T-EMDE: Sketching-based global similarity for cross-modal retrieval

Barbara Rychalska*  
Warsaw University of Technology  
Poland

Mikolaj Wieczorek*  
Warsaw University of Technology  
Poland

Jacek Dabrowski  
Synerise  
Poland

ABSTRACT

The key challenge in cross-modal retrieval is to find similarities between objects represented with different modalities, such as image and text. However, each modality embeddings stem from non-related feature spaces, which causes the notorious 'heterogeneity gap'. Currently, many cross-modal methods try to bridge the gap with self-attention. However, self-attention has been widely criticized for its quadratic complexity, which prevents many real-life applications. In response to this, we propose T-EMDE - a neural density estimator inspired by the recently introduced Efficient Manifold Density Estimator (EMDE) from the area of recommender systems. EMDE operates on sketches - representations especially suitable for multimodal operations. However, EMDE is non-differentiable and ingests precomputed, static embeddings. With T-EMDE we introduce a trainable version of EMDE which allows full end-to-end training. In contrast to self-attention, the complexity of our solution is linear to the number of tokens/segments. As such, T-EMDE is a drop-in replacement for the self-attention module, with beneficial influence on both speed and metric performance in cross-modal settings. It facilitates communication between modalities, as each global text/image representation is expressed with a standardized sketch histogram which represents the same manifold structures irrespective of the underlying modality. We evaluate T-EMDE by introducing it into two recent cross-modal SOTA models and achieving new state-of-the-art results on multiple datasets and decreasing model latency by up to 20%.

1 INTRODUCTION

Cross-modal retrieval refers to expressing a single entity by means of various modalities, such as image, text or sound. For example, a garden party scene can be identified either by its photo, textual description (“a group of people laughing and talking”), or the recorded sound of the scene. Image-text matching is particularly common, e.g. in database retrieval tasks of photos via their textual description [14, 18, 33], image captioning [22, 24, 57], and multimodal neural machine translation [25, 36].

Cross-modal matching remains a challenge due to the 'heterogeneity gap', caused by the fact that modalities are represented by inherently nonmatching feature spaces [15, 32, 50]. Cross-modal methods strive to bridge the gap with various neural architectures, such as Graph Neural Networks (GNNs) and multiple forms of attention [6, 15, 31, 51, 60] in order to compute per-modality vectors which can be compared. The similarity (or distance) between these vectors is used to compute the probability of a correct match between image and caption. Unfortunately, both GNNs and attention networks are known for poor scalability and specific performance issues [3, 7, 59].

An important flavor of attention which is often found in cross-modal systems is the self-attention. It is used in various deep learning domains to obtain a summary representation of text or image. Self-attention has proven very successful, especially in the Transformer architecture [46], which has has led to many performance breakthroughs in areas such as machine translation, language modeling, and image classification [16, 38, 42, 53]. Yet, the core operation of self-attention is the dot product between a sequence representation and itself, which results in quadratic complexity. This characteristic makes self-attention inefficient in many real-life industrial applications. For this reason, attempts are made at devising more efficient architectures which could replace self-attention [4, 47].

In this paper we address the above mentioned problems with an efficient neural module which can serve as a drop-in replacement for self-attention, which we call T-EMDE. Our proposed solution is especially appropriate for computation of global representations of text and image in multimodal retrieval scenarios. T-EMDE is inspired by the recently introduced EMDE (Efficient Manifold Density Estimator) [17] which has proven competitive in multimodal recommendation scenarios. We design T-EMDE with three main goals in mind:

- **High efficiency.** T-EMDE has linear complexity with respect to the sequence size (text tokens or image segments). This stands in contrast to quadratic complexity of self-attention.
- **Unified cross-modal representation.** T-EMDE maps any modality to the same type of representation - a semi-histogram on multidimensional data manifolds. This unified representation bridges the problem of heterogeneity gap.
- **Differentiability.** While the EMDE architecture has been shown to achieve high results and avoid scalability problems characteristic for complex neural architectures, it is a non-differentiable algorithm which prevents the fine-tuning of embeddings. With T-EMDE we exploit the same ideas which give EMDE its advantages, while making it fully trainable.

We evaluate T-EMDE by introducing it into two recent high-performing models which use self-attention: SAF (Similarity Attention Filtration) and SGR (Similarity Graph Reasoning) [15]. By introducing T-EMDE, we are able to reach new state-of-the-art results on MSCOCO and Flickr30k datasets. We observe especially significant gains in the Recall@1 metric, representing the quality of the top returned recommendation. In multiple cases, the gains in Recall@1 are well over 1 pp. with comparison to the baseline SAF and SGR. At the same time, due to reduced complexity T-EMDE lowers model inference latency by up to 20%. We propose T-EMDE...

*Both authors contributed equally to this research.
as a step in the direction of replacing computationally expensive self-attention with more efficient approaches.

2 RELATED WORK

2.1 Text-image alignment

A number of prior works [19, 48, 62] focused on matching global image and text features in a joint embedding space. The aim was to learn to map both modalities to the same vector space, where the similarity measure could be applied for the cross-modal retrieval. One of the shortcomings of methods using only global representation was their ignorance of the fact that the notion of similarity may arise from an aggregation of numerous local similarities, such that salient objects in the image and keywords in the text. To address the problem, [26] explored mutual relations between image regions and words a in text. The authors proposed to infer a similarity score between each pair of an image region and each word to define the alignment of the corresponding fragment features in the shared embedding space. This approach of computing similarity on the level of a smaller units from each modality was explored in many subsequent works.

Since the Transformer architecture proved successful in various domain, the self-attention mechanism has been explored by many researchers in cross-modal retrieval settings [6, 15, 23, 31, 41, 52, 60]. [51] introduced Position Focused Attention Network (PFAN), which incorporates object position clues into the learning architecture. Images are split into blocks and attention mechanism is used to generate meaningful region features that are later fed into visual-textual attention to learn the joint-embedding space. [8] proposed Dual Path Recurrent Neural Network (DP-RNN), which symmetrically processes images and sentences. It 'reads' the image objects in the order the text indicates, reorders them based on the object-word matching and an RNN learns joint information embeddings of semantically related objects. Finally, a module consisting of attention and self-attention mechanism is applied to compute the text-image similarity from recurrent visual and textual features. [60] observe that many attention-based methods ignore the fact that image segments or words might have various semantic meaning that depends on the global context, which in turn was defined as intra-modal and inter-modal relations. They devised Context-Aware Attention Network (CAAN), which aggregates context information of alignments between modalities and within a single modality simultaneously. [6] proposed to use Iterative Matching with Recurrent Attention Memory (IMRAM) method to capture the mutual image and text alignments iteratively. It consists of two main parts: (1) a cross-modal attention applied multiple times that builds the fine-grained correspondence between modalities progressively, (2) a memory distillation unit which aggregates alignment knowledge from earlier steps and passes it to later ones.

[32] is an example of a Graph Convolutional Network model which generates visual features of salient objects and their semantic relationship. The gate and memory mechanism [12] is used to perform semantic reasoning, which considers both local and global semantic correlations. It enables the model to select information that contributes to a meaningful representation of the whole scene and ignore redundant cues at the same time. [50] introduces Scene Graph Matching (SGM) - an approach using separate graphs for each modality to capture object features and their mutual relationships in corresponding modality. When the object-level and relationship-level features are extracted from both modalities, the final matching on both levels takes place.

[31] proposed Stacked Cross Attention Network (SCAN), which is trained to discover the full latent alignment between salient objects and words. The SCAN can be defined as two complimentary formulations, where each modality is used as a context to each other and two separate similarity scores are inferred. The alignment is done on the local-region level, which is next aggregated to build a global representation. While SCAN used the cross-modal attention with success it lacked the self-attention part. [15] builds upon SCAN, extends it with self-attention and applies it to each modality separately. It introduced two modules - Similarity Attention Filtration (SAF) and Similarity Graph Reasoning (SGR) to boost the accuracy of alignments between modalities. SGR is based on a graph convolutional neural network (GCNN). It builds a similarity graph that propagates similarities at local and global levels to obtain the alignment scores. Similarity Attention Filtration (SAF) module was introduced as the authors noticed that using the local alignments generally boosts the performance, but incorporating the less relevant elements into the global representation decreases the final results. SAF aim is to decide which alignments are significant and which should be filtered out. The SGRAF model is an ensemble of SRG and SAF models, which similarity scores are averaged before the retrieval takes place.

2.2 Attention Mechanism

Attention is a mechanism that allows the network to learn correspondence between input and target sequence/set elements. One of the first works that introduced the popular attention mechanism was [2]. The authors proposed a simple additive module that computed 'attention scores' for each input element. The weights of the alignments were computed by feeding RNN hidden state and a word latent vector into a single-layer feedforward neural network. This allowed the RNN decoder to attend to any part of the source sentence and in turn it increased the quality of machine translation task. A similar approach can be found in Computer Vision, where attention was used to help the network find relevant images patches and object in the image to generate captions [55].

The Transformer architecture [46] introduced a scaled dot product self-attention computed over three elements: the Query, Key, and Value. This can be deemed as an extension of the attention from [2]. The Transformer architecture computes the self-attention multiple times separately, which is known as multi-head attention. According to the authors this allows the model to attend to information from different representation subspaces at different positions. Furthermore, the attention mechanism was extended by residual connections and normalization layers. Transformer model and its self-attention mechanism has started to dominate various domains. For example, it achieved state-of-the-art results in machine translation [38, 45], text summarizing [38], visual question answering [42, 43].

However, the Transformer architecture is notorious for its difficulty to train and its demand for large amounts of data. Most
current state-of-the-art solutions in cross-modal retrieval are non-Transformer models, which nevertheless use simplified dot-product self-attention modules commonly [6, 15, 23, 31, 41, 51, 56, 61].

2.3 Optimizing Self-Attention
Unfortunately, the time and memory complexity of self-attention is quadratic as it needs to compute the weights between each pair of inputs. It prevents efficient scalability of attention-based models, especially where longer sequences are present in the data. To tackle this problem many researchers introduced the extensions and improvements, which are however primarily dedicated for the Transformer architecture.

Some works attempt to limit Transformer self-attention by attending only to some inputs according to a fixed block/patterns [5, 9, 13, 37, 44]. Another approach to improve the complexity of Transformer self-attention is to use low-rank factorization methods. [45, 49] assumed low-rank structure of self-attention. They project the Transformer Key and Value vectors into a lower-dimensional representation. Some works [11, 27] used kernel functions to bring down the dimensionality of the elements on which the Transformer self-attention operates. [40, 54] introduced centroids/clusters into the Transformer self-attention. The idea is to calculate the attention scores only among the tokens belonging to the same cluster/centroid. [28] proposed Transformer LSH Attention, which uses Locally-Sensitive Hashing method to obtain hashes for both the Keys and the Queries.

Thus, most attempts at limiting the complexity of self-attention are dedicated for the Transformer’s formulation of self-attention, assuming the creation of multiple Query, Key, Value vectors to build multi-head attention within a strictly defined neural architecture. Yet, few attempts are made at optimization or replacement of the basic formulation of (self) attention. This comes at a disadvantage to models in which the simple classic attention composed of a single dot product and weighted summation of vectors is the preferred choice [15, 23, 51, 56]. With T-EMDE, we mean to bridge the gap by introducing a conceptually simple and fast alternative to classic self-attention. Additionally, we aim at easy communication between various modality representations, which is not handled inherently by neither formulation of attention.

3 PROPOSED METHOD
In this section we describe the following architectures: EMDE [17] which serves as an inspiration model, T-EMDE - our proposed module, and SAF/SGR [15] - models which serve as scaffolds into which T-EMDE is introduced.

3.1 Non-Trainable Efficient Manifold Density Estimator
Efficient Manifold Density Estimator (EMDE) introduced in [17] is a probability density estimator inspired by Count-Min Sketch algorithm (CMS) and local sensitive hashing (LSH). The overview of the algorithm is shown in Figure 2. EMDE ingests input data represented by vectors embedded on manifolds spanned by various upstream representation learning methods. The manifolds are then partitioned via a data-dependent LSH method (DLSH). The partitioning method divides the manifold into regions, analogous to CMS buckets. The purpose of manifold partitioning follows the logic of LSH: similar data points should be mapped to the same region. While a single region is large (typically 64-256 regions form a single partitioning covering the whole manifold, as described in [17]), multiple independent partitionings allow to obtain a high resolution map of the manifold via intersection or ensembling.

The resulting region assignment vectors (sketches) can be thought of as a form of a histogram. Each position within a sketch corresponds to a region, and each value represents the number of data points in the given region. For example, a sketch of a single item can take the form of \( I = [0, 0, 1, 0, 0] \), which means that the data point in question is located in region number 2 out of 5 total regions. All items are represented with sketches of the same dimensionality, and representations of item sets are computed by simple summation of individual item sketches. For example, a shopping basket of 4 items can be represented with as sketch \( S = [1, 0, 2, 0, 1] \) containing 4 items in total, one located in region number 0, two in region number 2, and one in region number 4. EMDE precomputes sketch representations of all items within the inventory, computes aggregate shopping basket sketches, and uses simple feed-forward network for mapping a basket sketch to a predicted sketch of items which will be likely bought next.

The algorithm is shown to achieve competitive results in product recommendation settings, which are often multimodal. Products can be characterized by user interactions (e.g. assignment to shopping baskets), textual descriptions, photos, and categorical features. In EMDE, all per-modality representations are transformed to the histogram-like sketches, which follow the same logic irrespective of the underlying modality or representation method. EMDE is shown to be fast and efficient - thanks to the application of the hashing-based CMS structure, it works in sublinear space. The neural architecture of EMDE is very simple, with just feed-forward layers.

However, in [17] the DLSH method is neither differentiable nor end-to-end trainable, due to the highly discontinuous binary-unity conversion being a key operation during assignment of inputs to
region indices. Thus, manifold partitionings must be precomputed before actual training and are based on pretrained vectors, which prevents the model from learning own embedding vectors or fine-tuning embeddings derived from upstream representations. A drop-in replacement module for attention must be trainable, as otherwise it would block off the gradient in a whole portion of the underlying network. In particular, cross-modal architectures often do not use precomputed embeddings but rather train their own representations from scratch, which makes it obligatory for all parts of the network to be differentiable.

3.2 Trainable Efficient Manifold Density Estimator

Taking into the account all key characteristics which an attention replacement module should possess, we introduce the T-EMDE as a trainable version of EMDE. We retain the core properties of EMDE: high scalability due to the application of a hashing-based data structure and a unified representation used for all modalities to bridge the heterogeneity gap. Moreover, we aim to retain the ability of EMDE to model item sets, which in our case will be sets of tokens and image segments. The ability of computing a single object representation out of its composing part sequence (of variable size) makes T-EMDE a valid candidate for replacing self-attention.

T-EMDE aims to partition the underlying data manifold, similarly as EMDE. However, instead of a static assignment of inputs to specific regions of the manifold, we propose to use trainable centroids. The centroids are created within a space of arbitrary dimensionality and represent the basis of item representation - each token/image segment will be characterized by a vector of distances to all centroids. An overview of EMDE is shown in Figure 1, presenting its application to multiple modalities.

Our algorithm proceeds as follows:

1. Centroid initialization. We initialize a trainable centroid tensor \( C \) of dimensionality \( N \times K \times D \). The tensor will hold \( K \) centroids for each of \( N \) independent manifold divisions. Independent manifold divisions are introduced with regard to the observation from [17] that a single manifold partitioning will usually not be perfect and multiple independent divisions can achieve an ensembling effect, which is beneficial to performance. Each centroid will be located within \( D \)-dimensional space and its coordinates will change during training to represent data assignments most appropriately. The parameter \( N \) corresponds to sketch depth from [17] (number of independent manifold partitionings), while \( K \) corresponds to sketch width (number of regions produced by each partitioning).

2. Embedding projection. Subsequently, we feed each item modality embedding to a simple linear layer to obtain a projected representation \( X \) of dimensionality \( N \times K \times D \). This way we bring the input embedding into the \( D \)-dimensional space in which the cluster centroids live. After this transformation, input data representation can be represented by its proximity to centroids. The vectors \( X \) are batch-normalized.

Distance computation. After bringing input data into the centroid space, we compute the distance between each \( X \) and each centroid as \( d = (X - C)^2 \). This represents a squared distance, but other distance metrics can be used if deemed appropriate. We then sum across \( D \), to obtain squared euclidean distance vectors \( X_{\text{euclid}} \) of dimensionality \( N \times K \). Finally, the obtained representations are normalized with \( Y = \text{softmax}(X_{\text{euclid}}) \) across the \( K \) dimension. The resulting vectors can be thought of as representing soft assignments to \( K \) centroids each, over \( N \) independent space divisions.

T-EMDE can be implemented in a few lines of code. We attach its pseudocode written in PyTorch convention in Code Listing 1.

It can be seen that T-EMDE exploits the feature space of various modalities, yet at the same time the output vector (the sketch) will represent solely the distances to centroids, losing all modality-specific characteristics. Comparison of two sketches coming from two T-EMDE coders for various modalities (e.g. text and image) will consist in defining the relationships between two sets of centroids.

3.3 SAF and SGR

SAF (Similarity Attention Filtration) and SGR (Similarity Graph Reasoning) are two recent state-of-the art models proposed by [15]. Like most recent cross-modal models, they are complex architectures composed of multiple submodules. They form the backbone of our proposed solution. Below we describe the composing modules.
of each solution (some are shared between models). An overview of SAF and SGR architectures is given in Figure 1.

**Image Representation.** Both models represent images as extracted segment embeddings. Such visual features are computed following [1] to extract K region-level visual features, with the Faster R-CNN [39] model pretrained on Visual Genomes [29]. Only segment embeddings are considered, without recognizing additional features such as segment frame locations or captions. Each segment is projected through a fully-connected layer to transform it to a desired size.

**Text Representation.** Captions are tokenized with the nltk package [35] and each token is represented by a randomly initialized embedding. The embeddings are fed to a bi-directional recurrent GRU network [10] and token representations are obtained by averaging the hidden states from the forward and backward GRU pass.

**Global Similarity Module.** Global representations are computed for text and image separately with self-attention modules. First, all representations are run through separate sequences of multiple linear layers alternating with batch normalization, to form local $L$ and global $G$ variants of each modality representations. Global representation is obtained by averaging per-segment or per-token representations, while local representation consists of token/segment-level vectors. Then, self-attention is computed with a dot product over the two inputs, with $G$ repeated to match the number of segments/tokens within $L$:

$$
\text{attn} \_\text{weights} = \text{softmax}(L \times \text{repeat}(G, \text{len}(L)))
$$

$$
\text{new} \_G = \sum_{j=1}^{n\_\text{columns}} (\text{attn} \_\text{weights} \cdot L)_{ij}
$$

After additional L2-normalization, $\text{new} \_G$ is the final per-modality global representation. Global similarity vector is computed by subtracting the global caption representation from the global image representation.

The Global Similarity Module is one of the computationally heavy parts of the architecture, especially that it needs to be computed two times - always both for text and image. It is replaced with T-EMDE in our solution.

**Local Similarity Module.** Local Similarity Module also computes attention, but in contrast to Global Similarity Module it is done between the textual and image representations. It uses the cross-modal attention formula from [31] and obtains a summary textual-visual representation.

**Similarity Graph Module.** This module uses a Graph Neural Network architecture from [30] to model the strength of connection between tokens and image parts. Token/segment local representations together with the global representation computed by the Global Similarity Module are treated as graph nodes within a fully-connected graph. Edges are computed by weighted multiplication of input and output node representations, with each node representation multiplied by trainable matrices $W_{in}$ and $W_{out}$. Similarity reasoning is performed by summing the edge representations of each node $p$, multiplied by the representations of neighbors of $p$, and running them through an extra layer with multiplication against a trainable matrix $W_r$ with a nonlinearity. The global representation nodes are finally fed to a linear layer to return the text-image similarity score.

**Similarity Attention Filtration.** Similarity Attention Filtration is designed to suppress tokens or segments which have a small influence on the final similarity (such as the function words “a”, “be”, etc.). It ingests vector similarity representations from previous modules (e.g. Global Similarity and Local Similarity) and runs them through additional linear layers, applying weights assessing

![Figure 2: T-EMDE module, with comparison to the static EMDE.](image-url)
the validity of each similarity alignment. A final linear layer ingesting weighted similarity representations returns the text-image similarity score.

As shown in Figure 1, SAF is composed of two Global Similarity Modules (each per modality), a Local Similarity Module, and the Attention Filtration which computes the final similarity scores. SGR also has two Global Similarity Modules and a Local Similarity Module, followed by Similarity Graph Module.

3.3.1 T-EMDE in SAF/SGR. T-EMDE can be easily introduced in place of the Global Similarity modules (or in fact any other formulation of self-attention). Thanks to the application of two very different neural architectures in SAF and SGR (a graph neural network and an attention filter), the versatility of application of T-EMDE to various architectures can be tested.

The module is applied in the same way to both SAF and SGR, replacing the Global Similarity modules for both image and text. In total, two T-EMDE modules are created, separately for image and text. The segment/token representations fed by the Image Representation and Token Representation modules are fed as input to respective T-EMDE modules. Individual cluster assignments of each token/segment are then retrieved. For each image and caption, the appropriate token/segment soft assignment vectors are summed, exploiting the additivity property of EMDE and Count-Min Sketch [17]. In EMDE, summation is done on item sketches which belong to one shopping basket/user in order to get a global basket/user representation. Similarly in cross-modal retrieval, summation can be thought of as aggregating multiple partwise histograms to get an overview profile of text/image - its global representation. The summation allows for keeping the global representation size constant irrespective of how many composing parts (tokens/segments) need to be aggregated. This mechanism allows us to handle the problem of variable sequence length and escape the necessity of comparing each item to all others which results in quadratic complexity of self-attention.

In contrast to Global Similarity module, the global representations are not subtracted, as corresponding centroids can get assigned to different places in the output sketches. Instead, we concatenate the text and image sketches (this can be done because their size is kept constant at all times) and map them to a smaller size with a linear layer, followed by ReLU activation.

4 EXPERIMENTS

4.1 Datasets

Our evaluation is aligned with [15] for fair comparison, we reuse their evaluation code\footnote{https://github.com/Paranioar/SGRAF} for full credibility and evaluate on the same datasets. Thus, we use two popular cross-modal datasets: MSCOCO [34] and Flicker30K [58] datasets. The MSCOCO dataset is a large-scale object detection, segmentation, and captioning dataset. It contains 123,287 photos, mainly from daily life scenes. Each image is annotated with 5 captions produced by crowdworkers from Amazon Mechanical Turk. We use a popular cross-modal retrieval split (the “Karpathy” split) which assigns 113,287 images for training, 5,000 images with 25,000 matching captions for validation, and 5,000 images for testing. We use two evaluation protocols:

- MSCOCO 1K - the results are computed by averaging over 5 folds, each of 1000 test images.
- MSCOCO 5K - the results are computed on the full set of 5000 images.

The Flicker30K dataset contains 31,783 images with 5 corresponding captions each. Generally, the Flicker30K captions are much longer and are in many cases more detailed than in MSCOCO. [21]. Following the split in [20], we use 1,000 images for validation, 1,000 images for testing and the rest for training.

### Table 1: Performance results on the MSCOCO 5K dataset (averaged over 5 runs).

| Model               | MSCOCO 5K dataset |
|---------------------|-------------------|
|                     | Text Retrieval    | Image Retrieval |
|                     | R@1   | R@10  | MRR   | R@1   | R@10  | MRR   |
| SAF noglobal        | 53.7  | 90.4  | 0.663 | 39.8  | 80.0  | 0.532 |
| SAF reported        | 53.3  | 90.1  | -     | 39.8  | 80.2  | -     |
| SAF reproduced      | 55.6  | 90.7  | 0.677 | 40.3  | 80.3  | 0.536 |
| T-EMDE              | 56.7  | 90.7  | 0.682 | 40.3  | 80.4  | 0.537 |
| SGR noglobal        | 54.2  | 90.3  | 0.668 | 39.3  | 79.3  | 0.525 |
| SGR reported        | 56.9  | 90.5  | -     | 40.2  | 79.8  | -     |
| SGR reproduced      | 55.7  | 90.3  | 0.677 | 39.8  | 79.9  | 0.531 |
| T-EMDE              | 57.0  | 91.0  | 0.685 | 40.0  | 80.1  | 0.533 |
| SGRAF reported      | 58.8  | 91.6  | -     | 41.6  | 81.5  | -     |
| SGRAF reproduced    | 57.9  | 91.6  | 0.697 | 41.8  | 81.5  | 0.550 |
| T-EMDE              | 59.1  | 91.8  | 0.703 | 41.8  | 81.7  | 0.551 |

\[ MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \]

We compute MRR on T-EMDE, SAF and SGR in order to differentiate their performance results better.

4.2 Performance Metrics

As our main performance measure we adapt the Recall@k (R@k) metric applied in [15]. Recall@k can be understood as the proportion of relevant items from the whole list of retrieved items. Since the performance of recent cross-modal retrieval models seems to be saturating and the results are often very similar, we also employ an additional performance measure: the mean reciprocal rank (MRR), a popular metric in information retrieval. MRR is the average of the reciprocals of the ranks of the relevant items from the whole query set Q:

4.3 Training configuration

For the MSCOCO dataset we set $N = 20$ and $K = 8$, and for the Flicker30K - $N = 16$ and $K = 8$, selected experimentally. The inner
The cases where T-EMDE gives non-SOTA results are usually at larger \( k \) in Recall@\( k \) (which consider items from lower recommendation list and determine the first user experience). The performance difference to the top-performing models described in the Related Work Section. We can observe that T-EMDE brings significant improvements in most metrics, especially the R@1 metric (up to 1.7 pp.), in many cases establishing new state-of-the-art results, which leads us to think that it facilitates text/image matching while providing very fine-grained representations. The performance gains can be especially important in practical scenarios, as the R@1 metric denotes the top retrieved items which appear at the start of user recommendation list and determine the first user experience.

We can observe that T-EMDE brings significant improvements in most metrics, especially the R@1 metric (up to 1.7 pp.), in many cases establishing new state-of-the-art results, which leads us to think that it facilitates text/image matching while providing very fine-grained representations. The performance gains can be especially important in practical scenarios, as the R@1 metric denotes the top retrieved items which appear at the start of user recommendation list and determine the first user experience.

We can observe that T-EMDE brings significant improvements in most metrics, especially the R@1 metric (up to 1.7 pp.), in many cases establishing new state-of-the-art results, which leads us to think that it facilitates text/image matching while providing very fine-grained representations. The performance gains can be especially important in practical scenarios, as the R@1 metric denotes the top retrieved items which appear at the start of user recommendation list and determine the first user experience.
5 ANALYSIS

5.1 Training and Inference Time
As shown in Table 3, T-EMDE makes both SAF and SGR faster by 15%-20% in every tested scenario in terms of inference time. This difference stems from reducing the quadratic complexity of the global similarity computed with self-attention to linear complexity of the trainable coder. Such performance difference can be significant in database retrieval scenarios, where low latency is of key importance. Likewise, training times are significantly reduced by a similar proportion as in the case of inference times. Note that the training speed difference does not stem from early stopping, as all benchmarked models were required to complete a defined number of epochs (20 epochs for MSCOCO, 30 epochs for Flickr30K SAF, 40 epochs for Flickr30K SGR).

Benchmarking was performed on a machine with 128 GB RAM, 14 core (28 HT threads) Intel Core i9-9940X 3.30GHz CPUs and GeForce RTX 2080 Ti 11GB GPU.

5.2 Optimal N/K Configuration
In Figure 5 and Table 4 we present R@1, R@10, and MRR performance scores for various $N/K$ parameter configurations of the T-EMDE/SAF model on the Flickr30K dataset. It can be observed that the T-EMDE sketch size can be indeed small, on the scale of $N \in \{8, 16, 20\}$ and $K = 8$. This is consistent with observations from [17]: too large $K$ results in the loss of the information prior (too many centroids are created and even related items are assigned to different centroids). On the other hand, $N$ can often be large as its function is to introduce an ensembling effect. Independent space partitionings can thus be thought of as individual ensemble models.

Overall, the resulting output sketch dimensionality can be dropped to $N = 8$ and $K = 8$, which produces a particularly small vector of size $8 \times 8 = 64$, which is beneficial for model latency. At the same time, the performance loss from a very small coder can be considered low especially in practical scenarios (see Table 4), which allows for further compression of the T-EMDE models.

5.3 Qualitative Analysis
Figure 4 shows 2D U-Map projections of token sketches obtained from T-EMDE, coming from individual space partitionings. The evaluated T-EMDE coder contained 8 centroids, whose locations are discernible in the pictures. Particularly visible are boundary lines composed of individual tokens, akin to Voronoi cell boundaries. This suggests that many tokens are purposefully placed at equal distances to multiple centroids.

In Figure 3 we display the results of a 2D projection of a single 8-centroid space partitioning, but with annotated locations of particular tokens. It can be seen that various token classes occupy
particular regions in space. For example, the words denoting sizes - big and small tend to group rather at the right bottom "legs" of the visualization. An interesting case are the function words (exemplified with there and these) which are common in captions but bring in little semantic value. Such words are clearly grouped in a single cluster, which seems purposeful so that they can be ignored by the rest of the network.

6 CONCLUSIONS

In this paper we have presented T-EMDE - a highly efficient module for computing global representations of text and image in multi-modal retrieval scenarios. By replacing a self-attentive module in recent high-performing architectures, we are able to reach new state-of-the-art results on MSCOCO and Flickr30k datasets, with especially significant gains in the Recall@1 metric. At the same time, we reduce the model latency by up to 20%. We propose T-EMDE as a step in the direction of replacing computationally expensive self-attention with more efficient approaches.

REFERENCES

[1] Peter Anderson, Xiaodong Ho, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society, 6077–6086. https://doi.org/10.1109/CVPR.2018.00836

[2] Dzmitry Bahdanau, Kyung Hoon Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, ICLR 2015 – Conference Track Proceedings. International Conference on Learning Representations, ICLR. arXiv:1409.0473v3 https://arxiv.org/abs/1409.0473v3

[3] Jiuyi Bai, Yuxiang Ren, and J. Zhang. 2020. Ripple Walk Training: A Subgraph-based training framework for Large and Deep Graph Neural Network. ArXiv abs/2002.07206 (2020).

[4] Irwan Bello. [n.d.]. LambdaNetworks: Modeling long-range Interactions without Attention. In International Conference on Learning Representations (ICLR) 2021.

[5] Is Belbagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. Technical report. arXiv:2004.05150 https://github.com/allenai/longformer

[6] Hui Chen, Guiguang Ding, Xudong Liu, Zipin Liu, Ji Liu, and Jiungha Han. 2020. IMRAM: Iterative Matching with Recurrent Attention Memory for Cross-Modal Image-Text Retrieval. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (march 2020), 12652–12660. https://doi.org/10.1109/CVPR2020.2020.01267 arXiv:2003.03772

[7] Jianfei Chen, Jun Zhu, and Le Song. [n.d.]. Stochastic Training of Graph Convolutional Networks with Variance Reduction. In Proceedings of the 35th International Conference on Machine Learning.

[8] Tianlang Chen and Jiebo Luo. 2020. Expressing objects just like words: Recurrent Approaches. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2020. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. arXiv:2006.16236 https://doi.org/10.1145/3394574.3399196

[9] Runghoo Chung, Yulho Park, Dae Sung Kim, and Rong Jin. 2019. Cross-modal Audio Search and Retrieval with Joint Embeddings Based on Text and Audio. In ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 4095–4099. https://doi.org/10.1109/ICASSP.2019.8682632

[10] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. VSE++: Improving visual-semantic embeddings with hard negatives. arXiv:1707.05612 http://arxiv.org/abs/1707.05612

[11] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc’ Aurelio Ranzato, and Tomas Mikolov. 2013. DeViSE: A Deep Visual-Semantic Embedding Model. In Advances in Neural Information Processing Systems, C. J. C. Burges, L. Bottou, M. Walling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2013/file/7ce6c95977442717202375c723c723-Paper.pdf

[12] Zhuan Guan, Kang Liu, Ma Yan, Xun Qian, and Tongkai Ji. 2018. Sequential Dual Attention: Coarse-to-Fine-Grained Hierarchical Generation for Image Captioning. Symmetry 10 (11 2018), 626. https://doi.org/10.3390/sym10110626

[13] MD. Zakir Hossain, Ferdous Sohel, Mohd Fauzul Shatruddin, and Hamid Laga. 2019. A Comprehensive Survey of Deep Learning for Image Captioning. ACM Comput. Surv. 51, 6, Article 118 (Feb 2019), 36 pages. https://doi.org/10.1145/3295748

[14] Anglos Karathopoulos, Apostov Vyas, Nikolos Pappas, and Francois Fleuret. 2020. Transformers are RNNS: Fast Autoregressive Transformers with Linear Attention. arXiv:2006.16236 https://doi.org/10.1145/3394574.3399196

[15] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Refformer: The Efficient Transformer. Technical report. arXiv:2001.04451 https://huggingface.co/2019/leaderboards/

[16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting Language and Vision Using Crowd-sourced Dense Image Annotations. Int. J. Comput. Vis. 123, 1 (2017), 32–73. https://doi.org/10.1007/s11263-016-0981-7

[17] Zhang Huang, Yiming Gao, Guanbin Li, Ping Luo, Yumin Chen, Liang Lin, and Wayne Zhang. 2019. Fashion Retrieval via Graph Reasoning Networks on a Similarity Pyramid. In 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. IEEE, 3066–3075. https://doi.org/10.1109/ICCV.2019.00316

[18] Kuang Hui Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked Cross Attention for Image-Text Matching. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11208 LNCS (mar 2018), 212–228. https://doi.org/10.1007/978-3-030-01225-0_13 arXiv:1803.08024
