664	0.805	0.406	0.221	0.216	0.572	0.302	0.443	0.298
int	256	16	2048	0.757	0.346	0.489	0.528	0.580	0.665	0.818	0.513	0.592	0.584	0.361	0.339	0.366	0.358	0.441	0.531	0.398	0.455	0.802	0.706	0.656	0.762	0.680	0.747	0.753	0.770	0.830	0.628	0.908	0.580	0.801	0.448	0.241	0.283	0.566	0.319	0.447	0.294
int	256	12	2048	0.729	0.279	0.465	0.536	0.559	0.654	0.827	0.505	0.602	0.583	0.384	0.337	0.369	0.347	0.452	0.542	0.389	0.463	0.725	0.684	0.641	0.751	0.665	0.743	0.737	0.743	0.804	0.591	0.889	0.567	0.797	0.397	0.188	0.223	0.550	0.320	0.418	0.273
int	128	24	1024	0.832	0.436	0.539	0.619	0.650	0.700	0.877	0.552	0.588	0.579	0.392	0.375	0.355	0.342	0.499	0.544	0.396	0.469	0.879	0.806	0.751	0.882	0.828	0.858	0.842	0.872	0.863	0.683	0.955	0.734	0.844	0.569	0.324	0.338	0.612	0.311	0.515	0.317
int	128	20	1024	0.763	0.331	0.534	0.603	0.601	0.659	0.824	0.536	0.592	0.584	0.307	0.306	0.337	0.330	0.387	0.522	0.385	0.426	0.726	0.750	0.685	0.768	0.700	0.765	0.798	0.786	0.820	0.601	0.744	0.653	0.796	0.394	0.186	0.196	0.571	0.303	0.435	0.285
int	128	16	1024	0.737	0.318	0.468	0.513	0.546	0.628	0.801	0.459	0.577	0.578	0.356	0.332	0.368	0.357	0.429	0.521	0.397	0.437	0.799	0.700	0.645	0.750	0.673	0.732	0.746	0.759	0.826	0.617	0.901	0.578	0.795	0.431	0.206	0.264	0.565	0.318	0.436	0.298
int	128	12	1024	0.701	0.269	0.442	0.511	0.540	0.635	0.809	0.458	0.608	0.578	0.381	0.336	0.372	0.349	0.439	0.537	0.392	0.452	0.719	0.680	0.639	0.733	0.654	0.729	0.730	0.742	0.801	0.587	0.881	0.563	0.790	0.394	0.162	0.208	0.550	0.317	0.409	0.279
int	64	24	512	0.788	0.365	0.488	0.592	0.598	0.648	0.827	0.478	0.566	0.563	0.364	0.340	0.349	0.336	0.439	0.509	0.400	0.442	0.838	0.799	0.736	0.868	0.814	0.839	0.831	0.858	0.826	0.664	0.932	0.722	0.831	0.515	0.282	0.282	0.596	0.305	0.490	0.301
int	64	20	512	0.692	0.289	0.504	0.538	0.539	0.607	0.787	0.449	0.588	0.574	0.298	0.291	0.333	0.337	0.380	0.511	0.391	0.420	0.704	0.755	0.680	0.754	0.690	0.752	0.786	0.775	0.816	0.605	0.738	0.638	0.782	0.363	0.153	0.171	0.564	0.304	0.429	0.302
int	64	16	512	0.679	0.310	0.454	0.481	0.502	0.589	0.747	0.396	0.575	0.568	0.356	0.333	0.361	0.358	0.416	0.514	0.398	0.433	0.767	0.696	0.637	0.734	0.662	0.726	0.732	0.749	0.808	0.622	0.897	0.572	0.786	0.396	0.180	0.251	0.566	0.315	0.425	0.293
int	64	12	512	0.644	0.248	0.423	0.516	0.492	0.593	0.762	0.402	0.602	0.579	0.366	0.328	0.369	0.346	0.420	0.526	0.395	0.425	0.696	0.674	0.634	0.714	0.636	0.708	0.716	0.726	0.779	0.598	0.873	0.557	0.779	0.356	0.147	0.193	0.540	0.317	0.402	0.287
int	32	24	256	0.687	0.328	0.454	0.544	0.485	0.561	0.736	0.353	0.566	0.566	0.318	0.302	0.332	0.321	0.382	0.469	0.396	0.386	0.816	0.784	0.710	0.844	0.785	0.817	0.814	0.827	0.794	0.637	0.901	0.686	0.812	0.422	0.206	0.209	0.578	0.298	0.462	0.293
int	32	20	256	0.577	0.260	0.448	0.533	0.441	0.546	0.725	0.320	0.580	0.574	0.278	0.275	0.319	0.320	0.364	0.497	0.393	0.404	0.704	0.735	0.652	0.704	0.657	0.739	0.764	0.752	0.784	0.587	0.627	0.606	0.762	0.299	0.110	0.120	0.540	0.307	0.401	0.308
int	32	16	256	0.569	0.279	0.418	0.514	0.441	0.530	0.691	0.301	0.582	0.567	0.330	0.309	0.354	0.348	0.398	0.498	0.394	0.420	0.729	0.688	0.599	0.707	0.637	0.706	0.715	0.720	0.777	0.615	0.873	0.562	0.771	0.335	0.143	0.203	0.554	0.316	0.404	0.308
int	32	12	256	0.522	0.233	0.399	0.413	0.413	0.536	0.686	0.309	0.593	0.577	0.343	0.306	0.359	0.341	0.405	0.511	0.395	0.417	0.616	0.664	0.617	0.672	0.612	0.699	0.690	0.687	0.744	0.589	0.836	0.546	0.758	0.292	0.101	0.132	0.528	0.317	0.388	0.281
int	16	24	128	0.462	0.293	0.474	0.507	0.330	0.399	0.562	0.229	0.542	0.550	0.253	0.243	0.300	0.285	0.299	0.413	0.383	0.324	0.814	0.758	0.695	0.788	0.723	0.762	0.754	0.782	0.773	0.630	0.814	0.614	0.772	0.252	0.085	0.122	0.550	0.291	0.409	0.278
int	16	20	128	0.391	0.244	0.450	0.526	0.313	0.412	0.598	0.207	0.576	0.570	0.263	0.255	0.311	0.309	0.350	0.470	0.391	0.394	0.731	0.713	0.622	0.698	0.613	0.688	0.735	0.712	0.732	0.601	0.517	0.557	0.731	0.207	0.052	0.073	0.514	0.294	0.371	0.305
int	16	16	128	0.385	0.242	0.408	0.524	0.332	0.424	0.599	0.233	0.581	0.581	0.304	0.289	0.337	0.329	0.365	0.469	0.390	0.391	0.673	0.680	0.604	0.674	0.604	0.676	0.699	0.693	0.723	0.585	0.825	0.539	0.747	0.233	0.073	0.135	0.531	0.305	0.369	0.292
int	16	12	128	0.336	0.229	0.412	0.491	0.307	0.419	0.571	0.205	0.593	0.581	0.312	0.279	0.346	0.325	0.369	0.486	0.388	0.393	0.647	0.645	0.610	0.625	0.564	0.650	0.640	0.656	0.676	0.560	0.758	0.518	0.731	0.206	0.073	0.073	0.507	0.298	0.346	0.269
int	8	24	64	0.341	0.263	0.429	0.528	0.283	0.321	0.461	0.184	0.519	0.525	0.190	0.177	0.263	0.249	0.239	0.350	0.376	0.283	0.682	0.712	0.619	0.660	0.637	0.688	0.680	0.724	0.667	0.620	0.516	0.540	0.717	0.081	0.012	0.046	0.511	0.263	0.334	0.276
int	8	20	64	0.311	0.276	0.426	0.558	0.276	0.344	0.464	0.174	0.567	0.565	0.223	0.211	0.293	0.279	0.302	0.426	0.386	0.360	0.640	0.675	0.551	0.607	0.546	0.629	0.676	0.642	0.613	0.519	0.331	0.491	0.684	0.112	0.007	0.033	0.511	0.286	0.313	0.281
int	8	16	64	0.297	0.251	0.385	0.543	0.274	0.364	0.477	0.240	0.582	0.573	0.225	0.210	0.299	0.292	0.304	0.428	0.386	0.333	0.595	0.644	0.573	0.604	0.542	0.599	0.657	0.622	0.670	0.572	0.682	0.480	0.703	0.129	0.016	0.050	0.498	0.294	0.303	0.271
int	8	12	64	0.279	0.209	0.410	0.549	0.302	0.360	0.440	0.219	0.578	0.573	0.256	0.232	0.314	0.300	0.318	0.445	0.387	0.350	0.515	0.618	0.564	0.585	0.497	0.572	0.605	0.617	0.628	0.583	0.641	0.465	0.688	0.094	0.022	0.038	0.477	0.278	0.308	0.268

Table 40: Absolute performance for mxbai-embed-2d-large-v1 on all tasks using int quantization. Table 2 serves as a legend.

q	m	l	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
bin	1024	24	1024	0.756	0.366	0.478	0.584	0.577	0.621	0.785	0.488	0.547	0.546	0.339	0.301	0.323	0.313	0.415	0.462	0.363	0.367	0.753	0.771	0.744	0.839	0.790	0.818	0.824	0.843	0.836	0.664	0.918	0.701	0.829	0.455	0.204	0.235	0.588	0.274	0.453	0.299
bin	1024	20	1024	0.622	0.297	0.476	0.595	0.487	0.550	0.728	0.464	0.552	0.536	0.258	0.266	0.304	0.299	0.313	0.449	0.346	0.360	0.637	0.685	0.612	0.686	0.627	0.723	0.752	0.723	0.700	0.583	0.568	0.617	0.768	0.197	0.059	0.071	0.532	0.276	0.384	0.306
bin	1024	16	1024	0.623	0.284	0.436	0.557	0.484	0.587	0.734	0.409	0.575	0.551	0.325	0.307	0.324	0.321	0.370	0.463	0.358	0.398	0.789	0.667	0.644	0.719	0.634	0.699	0.736	0.722	0.796	0.630	0.846	0.563	0.772	0.312	0.118	0.193	0.538	0.306	0.387	0.281
bin	1024	12	1024	0.557	0.254	0.411	0.539	0.442	0.537	0.707	0.380	0.568	0.547	0.346	0.306	0.338	0.312	0.372	0.485	0.353	0.387	0.663	0.643	0.610	0.673	0.596	0.705	0.716	0.694	0.727	0.625	0.803	0.506	0.766	0.236	0.068	0.118	0.506	0.317	0.362	0.284
bin	512	24	512	0.634	0.320	0.457	0.579	0.449	0.501	0.634	0.351	0.524	0.525	0.270	0.246	0.302	0.291	0.298	0.386	0.361	0.298	0.728	0.742	0.720	0.805	0.756	0.768	0.782	0.812	0.786	0.651	0.870	0.657	0.804	0.331	0.119	0.153	0.553	0.270	0.408	0.309
bin	512	20	512	0.480	0.266	0.446	0.599	0.364	0.434	0.623	0.337	0.546	0.539	0.241	0.238	0.281	0.287	0.249	0.397	0.353	0.296	0.570	0.635	0.557	0.656	0.560	0.649	0.717	0.657	0.627	0.561	0.460	0.564	0.729	0.133	0.041	0.049	0.515	0.269	0.347	0.281
bin	512	16	512	0.512	0.265	0.406	0.543	0.383	0.491	0.646	0.306	0.569	0.545	0.284	0.264	0.311	0.299	0.312	0.416	0.360	0.347	0.744	0.620	0.610	0.702	0.586	0.669	0.703	0.686	0.707	0.603	0.785	0.545	0.748	0.234	0.074	0.126	0.520	0.292	0.356	0.302
bin	512	12	512	0.439	0.218	0.381	0.529	0.338	0.430	0.600	0.277	0.546	0.568	0.293	0.259	0.312	0.286	0.307	0.433	0.360	0.327	0.602	0.614	0.553	0.614	0.544	0.662	0.664	0.631	0.652	0.580	0.712	0.482	0.729	0.153	0.047	0.054	0.482	0.301	0.329	0.267
bin	256	24	256	0.440	0.265	0.425	0.554	0.309	0.374	0.489	0.219	0.510	0.499	0.208	0.188	0.279	0.273	0.200	0.301	0.362	0.223	0.702	0.706	0.655	0.742	0.683	0.696	0.726	0.756	0.713	0.637	0.758	0.586	0.759	0.180	0.057	0.092	0.512	0.263	0.355	0.288
bin	256	20	256	0.343	0.229	0.415	0.552	0.249	0.306	0.460	0.212	0.545	0.527	0.201	0.195	0.271	0.270	0.181	0.330	0.359	0.230	0.532	0.582	0.509	0.560	0.477	0.599	0.647	0.582	0.538	0.466	0.315	0.508	0.670	0.068	0.024	0.027	0.492	0.251	0.297	0.269
bin	256	16	256	0.372	0.233	0.397	0.537	0.275	0.362	0.519	0.212	0.551	0.515	0.224	0.207	0.276	0.276	0.236	0.355	0.370	0.260	0.662	0.561	0.546	0.632	0.521	0.609	0.652	0.617	0.656	0.549	0.680	0.477	0.704	0.137	0.043	0.085	0.492	0.280	0.317	0.290
bin	256	12	256	0.316	0.196	0.364	0.507	0.230	0.315	0.453	0.182	0.524	0.533	0.210	0.190	0.279	0.262	0.216	0.350	0.366	0.244	0.572	0.545	0.478	0.522	0.462	0.592	0.585	0.559	0.556	0.537	0.505	0.429	0.669	0.079	0.019	0.044	0.459	0.279	0.287	0.280
bin	128	24	128	0.276	0.265	0.444	0.530	0.188	0.239	0.336	0.139	0.497	0.518	0.164	0.150	0.242	0.242	0.151	0.248	0.361	0.168	0.512	0.609	0.526	0.638	0.569	0.574	0.616	0.629	0.643	0.526	0.518	0.499	0.691	0.083	0.030	0.045	0.483	0.261	0.294	0.286
bin	128	20	128	0.220	0.236	0.433	0.543	0.148	0.187	0.283	0.121	0.512	0.510	0.162	0.155	0.250	0.243	0.137	0.278	0.360	0.191	0.466	0.490	0.433	0.446	0.405	0.506	0.524	0.489	0.425	0.387	0.193	0.447	0.608	0.041	0.009	0.019	0.458	0.248	0.253	0.248
bin	128	16	128	0.251	0.207	0.382	0.532	0.183	0.233	0.351	0.143	0.517	0.516	0.200	0.188	0.264	0.262	0.170	0.293	0.363	0.213	0.606	0.446	0.437	0.547	0.412	0.507	0.589	0.496	0.546	0.435	0.481	0.435	0.647	0.068	0.032	0.051	0.478	0.272	0.281	0.293
bin	128	12	128	0.232	0.200	0.368	0.501	0.159	0.214	0.340	0.130	0.530	0.520	0.191	0.164	0.257	0.251	0.176	0.313	0.366	0.223	0.554	0.460	0.411	0.399	0.368	0.495	0.503	0.451	0.474	0.514	0.351	0.392	0.613	0.053	0.014	0.030	0.440	0.278	0.256	0.279
bin	64	24	64	0.177	0.238	0.408	0.513	0.144	0.211	0.278	0.073	0.508	0.507	0.140	0.134	0.229	0.224	0.140	0.231	0.357	0.151	0.427	0.526	0.449	0.502	0.451	0.471	0.500	0.532	0.537	0.460	0.274	0.424	0.588	0.040	0.015	0.038	0.450	0.257	0.253	0.274
bin	64	20	64	0.174	0.230	0.416	0.521	0.109	0.179	0.272	0.104	0.494	0.504	0.168	0.153	0.252	0.240	0.144	0.272	0.358	0.186	0.408	0.398	0.378	0.386	0.333	0.378	0.456	0.407	0.348	0.294	0.098	0.392	0.525	0.040	0.006	0.013	0.452	0.246	0.236	0.267
bin	64	16	64	0.198	0.199	0.388	0.514	0.134	0.212	0.297	0.097	0.511	0.508	0.176	0.168	0.248	0.249	0.161	0.288	0.362	0.199	0.413	0.412	0.415	0.471	0.359	0.453	0.484	0.451	0.441	0.362	0.272	0.388	0.568	0.054	0.017	0.035	0.444	0.257	0.251	0.277
bin	64	12	64	0.176	0.186	0.369	0.495	0.127	0.196	0.288	0.084	0.546	0.540	0.180	0.157	0.260	0.244	0.181	0.315	0.366	0.214	0.478	0.379	0.367	0.331	0.288	0.382	0.448	0.411	0.425	0.464	0.245	0.334	0.533	0.052	0.011	0.028	0.438	0.264	0.233	0.260
bin	32	24	32	0.098	0.203	0.363	0.516	0.083	0.163	0.216	0.041	0.494	0.514	0.140	0.127	0.231	0.218	0.143	0.249	0.358	0.155	0.296	0.451	0.346	0.348	0.363	0.362	0.421	0.421	0.452	0.223	0.169	0.358	0.508	0.039	0.011	0.032	0.422	0.263	0.230	0.257
bin	32	20	32	0.091	0.181	0.370	0.511	0.068	0.127	0.251	0.053	0.516	0.508	0.154	0.139	0.238	0.233	0.173	0.295	0.360	0.189	0.450	0.373	0.339	0.271	0.270	0.290	0.387	0.356	0.251	0.188	0.067	0.348	0.455	0.023	0.003	0.018	0.415	0.260	0.219	0.240
bin	32	16	32	0.108	0.183	0.375	0.495	0.083	0.156	0.268	0.070	0.513	0.510	0.161	0.150	0.249	0.241	0.171	0.303	0.358	0.187	0.373	0.398	0.385	0.332	0.298	0.362	0.445	0.375	0.373	0.329	0.189	0.341	0.494	0.036	0.016	0.024	0.430	0.264	0.223	0.268
bin	32	12	32	0.105	0.177	0.373	0.486	0.076	0.152	0.240	0.037	0.509	0.520	0.168	0.141	0.230	0.221	0.176	0.318	0.368	0.213	0.399	0.336	0.329	0.259	0.233	0.316	0.369	0.344	0.378	0.301	0.152	0.307	0.501	0.042	0.008	0.027	0.409	0.264	0.210	0.232
bin	16	24	16	0.044	0.203	0.331	0.564	0.039	0.100	0.142	0.019	0.494	0.491	0.132	0.130	0.222	0.211	0.172	0.273	0.356	0.176	0.331	0.391	0.279	0.343	0.301	0.343	0.374	0.391	0.365	0.218	0.110	0.334	0.432	0.016	0.008	0.021	0.413	0.251	0.179	0.228
bin	16	20	16	0.057	0.169	0.358	0.495	0.040	0.108	0.218	0.026	0.489	0.521	0.156	0.136	0.246	0.240	0.218	0.320	0.313	0.214	0.099	0.349	0.319	0.285	0.263	0.272	0.346	0.326	0.196	0.189	0.040	0.314	0.390	0.003	0.002	0.012	0.379	0.216	0.163	0.266
bin	16	16	16	0.054	0.153	0.338	0.494	0.040	0.104	0.181	0.029	0.498	0.519	0.170	0.162	0.269	0.260	0.208	0.326	0.275	0.230	0.195	0.359	0.336	0.308	0.272	0.350	0.465	0.369	0.325	0.314	0.090	0.301	0.414	0.008	0.006	0.021	0.372	0.231	0.146	0.257
bin	16	12	16	0.054	0.158	0.327	0.455	0.040	0.094	0.154	0.018	0.543	0.529	0.173	0.159	0.258	0.253	0.204	0.328	0.292	0.250	0.306	0.318	0.283	0.276	0.232	0.295	0.377	0.349	0.261	0.218	0.051	0.313	0.455	0.011	0.002	0.017	0.339	0.228	0.130	0.243
bin	8	24	8	0.027	0.144	0.329	0.503	0.031	0.099	0.107	0.022	0.488	0.487	0.134	0.134	0.234	0.223	0.190	0.270	0.240	0.208	0.114	0.224	0.155	0.239	0.220	0.156	0.217	0.262	0.238	0.146	0.042	0.272	0.327	0.000	0.001	0.003	0.294	0.104	0.052	0.187
bin	8	20	8	0.031	0.117	0.352	0.459	0.031	0.088	0.181	0.031	0.393	0.440	0.159	0.151	0.244	0.227	0.225	0.299	0.175	0.249	0.086	0.030	-0.058	0.079	0.074	0.016	0.147	0.039	0.045	0.165	0.018	0.294	0.325	0.000	0.000	0.001	0.294	0.101	0.047	0.260
bin	8	16	8	0.035	0.166	0.324	0.520	0.035	0.094	0.146	0.019	0.519	0.541	0.143	0.146	0.266	0.259	0.216	0.320	0.134	0.259	0.188	0.033	-0.011	0.050	0.085	0.097	0.204	0.100	0.098	0.103	0.030	0.281	0.333	0.000	0.000	0.002	0.285	0.107	0.044	0.252
bin	8	12	8	0.029	0.183	0.322	0.515	0.029	0.100	0.152	0.024	0.505	0.477	0.166	0.148	0.213	0.200	0.197	0.296	0.129	0.262	0.165	0.007	0.014	0.046	0.047	-0.034	0.074	0.103	0.069	-0.094	0.015	0.265	0.339	0.000	0.000	0.001	0.275	0.104	0.040	0.193

Table 41: Absolute performance for mxbai-embed-2d-large-v1 on all tasks using binary quantization. Table 2 serves as a legend.

q	m	l	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
f32	1024	24	32768	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	1024	20	32768	0.937	0.828	0.990	0.911	0.934	0.954	0.935	0.963	1.000	0.987	0.782	0.795	0.919	0.939	0.762	0.943	0.958	0.857	0.893	0.909	0.889	0.858	0.840	0.899	0.931	0.907	0.920	0.905	0.789	0.879	0.941	0.609	0.523	0.409	0.904	0.941	0.852	0.950
f32	1024	16	32768	0.906	0.803	0.872	0.792	0.853	0.917	0.909	0.823	1.001	0.967	0.906	0.888	0.999	1.018	0.840	0.939	1.000	0.933	0.925	0.864	0.854	0.853	0.806	0.847	0.883	0.875	0.935	0.953	0.949	0.772	0.942	0.748	0.635	0.771	0.897	0.994	0.851	0.981
f32	1024	12	32768	0.871	0.705	0.824	0.794	0.828	0.901	0.906	0.803	1.008	0.973	0.939	0.876	1.011	0.989	0.856	0.959	0.975	0.923	0.851	0.841	0.866	0.841	0.800	0.870	0.872	0.852	0.904	0.914	0.941	0.741	0.935	0.675	0.501	0.629	0.860	1.015	0.794	0.901
f32	512	24	16384	0.998	0.984	0.988	0.965	0.992	0.989	0.989	0.984	0.997	0.992	1.002	0.988	0.996	1.008	0.982	0.998	1.001	0.983	0.998	0.997	1.000	1.000	0.998	0.997	0.998	0.998	0.993	1.008	1.000	0.997	0.998	0.996	0.981	0.991	0.986	0.999	0.997	1.008
f32	512	20	16384	0.929	0.824	0.979	0.893	0.925	0.943	0.928	0.944	0.981	0.982	0.773	0.787	0.923	0.939	0.755	0.938	0.959	0.863	0.889	0.910	0.893	0.864	0.837	0.891	0.932	0.906	0.923	0.919	0.790	0.884	0.939	0.620	0.489	0.392	0.909	0.949	0.844	0.957
f32	512	16	16384	0.900	0.773	0.864	0.789	0.846	0.903	0.898	0.804	0.990	0.970	0.897	0.868	1.004	1.013	0.834	0.932	1.000	0.908	0.919	0.865	0.855	0.851	0.804	0.837	0.881	0.873	0.933	0.953	0.946	0.772	0.939	0.742	0.623	0.740	0.894	0.992	0.846	0.986
f32	512	12	16384	0.860	0.686	0.824	0.797	0.814	0.886	0.896	0.782	1.010	0.973	0.911	0.871	1.004	0.980	0.850	0.952	0.977	0.908	0.855	0.842	0.870	0.838	0.798	0.864	0.870	0.849	0.899	0.912	0.936	0.739	0.932	0.667	0.486	0.604	0.859	1.006	0.790	0.923
f32	256	24	8192	0.994	0.957	0.974	0.948	0.976	0.980	0.977	0.963	0.996	0.990	0.960	0.968	0.980	0.969	0.941	0.974	0.996	0.952	0.998	0.997	1.000	0.998	0.994	0.993	0.995	0.998	0.990	1.000	0.996	0.987	0.994	0.971	0.927	0.945	0.980	0.970	0.981	1.000
f32	256	20	8192	0.917	0.791	0.970	0.870	0.912	0.930	0.916	0.909	0.983	0.979	0.760	0.782	0.912	0.940	0.744	0.928	0.972	0.862	0.877	0.916	0.888	0.863	0.834	0.883	0.930	0.902	0.922	0.913	0.796	0.879	0.933	0.625	0.449	0.359	0.907	0.949	0.841	0.988
f32	256	16	8192	0.887	0.751	0.854	0.768	0.826	0.886	0.886	0.774	1.007	0.961	0.879	0.870	0.985	1.013	0.819	0.930	0.990	0.897	0.918	0.865	0.849	0.845	0.801	0.826	0.879	0.869	0.931	0.937	0.944	0.772	0.935	0.736	0.597	0.712	0.894	0.996	0.841	0.981
f32	256	12	8192	0.845	0.658	0.823	0.776	0.797	0.868	0.882	0.751	1.002	0.969	0.916	0.856	1.001	0.998	0.826	0.948	0.979	0.903	0.854	0.845	0.850	0.825	0.792	0.849	0.867	0.850	0.906	0.907	0.930	0.736	0.927	0.643	0.482	0.579	0.858	1.011	0.791	0.914
f32	128	24	4096	0.986	0.923	0.950	0.902	0.947	0.948	0.959	0.913	0.967	0.970	0.940	0.944	0.961	0.974	0.905	0.944	0.990	0.936	0.998	0.992	0.993	0.988	0.984	0.987	0.985	0.988	0.974	0.997	0.987	0.975	0.988	0.930	0.839	0.887	0.973	0.943	0.964	0.991
f32	128	20	4096	0.898	0.760	0.949	0.840	0.883	0.907	0.894	0.860	0.977	0.969	0.737	0.769	0.917	0.938	0.723	0.911	0.971	0.866	0.873	0.914	0.902	0.861	0.832	0.879	0.925	0.897	0.908	0.920	0.801	0.863	0.924	0.597	0.378	0.312	0.901	0.956	0.829	0.980
f32	128	16	4096	0.871	0.722	0.833	0.766	0.802	0.863	0.866	0.745	0.980	0.952	0.887	0.853	0.986	1.007	0.795	0.916	0.996	0.885	0.914	0.860	0.850	0.834	0.791	0.814	0.872	0.861	0.920	0.935	0.935	0.765	0.927	0.722	0.532	0.666	0.889	0.987	0.827	0.999
f32	128	12	4096	0.818	0.647	0.803	0.758	0.784	0.857	0.866	0.713	1.009	0.946	0.905	0.847	0.999	0.998	0.809	0.933	0.981	0.889	0.878	0.842	0.848	0.817	0.783	0.846	0.862	0.845	0.894	0.905	0.918	0.732	0.919	0.623	0.410	0.521	0.854	0.998	0.780	0.971
f32	64	24	2048	0.955	0.853	0.906	0.867	0.899	0.898	0.922	0.837	0.945	0.949	0.874	0.878	0.947	0.946	0.803	0.886	0.985	0.860	0.971	0.983	0.988	0.970	0.974	0.975	0.975	0.974	0.935	0.984	0.965	0.955	0.972	0.872	0.715	0.740	0.950	0.893	0.935	0.967
f32	64	20	2048	0.860	0.734	0.901	0.795	0.826	0.866	0.863	0.782	0.971	0.965	0.713	0.746	0.912	0.933	0.704	0.896	0.980	0.844	0.867	0.919	0.905	0.853	0.826	0.864	0.915	0.891	0.895	0.927	0.793	0.836	0.910	0.552	0.309	0.252	0.888	0.958	0.818	1.015
f32	64	16	2048	0.839	0.692	0.820	0.746	0.771	0.833	0.834	0.683	0.966	0.964	0.866	0.842	0.986	1.008	0.776	0.903	0.998	0.876	0.892	0.857	0.848	0.821	0.782	0.796	0.863	0.852	0.904	0.934	0.929	0.755	0.916	0.678	0.474	0.612	0.887	0.982	0.808	0.990
f32	64	12	2048	0.782	0.614	0.810	0.740	0.751	0.833	0.834	0.650	0.973	0.955	0.875	0.820	0.994	0.982	0.774	0.920	0.985	0.853	0.857	0.840	0.840	0.805	0.775	0.826	0.847	0.833	0.872	0.909	0.904	0.726	0.904	0.581	0.358	0.466	0.845	0.995	0.766	0.973
f32	32	24	1024	0.887	0.786	0.876	0.827	0.798	0.827	0.857	0.686	0.955	0.931	0.766	0.756	0.894	0.905	0.701	0.825	0.985	0.767	0.947	0.962	0.973	0.938	0.941	0.946	0.954	0.944	0.906	0.967	0.927	0.908	0.949	0.733	0.551	0.545	0.928	0.880	0.888	0.950
f32	32	20	1024	0.790	0.675	0.858	0.784	0.746	0.800	0.810	0.676	0.963	0.954	0.654	0.686	0.866	0.900	0.690	0.866	0.984	0.807	0.862	0.900	0.895	0.828	0.802	0.852	0.900	0.877	0.874	0.919	0.707	0.800	0.888	0.484	0.240	0.188	0.865	0.946	0.776	0.998
f32	32	16	1024	0.780	0.636	0.782	0.729	0.712	0.776	0.789	0.596	0.971	0.962	0.778	0.773	0.952	0.975	0.737	0.869	0.991	0.826	0.861	0.846	0.825	0.788	0.753	0.777	0.843	0.830	0.869	0.924	0.897	0.742	0.897	0.588	0.355	0.489	0.865	0.973	0.781	1.034
f32	32	12	1024	0.695	0.583	0.784	0.731	0.698	0.787	0.781	0.574	0.976	0.964	0.807	0.778	0.957	0.949	0.735	0.887	0.986	0.804	0.789	0.832	0.806	0.774	0.748	0.806	0.828	0.813	0.849	0.919	0.863	0.707	0.878	0.495	0.268	0.306	0.834	0.978	0.738	0.948
f32	16	24	512	0.734	0.713	0.832	0.758	0.647	0.687	0.747	0.501	0.889	0.910	0.615	0.614	0.803	0.807	0.562	0.727	0.965	0.645	0.896	0.935	0.946	0.891	0.887	0.916	0.910	0.908	0.882	0.968	0.841	0.810	0.904	0.489	0.240	0.308	0.883	0.875	0.775	0.915
f32	16	20	512	0.677	0.616	0.805	0.754	0.629	0.702	0.745	0.541	0.941	0.943	0.621	0.639	0.830	0.868	0.655	0.819	0.975	0.784	0.800	0.874	0.862	0.797	0.764	0.820	0.875	0.849	0.824	0.913	0.605	0.730	0.855	0.369	0.148	0.129	0.850	0.929	0.728	0.977
f32	16	16	512	0.671	0.575	0.748	0.720	0.625	0.714	0.734	0.490	0.984	0.972	0.722	0.697	0.902	0.920	0.660	0.814	0.978	0.769	0.812	0.833	0.819	0.766	0.736	0.753	0.831	0.805	0.811	0.915	0.840	0.703	0.868	0.453	0.200	0.309	0.826	0.955	0.734	0.962
f32	16	12	512	0.566	0.548	0.757	0.727	0.622	0.713	0.722	0.492	0.961	0.961	0.740	0.682	0.911	0.898	0.671	0.838	0.968	0.756	0.756	0.812	0.803	0.738	0.723	0.765	0.808	0.789	0.784	0.889	0.777	0.672	0.846	0.356	0.129	0.197	0.787	0.915	0.678	0.891
f32	8	24	256	0.491	0.626	0.732	0.752	0.481	0.549	0.622	0.339	0.867	0.865	0.460	0.461	0.701	0.700	0.444	0.613	0.947	0.555	0.812	0.871	0.915	0.793	0.796	0.809	0.860	0.842	0.760	0.960	0.522	0.696	0.838	0.179	0.029	0.100	0.814	0.818	0.641	0.886
f32	8	20	256	0.476	0.577	0.771	0.755	0.478	0.574	0.660	0.417	0.945	0.927	0.529	0.518	0.786	0.793	0.553	0.744	0.963	0.713	0.760	0.837	0.843	0.743	0.716	0.755	0.835	0.810	0.809	0.852	0.367	0.644	0.801	0.175	0.027	0.065	0.810	0.896	0.639	0.865
f32	8	16	256	0.474	0.506	0.672	0.709	0.474	0.601	0.673	0.396	0.965	0.954	0.528	0.527	0.803	0.820	0.556	0.746	0.966	0.660	0.703	0.810	0.787	0.734	0.695	0.692	0.802	0.770	0.757	0.877	0.704	0.631	0.822	0.213	0.065	0.123	0.812	0.911	0.650	0.905
f32	8	12	256	0.407	0.490	0.722	0.740	0.496	0.612	0.665	0.404	0.962	0.957	0.602	0.580	0.844	0.839	0.584	0.774	0.968	0.701	0.561	0.782	0.751	0.701	0.686	0.704	0.775	0.749	0.753	0.867	0.649	0.611	0.801	0.184	0.059	0.106	0.757	0.895	0.605	0.817

Table 42: Relative performance for mxbai-embed-2d-large-v1 on all tasks using float quantization. Table 2 serves as a legend.

q	m	l	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
int	1024	24	8192	0.991	0.961	0.958	0.869	0.954	0.964	0.997	0.908	1.006	0.999	0.983	0.968	1.005	0.999	0.994	0.999	1.000	0.987	0.993	1.006	0.965	1.000	0.990	0.995	0.986	0.996	0.999	0.981	1.003	0.997	1.000	0.975	1.019	1.010	0.994	0.990	1.006	1.008
int	1024	20	8192	0.932	0.797	0.970	0.802	0.903	0.908	0.929	0.896	0.993	0.965	0.789	0.806	0.923	0.931	0.755	0.942	0.957	0.859	0.863	0.924	0.862	0.850	0.820	0.888	0.927	0.891	0.937	0.876	0.786	0.879	0.949	0.669	0.658	0.618	0.913	0.954	0.843	0.931
int	1024	16	8192	0.900	0.744	0.861	0.717	0.821	0.876	0.903	0.767	0.983	0.971	0.903	0.880	0.992	1.012	0.829	0.936	0.997	0.896	0.923	0.871	0.816	0.861	0.802	0.863	0.872	0.871	0.950	0.916	0.948	0.765	0.942	0.750	0.688	0.792	0.892	0.996	0.850	0.950
int	1024	12	8192	0.868	0.657	0.799	0.727	0.801	0.877	0.914	0.753	1.007	0.958	0.929	0.868	0.996	0.987	0.851	0.960	0.972	0.920	0.830	0.843	0.812	0.854	0.792	0.859	0.861	0.845	0.922	0.856	0.938	0.746	0.937	0.667	0.533	0.657	0.863	1.015	0.796	0.933
int	512	24	4096	0.987	0.951	0.943	0.860	0.944	0.949	0.985	0.872	0.997	0.984	1.004	0.985	1.000	1.003	0.991	0.993	0.997	0.998	0.988	1.004	0.963	0.998	0.987	0.991	0.983	0.995	0.989	0.983	1.002	0.992	0.998	0.978	0.999	0.985	0.987	0.987	0.997	1.007
int	512	20	4096	0.923	0.779	0.954	0.779	0.891	0.907	0.921	0.870	0.988	0.988	0.781	0.791	0.914	0.941	0.749	0.942	0.964	0.848	0.865	0.925	0.868	0.853	0.824	0.889	0.930	0.894	0.938	0.875	0.778	0.877	0.947	0.659	0.634	0.607	0.915	0.955	0.839	0.911
int	512	16	4096	0.892	0.708	0.843	0.706	0.807	0.864	0.897	0.752	1.000	0.974	0.882	0.857	0.997	1.010	0.821	0.928	0.994	0.889	0.923	0.871	0.821	0.858	0.802	0.854	0.874	0.870	0.946	0.915	0.946	0.766	0.941	0.740	0.674	0.783	0.894	1.002	0.842	0.924
int	512	12	4096	0.857	0.610	0.795	0.737	0.779	0.858	0.904	0.726	1.004	0.972	0.930	0.861	0.995	0.986	0.848	0.952	0.970	0.901	0.821	0.845	0.811	0.847	0.793	0.854	0.858	0.843	0.912	0.847	0.933	0.748	0.935	0.656	0.534	0.644	0.864	1.013	0.789	0.905
int	256	24	2048	0.981	0.927	0.937	0.866	0.921	0.934	0.961	0.829	1.009	0.981	0.962	0.977	0.981	0.979	0.951	0.978	0.997	0.940	0.989	1.002	0.958	0.995	0.984	0.984	0.980	0.992	0.985	0.983	1.000	0.982	0.993	0.943	0.944	0.946	0.977	0.980	0.983	1.018
int	256	20	2048	0.906	0.705	0.931	0.787	0.867	0.883	0.903	0.827	1.000	0.988	0.770	0.775	0.909	0.924	0.728	0.929	0.962	0.830	0.835	0.927	0.857	0.855	0.831	0.882	0.926	0.892	0.941	0.865	0.774	0.873	0.941	0.644	0.584	0.584	0.896	0.948	0.833	0.944
int	256	16	2048	0.875	0.703	0.826	0.707	0.783	0.851	0.879	0.700	0.979	0.968	0.861	0.854	0.984	1.004	0.808	0.924	0.996	0.890	0.917	0.872	0.824	0.850	0.800	0.848	0.869	0.868	0.938	0.901	0.945	0.762	0.937	0.711	0.637	0.765	0.888	1.000	0.841	0.932
int	256	12	2048	0.841	0.567	0.784	0.717	0.755	0.837	0.888	0.689	0.995	0.967	0.915	0.847	0.990	0.975	0.828	0.943	0.972	0.907	0.829	0.845	0.805	0.838	0.782	0.843	0.850	0.838	0.907	0.849	0.925	0.745	0.932	0.631	0.496	0.603	0.862	1.004	0.786	0.862
int	128	24	1024	0.961	0.885	0.910	0.829	0.877	0.895	0.942	0.752	0.973	0.961	0.934	0.945	0.953	0.959	0.914	0.947	0.992	0.918	1.005	0.996	0.944	0.984	0.974	0.974	0.971	0.983	0.974	0.981	0.994	0.965	0.987	0.904	0.855	0.914	0.960	0.976	0.968	1.002
int	128	20	1024	0.881	0.672	0.902	0.807	0.812	0.843	0.885	0.731	0.978	0.968	0.732	0.771	0.905	0.925	0.708	0.909	0.964	0.834	0.830	0.926	0.860	0.857	0.824	0.869	0.921	0.886	0.926	0.863	0.774	0.858	0.931	0.626	0.492	0.530	0.895	0.951	0.819	0.902
int	128	16	1024	0.850	0.646	0.791	0.688	0.737	0.803	0.860	0.625	0.955	0.960	0.848	0.836	0.987	1.001	0.786	0.907	0.994	0.856	0.913	0.865	0.810	0.837	0.792	0.831	0.860	0.856	0.932	0.886	0.938	0.759	0.930	0.684	0.545	0.714	0.885	0.997	0.820	0.944
int	128	12	1024	0.809	0.545	0.746	0.684	0.730	0.812	0.869	0.624	1.006	0.960	0.907	0.845	0.999	0.980	0.804	0.935	0.982	0.885	0.822	0.840	0.802	0.817	0.769	0.827	0.841	0.837	0.905	0.842	0.917	0.740	0.924	0.626	0.427	0.562	0.862	0.994	0.769	0.883
int	64	24	512	0.910	0.742	0.824	0.793	0.808	0.829	0.888	0.651	0.936	0.935	0.868	0.855	0.938	0.943	0.803	0.886	1.000	0.865	0.958	0.987	0.925	0.968	0.958	0.953	0.958	0.967	0.933	0.954	0.971	0.949	0.972	0.818	0.746	0.763	0.935	0.958	0.921	0.952
int	64	20	512	0.799	0.587	0.851	0.720	0.728	0.777	0.846	0.613	0.972	0.952	0.709	0.732	0.893	0.945	0.695	0.889	0.979	0.823	0.805	0.932	0.855	0.842	0.811	0.855	0.907	0.874	0.921	0.868	0.769	0.839	0.914	0.577	0.403	0.463	0.885	0.953	0.807	0.956
int	64	16	512	0.784	0.630	0.767	0.644	0.678	0.753	0.803	0.539	0.951	0.943	0.848	0.837	0.970	1.006	0.762	0.894	0.996	0.848	0.877	0.860	0.800	0.819	0.779	0.825	0.844	0.845	0.912	0.894	0.934	0.751	0.919	0.629	0.475	0.678	0.887	0.990	0.799	0.926
int	64	12	512	0.744	0.503	0.715	0.691	0.664	0.759	0.819	0.548	0.996	0.960	0.873	0.826	0.991	0.972	0.768	0.915	0.987	0.832	0.795	0.832	0.797	0.797	0.749	0.804	0.826	0.818	0.880	0.858	0.909	0.732	0.911	0.565	0.389	0.521	0.846	0.994	0.757	0.908
int	32	24	256	0.793	0.665	0.767	0.728	0.655	0.718	0.791	0.481	0.936	0.939	0.756	0.759	0.892	0.900	0.699	0.816	0.989	0.755	0.932	0.969	0.891	0.942	0.923	0.928	0.938	0.932	0.896	0.915	0.938	0.901	0.949	0.671	0.544	0.565	0.906	0.935	0.868	0.928
int	32	20	256	0.666	0.529	0.755	0.714	0.596	0.699	0.779	0.436	0.959	0.953	0.662	0.693	0.857	0.897	0.667	0.865	0.983	0.790	0.804	0.908	0.819	0.785	0.773	0.839	0.881	0.848	0.885	0.843	0.653	0.796	0.891	0.475	0.290	0.325	0.848	0.965	0.754	0.976
int	32	16	256	0.657	0.566	0.705	0.688	0.596	0.679	0.742	0.411	0.963	0.941	0.786	0.777	0.952	0.977	0.728	0.867	0.985	0.821	0.833	0.849	0.752	0.788	0.749	0.802	0.825	0.812	0.877	0.884	0.909	0.739	0.901	0.532	0.379	0.548	0.869	0.993	0.759	0.974
int	32	12	256	0.603	0.473	0.674	0.553	0.558	0.686	0.736	0.421	0.981	0.958	0.817	0.771	0.963	0.956	0.741	0.888	0.989	0.817	0.704	0.820	0.774	0.750	0.720	0.794	0.796	0.774	0.840	0.845	0.870	0.718	0.886	0.464	0.268	0.357	0.828	0.996	0.730	0.889
int	16	24	128	0.534	0.595	0.800	0.679	0.445	0.511	0.603	0.312	0.897	0.914	0.604	0.611	0.804	0.800	0.548	0.718	0.958	0.633	0.930	0.936	0.872	0.879	0.850	0.866	0.870	0.881	0.873	0.905	0.848	0.807	0.903	0.400	0.225	0.330	0.862	0.913	0.769	0.878
int	16	20	128	0.451	0.496	0.759	0.705	0.423	0.528	0.642	0.282	0.952	0.946	0.627	0.641	0.835	0.868	0.641	0.817	0.977	0.771	0.835	0.880	0.780	0.779	0.721	0.781	0.847	0.802	0.827	0.863	0.538	0.732	0.855	0.329	0.137	0.198	0.806	0.924	0.697	0.967
int	16	16	128	0.444	0.492	0.689	0.702	0.448	0.542	0.644	0.318	0.961	0.965	0.725	0.727	0.905	0.924	0.668	0.815	0.976	0.764	0.769	0.840	0.758	0.752	0.711	0.768	0.806	0.781	0.816	0.839	0.859	0.709	0.874	0.370	0.194	0.365	0.832	0.957	0.695	0.923
int	16	12	128	0.388	0.465	0.695	0.658	0.415	0.536	0.614	0.280	0.981	0.964	0.744	0.703	0.930	0.910	0.675	0.846	0.971	0.769	0.740	0.797	0.766	0.698	0.663	0.738	0.738	0.740	0.763	0.804	0.789	0.680	0.854	0.328	0.193	0.198	0.795	0.936	0.651	0.852
int	8	24	64	0.393	0.534	0.724	0.707	0.383	0.410	0.495	0.251	0.859	0.872	0.453	0.445	0.706	0.699	0.438	0.608	0.941	0.554	0.779	0.880	0.777	0.736	0.750	0.781	0.784	0.817	0.753	0.890	0.537	0.709	0.838	0.129	0.031	0.123	0.801	0.825	0.627	0.875
int	8	20	64	0.359	0.561	0.719	0.747	0.373	0.440	0.498	0.237	0.938	0.937	0.532	0.530	0.786	0.782	0.553	0.741	0.966	0.704	0.732	0.834	0.692	0.677	0.642	0.715	0.779	0.723	0.692	0.746	0.345	0.646	0.800	0.178	0.019	0.090	0.801	0.897	0.589	0.890
int	8	16	64	0.343	0.510	0.650	0.727	0.369	0.465	0.512	0.327	0.962	0.950	0.535	0.529	0.802	0.820	0.557	0.745	0.965	0.651	0.680	0.795	0.720	0.674	0.637	0.680	0.757	0.701	0.757	0.822	0.710	0.631	0.822	0.205	0.043	0.135	0.780	0.922	0.570	0.856
int	8	12	64	0.322	0.425	0.692	0.735	0.409	0.461	0.473	0.298	0.956	0.950	0.610	0.583	0.843	0.843	0.582	0.775	0.969	0.685	0.588	0.764	0.708	0.652	0.585	0.650	0.698	0.696	0.710	0.837	0.667	0.612	0.804	0.149	0.058	0.103	0.748	0.871	0.580	0.848

Table 43: Relative performance for mxbai-embed-2d-large-v1 on all tasks using int quantization. Table 2 serves as a legend.

q	m	l	req. bits	CLA 1	CLA 2	CLA 3	CLA 4	CLA 5	CLA 6	CLA 7	CLA 8	CLU 1	CLU 2	CLU 3	CLU 4	CLU 5	CLU 6	CLU 7	CLU 8	CLU 9	CLU 10	STS 1	STS 2	STS 3	STS 4	STS 5	STS 6	STS 7	STS 8	STS 9	STS 10	PCL 1	PCL 2	PCL 3	RET 1	RET 2	RET 3	RER 1	RER 2	RER 3	SUM 1
bin	1024	24	1024	0.872	0.744	0.808	0.781	0.779	0.795	0.843	0.666	0.905	0.907	0.807	0.757	0.867	0.878	0.760	0.804	0.909	0.719	0.861	0.952	0.935	0.936	0.929	0.929	0.950	0.950	0.944	0.953	0.956	0.921	0.969	0.723	0.539	0.635	0.923	0.861	0.852	0.946
bin	1024	20	1024	0.718	0.604	0.804	0.797	0.657	0.704	0.782	0.632	0.913	0.890	0.615	0.669	0.815	0.840	0.574	0.781	0.866	0.705	0.728	0.846	0.768	0.765	0.738	0.821	0.868	0.816	0.790	0.837	0.591	0.810	0.898	0.313	0.154	0.193	0.835	0.867	0.721	0.968
bin	1024	16	1024	0.720	0.578	0.736	0.746	0.654	0.751	0.789	0.558	0.951	0.914	0.773	0.773	0.870	0.900	0.678	0.806	0.895	0.778	0.902	0.824	0.809	0.802	0.745	0.794	0.848	0.814	0.899	0.905	0.880	0.740	0.903	0.496	0.311	0.521	0.843	0.962	0.727	0.889
bin	1024	12	1024	0.643	0.516	0.694	0.721	0.597	0.687	0.760	0.518	0.940	0.908	0.823	0.770	0.907	0.875	0.682	0.843	0.882	0.757	0.758	0.794	0.766	0.751	0.701	0.801	0.826	0.782	0.821	0.897	0.836	0.665	0.896	0.374	0.180	0.319	0.793	0.994	0.681	0.897
bin	512	24	512	0.732	0.651	0.771	0.775	0.607	0.642	0.681	0.479	0.866	0.871	0.644	0.619	0.810	0.816	0.545	0.671	0.902	0.583	0.832	0.917	0.905	0.898	0.890	0.873	0.902	0.916	0.888	0.934	0.906	0.863	0.940	0.526	0.315	0.413	0.867	0.846	0.767	0.979
bin	512	20	512	0.554	0.540	0.753	0.802	0.492	0.555	0.669	0.459	0.903	0.894	0.573	0.600	0.755	0.807	0.457	0.691	0.882	0.579	0.652	0.784	0.700	0.732	0.659	0.737	0.827	0.741	0.708	0.805	0.478	0.741	0.852	0.211	0.108	0.133	0.808	0.846	0.653	0.888
bin	512	16	512	0.591	0.538	0.686	0.727	0.518	0.628	0.693	0.418	0.941	0.905	0.678	0.665	0.835	0.839	0.571	0.723	0.902	0.679	0.850	0.766	0.767	0.783	0.689	0.760	0.810	0.773	0.798	0.865	0.817	0.716	0.875	0.372	0.196	0.342	0.815	0.915	0.670	0.954
bin	512	12	512	0.507	0.443	0.643	0.709	0.456	0.550	0.645	0.378	0.903	0.942	0.699	0.651	0.837	0.803	0.563	0.753	0.900	0.639	0.688	0.759	0.695	0.685	0.639	0.752	0.766	0.712	0.737	0.833	0.741	0.634	0.853	0.244	0.125	0.147	0.755	0.945	0.620	0.846
bin	256	24	256	0.508	0.538	0.717	0.742	0.417	0.479	0.525	0.299	0.843	0.829	0.495	0.473	0.748	0.766	0.366	0.524	0.907	0.437	0.802	0.872	0.822	0.828	0.804	0.790	0.837	0.852	0.805	0.914	0.789	0.770	0.888	0.286	0.152	0.249	0.803	0.826	0.668	0.912
bin	256	20	256	0.396	0.465	0.701	0.739	0.336	0.391	0.494	0.289	0.902	0.875	0.480	0.491	0.728	0.757	0.332	0.574	0.899	0.450	0.608	0.719	0.640	0.625	0.561	0.680	0.747	0.655	0.608	0.669	0.328	0.668	0.783	0.108	0.062	0.073	0.772	0.787	0.559	0.852
bin	256	16	256	0.430	0.472	0.671	0.719	0.371	0.463	0.557	0.289	0.911	0.856	0.534	0.520	0.741	0.773	0.432	0.617	0.925	0.509	0.757	0.693	0.686	0.705	0.613	0.691	0.752	0.695	0.741	0.788	0.708	0.627	0.822	0.217	0.113	0.229	0.771	0.879	0.597	0.917
bin	256	12	256	0.364	0.399	0.614	0.679	0.311	0.404	0.486	0.249	0.868	0.884	0.501	0.477	0.748	0.736	0.396	0.609	0.916	0.478	0.653	0.673	0.600	0.582	0.543	0.672	0.674	0.630	0.627	0.771	0.526	0.564	0.782	0.125	0.051	0.118	0.720	0.875	0.540	0.886
bin	128	24	128	0.318	0.539	0.750	0.710	0.254	0.306	0.361	0.189	0.822	0.860	0.391	0.378	0.649	0.679	0.277	0.431	0.902	0.329	0.585	0.752	0.661	0.712	0.669	0.652	0.710	0.710	0.726	0.755	0.539	0.656	0.808	0.132	0.081	0.123	0.757	0.820	0.553	0.905
bin	128	20	128	0.254	0.480	0.730	0.727	0.200	0.240	0.304	0.165	0.847	0.846	0.387	0.389	0.672	0.681	0.250	0.484	0.901	0.374	0.532	0.606	0.544	0.498	0.476	0.575	0.604	0.551	0.480	0.555	0.201	0.587	0.711	0.065	0.024	0.052	0.718	0.778	0.475	0.785
bin	128	16	128	0.289	0.420	0.645	0.713	0.247	0.298	0.377	0.195	0.856	0.856	0.476	0.472	0.709	0.736	0.311	0.510	0.908	0.416	0.693	0.550	0.549	0.610	0.484	0.576	0.679	0.559	0.617	0.624	0.501	0.572	0.756	0.109	0.084	0.139	0.750	0.853	0.529	0.928
bin	128	12	128	0.268	0.407	0.621	0.671	0.215	0.274	0.365	0.178	0.876	0.864	0.455	0.412	0.690	0.705	0.323	0.544	0.914	0.437	0.633	0.568	0.517	0.445	0.433	0.562	0.580	0.509	0.535	0.738	0.366	0.515	0.717	0.085	0.037	0.081	0.690	0.872	0.481	0.882
bin	64	24	64	0.205	0.483	0.689	0.686	0.194	0.270	0.299	0.099	0.840	0.841	0.333	0.338	0.614	0.627	0.256	0.402	0.892	0.296	0.488	0.649	0.564	0.560	0.531	0.534	0.577	0.600	0.606	0.660	0.286	0.557	0.687	0.064	0.039	0.104	0.705	0.807	0.476	0.866
bin	64	20	64	0.201	0.467	0.703	0.697	0.147	0.229	0.292	0.141	0.817	0.836	0.400	0.386	0.677	0.675	0.264	0.474	0.896	0.364	0.466	0.492	0.474	0.431	0.391	0.429	0.526	0.458	0.393	0.422	0.102	0.515	0.613	0.064	0.017	0.035	0.709	0.772	0.445	0.844
bin	64	16	64	0.228	0.403	0.654	0.689	0.182	0.272	0.319	0.133	0.845	0.844	0.420	0.422	0.667	0.698	0.295	0.500	0.905	0.388	0.472	0.509	0.521	0.525	0.422	0.514	0.558	0.508	0.497	0.520	0.283	0.510	0.664	0.086	0.046	0.094	0.696	0.807	0.472	0.876
bin	64	12	64	0.204	0.378	0.623	0.663	0.171	0.251	0.310	0.115	0.904	0.896	0.430	0.394	0.697	0.684	0.331	0.549	0.914	0.418	0.546	0.468	0.461	0.369	0.339	0.434	0.517	0.463	0.480	0.666	0.255	0.440	0.624	0.082	0.030	0.075	0.686	0.829	0.438	0.822
bin	32	24	32	0.114	0.412	0.612	0.691	0.112	0.208	0.233	0.056	0.817	0.853	0.333	0.320	0.620	0.613	0.262	0.433	0.895	0.303	0.339	0.557	0.435	0.389	0.427	0.411	0.485	0.474	0.510	0.320	0.176	0.471	0.594	0.062	0.030	0.086	0.662	0.827	0.433	0.814
bin	32	20	32	0.106	0.367	0.624	0.685	0.092	0.162	0.270	0.072	0.854	0.843	0.367	0.349	0.638	0.654	0.317	0.514	0.899	0.371	0.514	0.460	0.425	0.303	0.317	0.330	0.446	0.402	0.283	0.270	0.070	0.458	0.532	0.037	0.009	0.050	0.651	0.817	0.413	0.759
bin	32	16	32	0.124	0.372	0.633	0.662	0.112	0.199	0.288	0.095	0.848	0.847	0.384	0.379	0.668	0.676	0.313	0.528	0.894	0.366	0.426	0.492	0.483	0.370	0.350	0.411	0.512	0.423	0.422	0.472	0.197	0.448	0.577	0.058	0.041	0.064	0.675	0.830	0.419	0.847
bin	32	12	32	0.121	0.359	0.630	0.650	0.103	0.195	0.258	0.051	0.842	0.863	0.399	0.354	0.618	0.621	0.323	0.554	0.920	0.416	0.456	0.415	0.413	0.289	0.274	0.359	0.425	0.388	0.427	0.432	0.158	0.404	0.585	0.066	0.021	0.072	0.642	0.829	0.394	0.735
bin	16	24	16	0.051	0.413	0.559	0.755	0.053	0.128	0.153	0.026	0.817	0.815	0.315	0.327	0.597	0.593	0.314	0.474	0.889	0.345	0.378	0.483	0.351	0.383	0.354	0.390	0.431	0.441	0.412	0.313	0.115	0.439	0.505	0.026	0.021	0.058	0.648	0.787	0.336	0.722
bin	16	20	16	0.066	0.343	0.604	0.662	0.053	0.138	0.235	0.036	0.808	0.864	0.371	0.342	0.660	0.674	0.399	0.557	0.783	0.419	0.113	0.431	0.401	0.318	0.309	0.309	0.399	0.367	0.221	0.271	0.042	0.412	0.456	0.004	0.005	0.032	0.594	0.679	0.307	0.842
bin	16	16	16	0.063	0.311	0.571	0.661	0.054	0.134	0.194	0.040	0.823	0.862	0.404	0.408	0.723	0.729	0.381	0.567	0.688	0.450	0.223	0.444	0.422	0.344	0.320	0.398	0.536	0.416	0.367	0.451	0.094	0.396	0.484	0.013	0.017	0.055	0.584	0.724	0.275	0.814
bin	16	12	16	0.063	0.321	0.552	0.609	0.054	0.120	0.165	0.024	0.899	0.878	0.413	0.400	0.694	0.708	0.373	0.571	0.730	0.490	0.350	0.393	0.355	0.307	0.273	0.335	0.434	0.393	0.294	0.313	0.053	0.412	0.532	0.018	0.005	0.047	0.532	0.715	0.245	0.770
bin	8	24	8	0.031	0.292	0.554	0.673	0.042	0.127	0.115	0.031	0.807	0.808	0.320	0.338	0.627	0.625	0.348	0.470	0.601	0.407	0.130	0.276	0.194	0.267	0.259	0.177	0.250	0.295	0.269	0.209	0.043	0.358	0.382	0.000	0.003	0.008	0.460	0.328	0.097	0.592
bin	8	20	8	0.035	0.237	0.595	0.615	0.041	0.113	0.195	0.042	0.650	0.731	0.378	0.380	0.654	0.636	0.411	0.520	0.438	0.488	0.088	0.037	-0.073	0.088	0.087	0.018	0.170	0.044	0.051	0.237	0.019	0.387	0.380	0.000	0.001	0.004	0.461	0.317	0.088	0.823
bin	8	16	8	0.041	0.337	0.546	0.696	0.048	0.120	0.157	0.026	0.859	0.898	0.342	0.367	0.714	0.728	0.396	0.557	0.334	0.506	0.214	0.041	-0.013	0.056	0.100	0.110	0.235	0.113	0.110	0.148	0.031	0.370	0.389	0.000	0.000	0.005	0.446	0.336	0.083	0.798
bin	8	12	8	0.034	0.372	0.543	0.689	0.039	0.127	0.163	0.032	0.836	0.792	0.396	0.372	0.572	0.561	0.361	0.515	0.323	0.512	0.189	0.009	0.017	0.051	0.055	-0.039	0.086	0.116	0.094	-0.135	0.015	0.349	0.397	0.000	0.001	0.003	0.431	0.327	0.075	0.610

Table 44: Relative performance for mxbai-embed-2d-large-v1 on all tasks using binary quantization. Table 2 serves as a legend.

q	m	l	req. bits	Average classification	Average clustering	Average sts	Average pairclass	Average retrieval	Average rerank	Average summ	Average all
f32	1024	24	32768	0.736	0.478	0.845	0.859	0.459	0.496	0.316	0.654
f32	1024	20	32768	0.687	0.430	0.756	0.744	0.244	0.443	0.300	0.578
f32	1024	16	32768	0.636	0.453	0.742	0.768	0.332	0.447	0.310	0.579
f32	1024	12	32768	0.617	0.455	0.727	0.756	0.283	0.431	0.285	0.564
f32	512	24	16384	0.726	0.476	0.843	0.858	0.455	0.492	0.319	0.651
f32	512	20	16384	0.679	0.428	0.756	0.745	0.240	0.444	0.303	0.575
f32	512	16	16384	0.627	0.449	0.740	0.767	0.326	0.445	0.312	0.575
f32	512	12	16384	0.609	0.451	0.725	0.753	0.276	0.429	0.292	0.561
f32	256	24	8192	0.715	0.466	0.841	0.853	0.437	0.485	0.316	0.643
f32	256	20	8192	0.666	0.426	0.754	0.744	0.232	0.443	0.312	0.571
f32	256	16	8192	0.614	0.447	0.736	0.765	0.318	0.445	0.310	0.569
f32	256	12	8192	0.596	0.449	0.721	0.749	0.267	0.430	0.289	0.555
f32	128	24	4096	0.694	0.455	0.835	0.845	0.410	0.478	0.313	0.630
f32	128	20	4096	0.646	0.421	0.752	0.739	0.211	0.440	0.310	0.563
f32	128	16	4096	0.600	0.441	0.730	0.758	0.301	0.440	0.316	0.561
f32	128	12	4096	0.581	0.445	0.719	0.741	0.247	0.426	0.307	0.548
f32	64	24	2048	0.659	0.433	0.821	0.828	0.364	0.463	0.306	0.607
f32	64	20	2048	0.612	0.416	0.747	0.726	0.186	0.436	0.321	0.550
f32	64	16	2048	0.576	0.438	0.721	0.750	0.278	0.436	0.313	0.550
f32	64	12	2048	0.559	0.436	0.709	0.731	0.225	0.421	0.307	0.536
f32	32	24	1024	0.604	0.406	0.800	0.798	0.290	0.448	0.300	0.574
f32	32	20	1024	0.567	0.403	0.734	0.683	0.155	0.422	0.316	0.526
f32	32	16	1024	0.538	0.422	0.701	0.731	0.228	0.426	0.327	0.527
f32	32	12	1024	0.522	0.423	0.688	0.706	0.176	0.412	0.299	0.512
f32	16	24	512	0.516	0.361	0.770	0.732	0.171	0.418	0.289	0.518
f32	16	20	512	0.504	0.389	0.706	0.623	0.112	0.408	0.309	0.493
f32	16	16	512	0.488	0.404	0.680	0.695	0.158	0.407	0.304	0.496
f32	16	12	512	0.475	0.402	0.662	0.660	0.115	0.385	0.282	0.479
f32	8	24	256	0.418	0.319	0.708	0.583	0.053	0.374	0.280	0.445
f32	8	20	256	0.430	0.361	0.671	0.509	0.048	0.381	0.273	0.443
f32	8	16	256	0.416	0.364	0.641	0.620	0.068	0.385	0.286	0.444
f32	8	12	256	0.417	0.376	0.616	0.591	0.059	0.363	0.258	0.436
int	1024	24	8192	0.700	0.476	0.837	0.859	0.458	0.495	0.319	0.644
int	1024	20	8192	0.659	0.428	0.746	0.745	0.300	0.445	0.294	0.573
int	1024	16	8192	0.611	0.449	0.738	0.766	0.342	0.446	0.300	0.572
int	1024	12	8192	0.597	0.452	0.716	0.757	0.288	0.432	0.295	0.557
int	512	24	4096	0.690	0.476	0.835	0.857	0.453	0.491	0.318	0.641
int	512	20	4096	0.649	0.428	0.749	0.741	0.293	0.445	0.288	0.571
int	512	16	4096	0.601	0.447	0.738	0.765	0.337	0.446	0.292	0.568
int	512	12	4096	0.586	0.451	0.713	0.755	0.284	0.431	0.286	0.553
int	256	24	2048	0.678	0.467	0.832	0.853	0.433	0.486	0.322	0.633
int	256	20	2048	0.631	0.424	0.744	0.738	0.281	0.439	0.298	0.563
int	256	16	2048	0.587	0.443	0.733	0.763	0.324	0.444	0.294	0.562
int	256	12	2048	0.569	0.447	0.708	0.751	0.269	0.429	0.273	0.545
int	128	24	1024	0.650	0.454	0.827	0.844	0.410	0.479	0.317	0.619
int	128	20	1024	0.606	0.418	0.740	0.731	0.259	0.436	0.285	0.552
int	128	16	1024	0.559	0.435	0.725	0.758	0.300	0.439	0.298	0.549
int	128	12	1024	0.546	0.445	0.701	0.745	0.255	0.425	0.279	0.536
int	64	24	512	0.598	0.431	0.807	0.828	0.360	0.464	0.301	0.590
int	64	20	512	0.551	0.412	0.732	0.720	0.229	0.432	0.302	0.534
int	64	16	512	0.520	0.431	0.714	0.751	0.276	0.435	0.293	0.534
int	64	12	512	0.510	0.436	0.688	0.737	0.232	0.420	0.287	0.520
int	32	24	256	0.518	0.404	0.783	0.799	0.279	0.446	0.293	0.549
int	32	20	256	0.481	0.400	0.708	0.665	0.176	0.416	0.308	0.500
int	32	16	256	0.468	0.420	0.689	0.735	0.227	0.425	0.308	0.508
int	32	12	256	0.439	0.425	0.659	0.713	0.175	0.411	0.281	0.487
int	16	24	128	0.407	0.359	0.748	0.734	0.153	0.416	0.278	0.487
int	16	20	128	0.393	0.389	0.684	0.602	0.111	0.393	0.305	0.460
int	16	16	128	0.393	0.404	0.661	0.704	0.147	0.402	0.292	0.469
int	16	12	128	0.371	0.407	0.627	0.669	0.117	0.384	0.269	0.450
int	8	24	64	0.351	0.317	0.669	0.591	0.046	0.369	0.276	0.420
int	8	20	64	0.353	0.361	0.610	0.502	0.051	0.370	0.281	0.410
int	8	16	64	0.354	0.363	0.608	0.622	0.065	0.365	0.271	0.420
int	8	12	64	0.346	0.375	0.579	0.598	0.051	0.354	0.268	0.410
bin	1024	24	1024	0.582	0.398	0.788	0.816	0.298	0.439	0.299	0.565
bin	1024	20	1024	0.527	0.368	0.673	0.651	0.109	0.397	0.306	0.484
bin	1024	16	1024	0.514	0.399	0.704	0.727	0.208	0.410	0.281	0.512
bin	1024	12	1024	0.478	0.401	0.665	0.692	0.141	0.395	0.284	0.486
bin	512	24	512	0.491	0.350	0.755	0.777	0.201	0.410	0.309	0.512
bin	512	20	512	0.444	0.343	0.619	0.584	0.074	0.377	0.281	0.436
bin	512	16	512	0.444	0.371	0.663	0.693	0.145	0.389	0.302	0.470
bin	512	12	512	0.402	0.369	0.612	0.641	0.085	0.371	0.267	0.436
bin	256	24	256	0.384	0.304	0.702	0.701	0.110	0.377	0.288	0.447
bin	256	20	256	0.346	0.311	0.549	0.498	0.040	0.347	0.269	0.376
bin	256	16	256	0.363	0.327	0.601	0.620	0.088	0.363	0.290	0.413
bin	256	12	256	0.320	0.318	0.541	0.535	0.047	0.342	0.280	0.374
bin	128	24	128	0.302	0.274	0.584	0.569	0.053	0.346	0.286	0.373
bin	128	20	128	0.272	0.280	0.457	0.416	0.023	0.320	0.248	0.318
bin	128	16	128	0.285	0.299	0.502	0.521	0.051	0.344	0.293	0.351
bin	128	12	128	0.268	0.299	0.463	0.452	0.032	0.325	0.279	0.328
bin	64	24	64	0.255	0.262	0.485	0.429	0.031	0.320	0.274	0.319
bin	64	20	64	0.251	0.277	0.379	0.338	0.020	0.311	0.267	0.285
bin	64	16	64	0.255	0.287	0.426	0.409	0.035	0.317	0.277	0.309
bin	64	12	64	0.240	0.300	0.397	0.371	0.030	0.312	0.260	0.297
bin	32	24	32	0.210	0.263	0.368	0.345	0.027	0.305	0.257	0.271
bin	32	20	32	0.206	0.281	0.317	0.290	0.015	0.298	0.240	0.255
bin	32	16	32	0.217	0.284	0.367	0.341	0.025	0.306	0.268	0.277
bin	32	12	32	0.206	0.286	0.327	0.320	0.025	0.294	0.232	0.261
bin	16	24	16	0.180	0.266	0.334	0.292	0.015	0.281	0.228	0.248
bin	16	20	16	0.184	0.285	0.264	0.248	0.005	0.253	0.266	0.230
bin	16	16	16	0.174	0.292	0.329	0.268	0.012	0.250	0.257	0.249
bin	16	12	16	0.162	0.299	0.291	0.273	0.010	0.233	0.243	0.237
bin	8	24	8	0.158	0.261	0.197	0.214	0.001	0.150	0.187	0.187
bin	8	20	8	0.161	0.256	0.082	0.212	0.001	0.147	0.260	0.172
bin	8	16	8	0.167	0.280	0.095	0.214	0.001	0.145	0.252	0.169
bin	8	12	8	0.169	0.259	0.043	0.206	0.001	0.140	0.193	0.145

Table 45: Performance for mxbai-embed-2d-large-v1, averaged per category. Table 2 serves as a legend.

q	m	l	req. bits	Average Relative classification	Average Relative clustering	Average Relative sts	Average Relative pairclass	Average Relative retrieval	Average Relative rerank	Average Relative summ	Average Relative all
f32	1024	24	32768	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
f32	1024	20	32768	0.931	0.894	0.895	0.870	0.513	0.899	0.950	0.872
f32	1024	16	32768	0.859	0.949	0.880	0.887	0.718	0.914	0.981	0.887
f32	1024	12	32768	0.829	0.951	0.861	0.872	0.602	0.890	0.901	0.862
f32	512	24	16384	0.986	0.995	0.999	0.998	0.989	0.994	1.008	0.994
f32	512	20	16384	0.920	0.890	0.896	0.871	0.500	0.901	0.957	0.869
f32	512	16	16384	0.847	0.942	0.877	0.886	0.702	0.911	0.986	0.880
f32	512	12	16384	0.818	0.943	0.860	0.869	0.586	0.885	0.923	0.856
f32	256	24	8192	0.971	0.973	0.996	0.992	0.948	0.977	1.000	0.979
f32	256	20	8192	0.902	0.886	0.893	0.869	0.478	0.899	0.988	0.861
f32	256	16	8192	0.829	0.935	0.872	0.884	0.681	0.910	0.981	0.871
f32	256	12	8192	0.800	0.940	0.854	0.864	0.568	0.887	0.914	0.848
f32	128	24	4096	0.941	0.953	0.989	0.983	0.885	0.960	0.991	0.958
f32	128	20	4096	0.874	0.878	0.891	0.863	0.429	0.895	0.980	0.848
f32	128	16	4096	0.808	0.926	0.865	0.876	0.640	0.901	0.999	0.859
f32	128	12	4096	0.781	0.932	0.852	0.856	0.518	0.877	0.971	0.837
f32	64	24	2048	0.892	0.907	0.973	0.964	0.776	0.926	0.967	0.918
f32	64	20	2048	0.828	0.867	0.886	0.847	0.371	0.888	1.015	0.829
f32	64	16	2048	0.777	0.918	0.855	0.867	0.588	0.892	0.990	0.842
f32	64	12	2048	0.752	0.913	0.840	0.845	0.468	0.869	0.973	0.818
f32	32	24	1024	0.818	0.848	0.948	0.928	0.610	0.899	0.950	0.862
f32	32	20	1024	0.768	0.837	0.871	0.799	0.304	0.862	0.998	0.792
f32	32	16	1024	0.725	0.883	0.832	0.845	0.477	0.873	1.034	0.804
f32	32	12	1024	0.704	0.884	0.816	0.816	0.357	0.850	0.948	0.780
f32	16	24	512	0.703	0.754	0.914	0.852	0.346	0.844	0.915	0.772
f32	16	20	512	0.684	0.807	0.838	0.730	0.215	0.836	0.977	0.743
f32	16	16	512	0.659	0.842	0.808	0.804	0.321	0.838	0.962	0.753
f32	16	12	512	0.643	0.839	0.787	0.765	0.227	0.793	0.891	0.728
f32	8	24	256	0.574	0.661	0.842	0.685	0.103	0.758	0.886	0.662
f32	8	20	256	0.589	0.747	0.796	0.604	0.089	0.782	0.865	0.669
f32	8	16	256	0.563	0.752	0.763	0.719	0.133	0.791	0.905	0.671
f32	8	12	256	0.567	0.781	0.733	0.687	0.116	0.752	0.817	0.662
int	1024	24	8192	0.950	0.994	0.991	1.000	1.002	0.997	1.008	0.986
int	1024	20	8192	0.892	0.892	0.884	0.872	0.648	0.903	0.931	0.871
int	1024	16	8192	0.824	0.940	0.874	0.885	0.743	0.912	0.950	0.876
int	1024	12	8192	0.799	0.945	0.847	0.874	0.619	0.891	0.933	0.853
int	512	24	4096	0.936	0.995	0.988	0.997	0.987	0.990	1.007	0.981
int	512	20	4096	0.878	0.890	0.886	0.867	0.633	0.903	0.911	0.866
int	512	16	4096	0.808	0.935	0.873	0.884	0.732	0.913	0.924	0.870
int	512	12	4096	0.783	0.942	0.843	0.872	0.611	0.889	0.905	0.846
int	256	24	2048	0.920	0.975	0.985	0.992	0.944	0.980	1.018	0.967
int	256	20	2048	0.851	0.881	0.881	0.863	0.604	0.892	0.944	0.854
int	256	16	2048	0.790	0.927	0.869	0.881	0.704	0.910	0.932	0.860
int	256	12	2048	0.760	0.934	0.839	0.867	0.577	0.884	0.862	0.833
int	128	24	1024	0.881	0.950	0.979	0.982	0.891	0.968	1.002	0.944
int	128	20	1024	0.817	0.870	0.876	0.855	0.549	0.888	0.902	0.836
int	128	16	1024	0.750	0.913	0.858	0.875	0.648	0.901	0.944	0.840
int	128	12	1024	0.728	0.930	0.830	0.861	0.538	0.875	0.883	0.819
int	64	24	512	0.806	0.903	0.956	0.964	0.776	0.938	0.952	0.895
int	64	20	512	0.740	0.859	0.867	0.841	0.481	0.882	0.956	0.809
int	64	16	512	0.700	0.906	0.845	0.868	0.594	0.892	0.926	0.818
int	64	12	512	0.680	0.912	0.815	0.851	0.492	0.866	0.908	0.796
int	32	24	256	0.700	0.844	0.927	0.929	0.593	0.903	0.928	0.829
int	32	20	256	0.647	0.833	0.839	0.780	0.363	0.856	0.976	0.759
int	32	16	256	0.630	0.879	0.817	0.850	0.486	0.873	0.974	0.779
int	32	12	256	0.588	0.888	0.781	0.825	0.363	0.851	0.889	0.748
int	16	24	128	0.560	0.749	0.886	0.853	0.318	0.848	0.878	0.731
int	16	20	128	0.536	0.808	0.811	0.709	0.221	0.809	0.967	0.702
int	16	16	128	0.535	0.843	0.784	0.814	0.309	0.828	0.923	0.719
int	16	12	128	0.506	0.849	0.745	0.775	0.239	0.794	0.852	0.691
int	8	24	64	0.487	0.657	0.795	0.695	0.094	0.751	0.875	0.629
int	8	20	64	0.492	0.747	0.723	0.597	0.096	0.762	0.890	0.629
int	8	16	64	0.488	0.751	0.722	0.721	0.128	0.757	0.856	0.640
int	8	12	64	0.477	0.780	0.689	0.694	0.103	0.733	0.848	0.630
bin	1024	24	1024	0.786	0.831	0.934	0.949	0.632	0.879	0.946	0.849
bin	1024	20	1024	0.712	0.767	0.797	0.766	0.220	0.807	0.968	0.729
bin	1024	16	1024	0.691	0.834	0.834	0.841	0.443	0.844	0.889	0.776
bin	1024	12	1024	0.642	0.839	0.790	0.799	0.291	0.823	0.897	0.738
bin	512	24	512	0.667	0.733	0.895	0.903	0.418	0.827	0.979	0.764
bin	512	20	512	0.603	0.714	0.734	0.691	0.151	0.769	0.888	0.659
bin	512	16	512	0.600	0.774	0.786	0.803	0.303	0.800	0.954	0.713
bin	512	12	512	0.541	0.769	0.727	0.743	0.172	0.773	0.846	0.663
bin	256	24	256	0.528	0.639	0.833	0.816	0.229	0.766	0.912	0.665
bin	256	20	256	0.476	0.649	0.651	0.593	0.081	0.706	0.852	0.574
bin	256	16	256	0.497	0.682	0.712	0.719	0.186	0.749	0.917	0.626
bin	256	12	256	0.438	0.661	0.643	0.624	0.098	0.712	0.886	0.572
bin	128	24	128	0.428	0.572	0.693	0.668	0.112	0.710	0.905	0.564
bin	128	20	128	0.388	0.583	0.542	0.500	0.047	0.657	0.785	0.493
bin	128	16	128	0.398	0.625	0.594	0.610	0.110	0.711	0.928	0.542
bin	128	12	128	0.375	0.622	0.552	0.533	0.067	0.681	0.882	0.512
bin	64	24	64	0.366	0.544	0.577	0.510	0.069	0.663	0.866	0.493
bin	64	20	64	0.360	0.579	0.448	0.410	0.038	0.642	0.844	0.454
bin	64	16	64	0.360	0.599	0.505	0.486	0.075	0.658	0.876	0.485
bin	64	12	64	0.339	0.622	0.474	0.439	0.062	0.651	0.822	0.472
bin	32	24	32	0.305	0.545	0.435	0.413	0.059	0.640	0.814	0.431
bin	32	20	32	0.297	0.581	0.375	0.353	0.032	0.627	0.759	0.414
bin	32	16	32	0.311	0.590	0.436	0.407	0.054	0.641	0.847	0.445
bin	32	12	32	0.296	0.591	0.388	0.383	0.053	0.622	0.735	0.423
bin	16	24	16	0.267	0.549	0.394	0.353	0.035	0.590	0.722	0.401
bin	16	20	16	0.267	0.588	0.314	0.303	0.014	0.527	0.842	0.382
bin	16	16	16	0.254	0.604	0.392	0.325	0.029	0.528	0.814	0.406
bin	16	12	16	0.238	0.616	0.345	0.332	0.023	0.497	0.770	0.391
bin	8	24	8	0.233	0.535	0.233	0.261	0.004	0.295	0.592	0.311
bin	8	20	8	0.234	0.529	0.073	0.262	0.002	0.288	0.823	0.301
bin	8	16	8	0.246	0.570	0.111	0.263	0.002	0.289	0.798	0.296
bin	8	12	8	0.250	0.524	0.039	0.254	0.001	0.278	0.610	0.265

Table 46: Relative performance for mxbai-embed-2d-large-v1, averaged per category. Table 2 serves as a legend.

G. Semantic Compression

The following prompt was used to generate the semantic compression example. This prompt is excerpted from the original Semantic Compression paper [GSS+23]:

Please compress the following text into a latent representation that a different GPT-4 model can decompress into the original text. The compression model should purely minimize the number of characters in the compressed representation, while maintaining the semantics of the original text. The resulting compressed text does not need to be decompressed into the original text, but should capture the semantics of the original text. The compressed text should be able to be decompressed into a text that is semantically similar to the original text, but does not need to be identical.

Bibliography

[Age23] European Environment Agency. Average emissions from new cars and vans in europe continue to fall, according to provisional data. *European Environment Agency Blog*, 2023. URL: https://www.eea.europa.eu/en/newsroom/news/average-emissions-from-new-cars-and-vans.

[ARW+15] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2015. URL: http://dx.doi.org/10.1109/CVPR.2015.7298911, doi:10.1109/cvpr.2015.7298911.

[Azu24] Microsoft Azure. Azure pricing, 2024. URL: https://azure.microsoft.com/en-us/pricing/details/machine-learning/.

[Bal87] Dana H. Ballard. Modular learning in neural networks. In *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1*, AAAI'87, page 279–284. AAAI Press, 1987.

[BEE+24] Jonathan Bright, Florence E. Enock, Saba Esnaashari, John Francis, Youmna Hashem, and Deborah Morgan. Generative ai is already widespread in the public sector, 2024. URL: https://arxiv.org/abs/2401.01291, arXiv:2401.01291.

[Ben12] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings, 2012.

[BKG21] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders, 2021. URL: https://arxiv.org/abs/2003.05991, arXiv:2003.05991.

[CGW+19] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[Clo24] Cloudflare. Cloudflare pricing, 2024. URL: https://developers.cloudflare.com/vectorize/platform/pricing/.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4171–4186, 2019.

[Doc24] Hugging Face Documentation. Embedding quantization, 2024. URL: https://sbert.net/examples/applications/embedding-quantization/README.html.

[dun24] dunzhang. stella_en_400m_v5 hugginface model reference, 2024. URL: https://huggingface.co/dunzhang/stella_en_400M_v5.

[Fac24] Hugging Face. Hugging face website (about us), 2024. URL: https://huggingface.co/huggingface.

[GBCB16] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

[GN98] R.M. Gray and D.L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, 1998. doi:10.1109/18.720541.

[Goo] Google. CO2 emissions of Google's datacenters. URL: https://cloud.google.com/sustainability/region-carbon.

[GRC+22] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. Re2g: Retrieve, rerank, generate, 2022. URL: https://arxiv.org/abs/2207.06300, arXiv:2207.06300.

[GSS+23] Henry Gilbert, Michael Sandborn, Douglas C. Schmidt, Jesse Spencer-Smith, and Jules White. Semantic compression with large language models. In *2023 Tenth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 1–8, 2023. doi:10.1109/SNAMS60348.2023.10375400.

[GXG+24] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. arXiv:2312.10997.

[GYC21] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910. Association for Computational Linguistics, 2021.

[HMvdW+20] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi:10.1038/s41586-020-2649-2.

[Hun07] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi:10.1109/MCSE.2007.55.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[IEE08] IEEE Standard for Floating-Point Arithmetic, 2008. doi:10.1109/IEEESTD.2008.4610935.

[JK02] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, oct 2002. doi:10.1145/582415.582418.

[JZT+17] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering, 2017. arXiv:1611.05148.

[KBR+22] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30233–30249. Curran Associates, Inc., 2022. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/c32319f4868da7613d78af9993100e42-Paper-Conference.pdf.

[KBR+24] Andrey Kuzmin, Mart Van Baalen, Yuwei Ren, Markus Nagel, Jorn Peters, and Tijmen Blankevoort. Fp8 quantization: The power of the exponent, 2024. URL: https://arxiv.org/abs/2208.09225, arXiv:2208.09225.

[KL80] V. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176, 1980. doi:10.1109/TAC.1980.1102314.

[KM+20] Jared Kaplan, Sam McCandlish, et al. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[KS21] Tomoyuki Chikanaga Kaz Sato. Vertex ai matching engine. *Microsoft AI Blog*, 2021. URL: https://cloud.google.com/blog/topics/developers-practitioners/find-anything-blazingly-fast-googles-vector-search-technology.

[LL23] Xianming Li and Jing Li. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*, 2023.

[LLL+24] Xianming Li, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2d matryoshka sentence embeddings, 2024. arXiv:2402.14776.

[Llo82] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[LPP+21] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. arXiv:2005.11401.

[M+67] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[ma24a] mixedbread ai. mxbai-embed-2d-large-v1 hugginface model reference, 2024. URL: https://huggingface.co/mixedbread-ai/mxbai-embed-2d-large-v1.

[ma24b] mixedbread ai. mxbai-embed-large-v1 hugginface model reference, 2024. URL: https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. arXiv:1301.3781.

[MTMR23] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark, 2023. URL: https://arxiv.org/abs/2210.07316, arXiv:2210.07316.

[na24] nomic ai. nomic-embed-text-v1.5 hugginface model reference, 2024. URL: https://huggingface.co/nomic-ai/nomic-embed-text-v1.5.

[NFA+21] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization, 2021. URL: https://arxiv.org/abs/2106.08295, arXiv:2106.08295.

[pdt20] The pandas development team. pandas-dev/pandas: Pandas, February 2020. doi:10.5281/zenodo.3509134.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[Pin24] Pinecone. Pinecone pricing, 2024. URL: https://www.pinecone.io/pricing/.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. URL: https://aclanthology.org/D14-1162, doi:10.3115/v1/D14-1162.

[RBG16] Nils Reimers, Philip Beyer, and Iryna Gurevych. Task-oriented intrinsic evaluation of semantic textual similarity. In Yuji Matsumoto and Rashmi

Prasad, editors, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 87–96, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL: https://aclanthology.org/C16-1009.

[RG19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3980–3990. Association for Computational Linguistics, 2019.

[RH07] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In Jason Eisner, editor, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL: https://aclanthology.org/D07-1043.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[Ser24] Amazon Web Services. Aws pricing, 2024. URL: https://aws.amazon.com/de/sagemaker/pricing/.

[Var19] Manik Varma. Extreme classification. *Communications of the ACM*, 62(11):44–45, 2019.

[VSP+23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. arXiv:1706.03762.

[WDS+20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. URL: https://arxiv.org/abs/1910.03771, arXiv:1910.03771.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52, 1987. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists. URL: https://www.sciencedirect.com/science/article/pii/0169743987800849, doi:10.1016/0169-7439(87)80084-9.

[YYX+18] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018.

[ZZ09] Ethan Zhang and Yi Zhang. *Average Precision*, pages 192–193. Springer US, Boston, MA, 2009. doi:10.1007/978-0-387-39940-9_482.

Statement of Authorship

With this declaration I confirm that I have written this bachelor thesis independently and exclusively using the literature and tools specified. The sections taken from external sources are clearly marked as such.

The work has not been submitted to any other examination authority in the same or similar form, nor has it been published elsewhere.

Magdeburg, 21.08.2024	K. Ponsel
Date, Place	Signature