TACKLING THE RETRIEVAL TRILEMMA WITH CROSS-MODAL INDEXING

Anonymous authors
Paper under double-blind review

ABSTRACT

Current cross-modal retrieval methods still struggle with the **retrieval trilemma** to simultaneously satisfy three key requirements, including high accuracy, fast speed, and low storage. For example, the *cross-modal embedding* methods usually suffer from either slow query speed caused by the time-consuming modality interaction or the tremendous memory cost of dense vector storage. While the *cross-modal hashing* methods are typically unsatisfied in accuracy due to the lossy discrete quantization for vector compression. In this paper, we tackle the retrieval trilemma with a new paradigm named *Cross-Modal Indexing* (CMI) that *directly* maps queries into identifiers of the final retrieved candidates. Specifically, we firstly pre-define sequential identifiers (SIDs) for all candidates into a hierarchical tree that maintains data semantically structures. Then we train an encoder-decoder network that maps queries into SIDs with the supervision of the constructed SIDs. Finally, we directly sample SIDs of relevant candidates for queries with $\mathcal{O}(1)$ time complexity. By evading the unfavorable modality interaction, dense vector storage, and vector compression, the proposed CMI reaches a satisfactory balance in the retrieval trilemma. For example, experiments demonstrate that CMI achieves comparable accuracy with about 1000x storage reduction and 120x speedup compared to the state-of-the-art methods on several popular image-text retrieval benchmarks.

1 INTRODUCTION

Cross-Modal Retrieval (Wang et al., 2016a; Cao et al., 2020; 2022) aims to retrieve data across different modalities, *e.g.*, taking an image as a query to retrieve the most relevant texts in the gallery. It fundamentally requires i) high retrieval accuracy, ii) fast query speed, and iii) low memory storage in real scenarios with large-scale galleries, *e.g.*, image search (Luo et al., 2003) on the Internet and product retrieval (Rubio et al., 2017) for E-commerce. Unfortunately, existing approaches still struggle with satisfying these requirements simultaneously and usually compromise among them. For example, the cross-modal embedding (CME) methods (Chen et al., 2020c; Wang et al., 2020) trade speed or storage for high accuracy, and the cross-modal hashing (CMH) methods (Faghri et al., 2018; Hu et al., 2022) achieve fast speed and low storage with decayed retrieval performance. Formally, we identify the challenge posed by these requirements as the cross-modal *retrieval trilemma*.

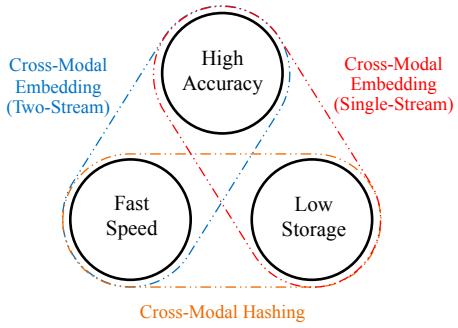


Figure 1: Cross-modal retrieval trilemma.

Figure 2 summarizes how current mainstream frameworks balance these three requirements. The cross-modal embedding (CME) paradigm follows the pipeline of 1) Feature Extraction for query and candidates across modalities, 2) Similarity Measurement between query and candidates embeddings, and 3) Brute-Force Search to acquire the final retrieval results. Specifically, the single-stream framework (Diao et al., 2021; Zhang et al., 2022) usually acquires high accuracy by performing fully

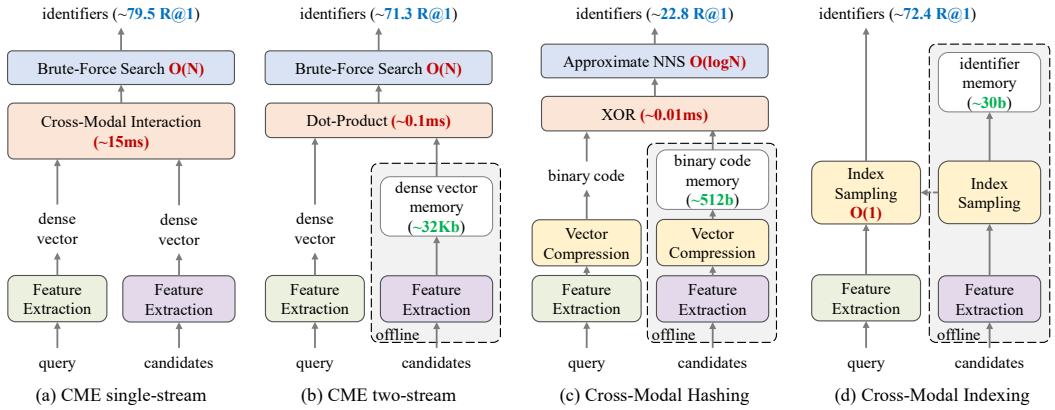


Figure 2: Comparison of cross-modal retrieval paradigms. **(a)** Single-stream framework relies on heavy modality interaction, leading to low query speed. **(b)** Two-stream framework stores dense vectors, suffering from huge memory costs. **(c)** Cross-modal hashing paradigm adopts vector compression, thus hurting the retrieval accuracy. **(d)** The proposed cross-modal indexing paradigm directly maps query into identifiers of relevant candidates, satisfying the retrieval trilemma. Blue, red, and green texts highlight the accuracy, speed, and storage performance, more details are in Section 4.

cross-modal interaction during the similarity measurement and zero memory cost since the computations are purely online. However, the heavy and online modality interaction also incurs unacceptable time consumption in practice. While, the two-stream framework (Chen et al., 2021; Wang et al., 2020) discards the cross-modal interaction with a dot product operation, as well as pre-computes and stores the candidate embeddings offline. With a light and offline similarity measurement, the two-stream framework speeds up online retrieval with the cost of memory storage. However, storing massive dense vectors still hinders practical applications. On the contrary, the cross-modal hashing (CMH) paradigm (Jiang & Li, 2017; Hu et al., 2022) follows the pipeline of 1) Feature Extraction, 2) Vector Compression that learns binary representations for query and candidates embeddings, and 3) Approximate Nearest Neighbor Search with low storage cost and fast retrieve speed. However, the vector compression with quantization error significantly hurts the retrieval accuracy. Even though they achieve promising performance on the coarse label-based retrieval, they usually fail in the fine-grained instance-level cross-modal matching.

In this paper, we aim to tackle the retrieval trilemma for all the above existing frameworks that failed. Inspired by the power of deep neural networks (Raghu et al., 2017; Lu & Lu, 2020), we propose to train a model that takes the query as input and then *directly* generates the identifiers of the relevant candidates. By avoiding the time-consuming cross-modal interaction, memory-costing dense vector storage, and accuracy-damaging vector compression, the proposed new paradigm named cross-modal indexing (CMI) successfully achieves a satisfactory trade-off with respect to the retrieval trilemma. We implement the CMI with sequential identifiers (SIDs) across modalities and the pipeline follows three steps: 1) pre-defines a unique SID for each data pair, 2) trains the indexing model with the data point-SID pairs offline, and 3) samples SIDs for each query as the final retrieval results online. Specifically, the SIDs are sequential and pre-defined by a hierarchical clustering tree according to their semantic embeddings, thus maintaining the semantic structure of SIDs, *i.e.*, semantically similar data points share SID prefixes. The indexing model is a multi-modal encoder-decoder network where the encoder extracts dense features for data points, and the decoder auto-regressively generates the sequential SIDs. The index sampling adopts a beam search strategy to enable top-k retrievals.

In summary, our contributions are threefold: 1) We propose a new paradigm named cross-modal indexing (CMI) that directly maps the query into identifiers of relevant candidates. 2) We realize the CMI with pre-defined sequential identifiers, the encoder-decoder indexing model, and the index sampling strategy with beam search. 3) Our experimental results on image-text retrieval benchmarks show that the proposed CMI achieves comparable retrieval performance with about a 1,000 compression ratio and 120 speedup ratio compared to current state-of-the-art methods.

2 BACKGROUND AND RELATED WORKS

In this section, we will introduce the background and recent advances of CME and CMH. Formally, the cross-modal retrieval task is to retrieve n relevant candidates from the gallery $\mathcal{C} = \{c_1, \cdots, c_N\}$ according to the query q, where c denotes the candidate and N is the size of the gallery.

2.1 CROSS-MODAL EMBEDDING

Background. The cross-modal embedding methods follow the pipeline of feature extraction, similarity measurement, and brute-force search, as illustrated in Figure 2. Formally, CME firstly extracts dense vectors of both query and candidates:

$$\boldsymbol{d_q} = \nu(q), \qquad \boldsymbol{d_c} = \varphi(c), \tag{1}$$

where $\nu(.)$ and $\varphi(.)$ are the mapping functions of query features and candidate features, respectively. Notably, for cross-modal retrieval, query q and candidate c are from different modalities thus $\nu(.)$ and $\varphi(.)$ are different networks, *e.g.*, ViT (Dosovitskiy et al., 2020) for images and BERT (Devlin et al., 2018) for texts.

After that CME adopts a similarity function $\mathrm{Sim}(.)$ to predict relevance score $s(q, c)$ for query q and each candidate c in the dense vector space, as,

$$s(q, c) = \mathrm{Sim}(\boldsymbol{d_q}, \boldsymbol{d_v}). \tag{2}$$

Most existing CME methods usually focus on designing a delicate similarity function. For example, the single-stream framework employs a heavy cross-modal interaction (*e.g.*, co-attention Li et al. (2017b) and graph neural network Liu et al. (2020a)) to strengthen the local similarities between two modalities, while the two-stream framework designs different distance computations (*e.g.*, Wasserstein Distance Wang et al. (2021) and Graph Optimal Transport Chen et al. (2020b)) to mitigate the semantic gap across the modality. CME methods are mainly trained by negative sampling and encourage the similarity of paired data maximum. Specifically, single-stream methods calculate similarity online thus the memory storage is 0, while two-stream methods pre-calculate the dense vectors $\boldsymbol{d_c} \in \mathbb{R}^D$ thus the **memory storage** is $32ND$ bits since each float consumes 32 bits, where D is usually 1,024 or 2,048 for most methods.

With the relevance scores, CME utilizes a brute-force search procedure to retrieve the final results, formally,

$$\text{results} = \mathrm{sort}(c \in \mathcal{C} \ \text{ based on } s(q, c))[:n]. \tag{3}$$

Since the sorting procedure has to be performed over all candidates in the gallery $\mathcal{C}$, the **time complexity** of CME is $\mathcal{O}(N)$.

Related works. Existing works on Cross-Modal Embedding fall into two categories: Single-stream models and Two-stream ones. Single-stream models (Lu et al., 2019; Chen et al., 2020c; Gan et al., 2020; Huang et al., 2020; Zhang et al., 2021) usually utilize cross-modal fusion modules like Transformer (Vaswani et al., 2017) layers to interact between image regions and text words and measure the similarity via model reasoning. Although sufficient interaction leads to superior accuracy performance, it suffers from huge computational costs and intolerable latency in real-world scenarios due to this online model reasoning fashion that matches a query with the whole gallery in a brute-force way in real time. To circumvent this shortcoming, two-stream models (Yan & Mikolajczyk, 2015; Wang et al., 2016b; Radford et al., 2021; Jia et al., 2021; Sun et al., 2021) mapping image and language to a joint embedding space where the embeddings can be pre-computed offline and the matching process can be accelerated via similarity calculation of dense vectors. However, the pre-computed dense vectors bring huge memory occupancy, and the linear time complexity is still unacceptable when facing massive data in the real world.

2.2 CROSS-MODAL HASHING

Background. The cross-modal hashing methods introduce a new perspective that maps the dense vectors into a discrete space with vector compression and performs an approximate nearest neighbor search (ANNS). This is essentially a kind of Production Quantization (PQ) (Jégou et al., 2011), a classical vector compression method for approximate nearest neighbor search. We next revisit this method concisely. PQ firstly splits a dense vector d into M sub-vectors.

Firstly, PQ defines M set of embeddings, each of which includes K centroid embeddings, denoted by $\boldsymbol{c}_{i,j} \in \mathbb{R}^{D/M}$ where $i \in [1, M]$ and $j \in [1, K]$. For each dense vector $\boldsymbol{d}$, PQ splits $\boldsymbol{d}$ into M sub-vectors as,

$$\boldsymbol{d} = \boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_M, \tag{4}$$

Then PQ quantizes each sub-vector $\boldsymbol{d}_i$ as the index of centroid embeddings where the quantization algorithm can be formulated as finding the closest PQ centroid embedding for $\boldsymbol{d}_i$ in the vector space,

$$\varrho_i(\boldsymbol{d}) = \arg\min_j \|c_{i,j} - \boldsymbol{d}_i\|^2, \tag{5}$$

Thus, the discrete representation of $\boldsymbol{d}$ is the concatenation of $\varrho_i(\boldsymbol{d})$,

$$\boldsymbol{d} \to \varrho(\boldsymbol{d}) = \varrho_1(\boldsymbol{d}), \varrho_2(\boldsymbol{d}), ..., \varrho_M(\boldsymbol{d}). \tag{6}$$

With the discrete code, the **memory storage** of CMH is NM, where M is usually 16/32/64 bits for NUS-WIDE, and 512 bits for Flickr30K.

With the discrete code $\varrho(\boldsymbol{d})$, CMH performs ANNS with a fast similarity function,

$$s^*(q, c) = \text{XOR}(\varrho(\boldsymbol{d_q}), \varrho(\boldsymbol{d_c})). \tag{7}$$

Similarly, the search procedure is,

$$\text{results} = \text{fast sort}(c \in \mathcal{C} \ \text{ based on } s^*(q, c))[: n] \tag{8}$$

Thanks to the fast hashing sorting algorithm, the **time complexity** is $\mathcal{O}(logN)$ for ANNS.

Related works. CMH is an option to cater to the demand for low storage cost and retrieval latency with Approximate Nearest Neighbor Search(ANNS). Prior CMH methods can be roughly divided into two groups: supervised and unsupervised. The supervised approaches (Li et al., 2017a; Deng et al., 2018; Liu et al., 2019; Hu et al., 2019; Liu et al., 2021) often learn the unified binary codes under the supervision of semantical labels, which is labor-intensive to gather a large quantity of annotated data for training. Moreover, the unicity of semantical labels prevents hash codes from representing rich semantics or accomplishing fine-grained retrieval tasks. As an alternative, the unsupervised methods (Liu et al., 2017; Zhang et al., 2018a; Li et al., 2019a; Hu et al., 2022) learn the hash codes by mapping features from multiple modalities into a common Hamming space via graph-based fusion (Liu et al., 2017), generative a/o adversarial mode (Zhang et al., 2018b;a; Li et al., 2018; 2019a; Bai et al., 2020) or contrastive learning (Li et al., 2020; Qiu et al., 2021; Hu et al., 2022). However, it's still hard for the binary code to carry enough semantic information for instance-level retrieval. In contrast, our SIDs contain hierarchical semantical information that fulfills fine-grained retrieval requirements.

3 Cross-Modal Indexing

Different from the CME and CMH, the proposed CMI directly maps the query q into the identifiers Γ_c corresponding to the relevant candidate c in the gallery $\mathcal{C}$, formally,

$$\text{result} = \text{lookup}(c \in \mathcal{C} \ \text{ based on } \Gamma_c), \ \text{ where } \Gamma_c \text{ sample from CMI}(q). \tag{9}$$

With the new paradigm CMI, the **memory storage** is NI bits where I is the size of identifiers which is usually about 30, and the **time complexity** is $\mathcal{O}(1)$ thanks to the lookup algorithm.

Next, we will elaborate the implementation of CMI by 1) define the identifiers Γ_c (Section 3.1), 2) train the cross-modal indexing model CMI (Section 3.2), and 3) design the sampling strategy(Section 3.3).

3.1 Index Construction

We represent the identifiers Γ as a sequence $\{\gamma_1, \cdots, \gamma_W\}$ with length as W. Without loss of generality, here we take image-text retrieval as an example.

Features extraction. Before constructing the clustering tree, we need to extract the image and text features. The linguistic or visual representations and the target sequence are closely related in semantics, thus the SIDs must represent correlative semantic information of the multi-modal data to

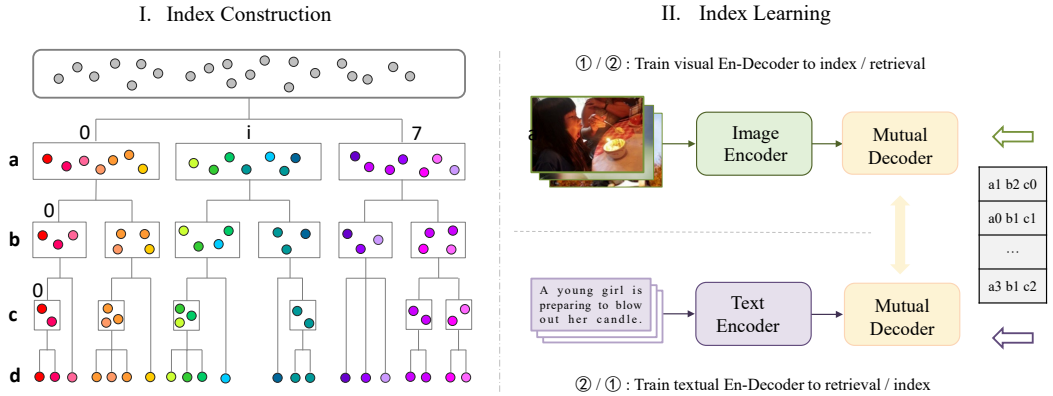


Figure 3: I. Index Construction pre-defines SIDs extraction from hierarchical clustering tree. II. Index Learning procedure with the encoder-decoder framework.

make the training of sequence mapping feasible. Here we introduce the hierarchical clustering tree algorithm to generate SIDs that implicate hierarchical semantics. Features extraction is crucial to the quality of the created sequential identifier. Considering the powerful generalization performance of CLIP (Radford et al., 2021), we extract image and text features with a pre-trained CLIP vision model and text model respectively. In practice, if a sample of one modality owns several matching pairs from another modality, all the embeddings of the sample and matching pairs will be extracted. Then we concatenate visual and textual features together into a group as the final representations of multi-modal data, which will be used to create the hierarchical clustering tree.

Recursive clustering algorithm. Given the embedding groups $\mathbb{G} = \{(v;t)^i\}$ of multi-modal data, we cluster them into 2^n clusters with K-means and number them from 0 to $2^n - 1$ in a recursive way until the number of samples in the clusters is no more than 2^m. Finally, we get a hierarchical clustering tree where the none-leaf nodes present the hierarchical clustering centers and leaf nodes are the embeddings of sample pairs. If we take the original embeddings as the first layer of the tree, we code from the second layer as a-z. Now we can create the sequential identifiers according to the path of every leaf node in the tree. For the none-leaf nodes in the path, we name them by composing their hierarchical layer and clustering number. And for the leaf nodes, we name them according to their similarity to the corresponding clustering center from 0 to $2^m - 1$. The final semantical identifier for each sample pair is the concatenation of all nodes' code from the path. Figure 3 (I) shows the visual illustration of this algorithm. And we formulate this process as:

$$\Gamma = \{\gamma_1, \cdots, \gamma_T\} \leftarrow \mathcal{R}(f_\theta(v;t)), \tag{10}$$

where T is the code length, $\mathcal{R}(.)$ is the recursive clustering function, $f_\theta(.)$ represents the CLIP encoder.

Then the identifiers are used as supervision of our model training. It serves as the transduction target of multimodal input in our encoder-decoder framework. The target vocabulary size of this discrete representation space is calculated as:

$$\mathcal{V} = (d-l)\cdot 2^n + l \cdot 2^m + 2 \tag{11}$$

where d is the depth of the hierarchical clustering tree and l is the number of the layer that contains leaf nodes, '2' represents two special tokens for marking the beginning and blank of the sequence (More detail can be found in Section 4.1). The letters (*i.e.,* a-z) provide hierarchical information in training yet will be removed in the pre-computed SID gallery. The memory occupancy of SID is $(d-1)\cdot n + m$.

3.2 Index Learning

Similar to CMH, CMI maps the dense vectors into a discrete semantic space, which is also a process of production quantization. The difference is we fulfill the quantization of dense vectors as an auto-regression decoding process:

$$\varrho_1(\boldsymbol{x}), \varrho_2(\boldsymbol{x}), ..., \varrho_M(\boldsymbol{x}) \leftarrow \mathcal{D}(\boldsymbol{x}), \tag{12}$$

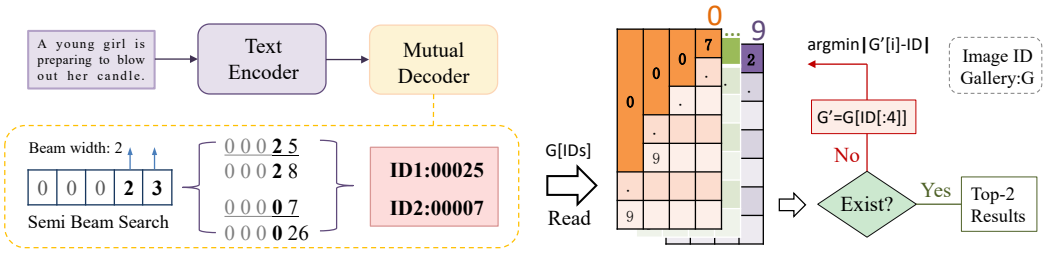


Figure 4: Illustration of Index Sampling (Taking text-to-image retrieval as an example). CMI gets top-K results in the decoding process via semi-beam search with the beam width K. For the sake of description, we set K=2 and the SID length as 5. Thus decoder searches at the last $\lfloor \frac{5}{2} \rfloor = 2$ codes.

where x represents the embeddings of input and $\mathcal{D}(.)$ is the decoder. The preliminary of PQ is non-differentiable while our encoder-decoder model can be optimized in an end-to-end fashion with the following seq2seq (Image or text tokens $\rightarrow$ ID) cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} \boldsymbol{y}_{i,j} \log \hat{\boldsymbol{y}_{i,j}} \tag{13}$$

where $\hat{\boldsymbol{y}_{i,j}}$ is the probability distribution of the j_{th} code for the i_{th} sample of the mini-batch (Size N). And $\boldsymbol{y}_{i,j}$ is the corresponding ground-truth distribution, which is obtained via the beforehand hierarchical clustering operation elaborated in the following Section 3.1. M is the discrete sequence (SID) length. Finally, we describe the retrieval pipeline via discrete sequence (ID). Taking the text-to-image retrieval task as an example, we first train the visual encoder-decoder model with the objective 13 to index the image gallery as the discrete sequences, then we finetune the textual encoder-decoder model with the objective 13 to generate IDs for textual queries. A correct retrieval means:

$$\mathcal{D}(\nu(\boldsymbol{t}_i)) = \mathcal{D}(\varphi(\boldsymbol{v}_i)) \tag{14}$$

As for image-to-text retrieval, vice versa.

Different from traditional cross-modal retrieval methods that usually have only one training goal, learning appropriate embeddings of two modalities and then pulling them as close as possible in the common subspace. As for CMI, it is a completely different paradigm containing two training goals, learning to index and learning to inference. To this end, we explored two modeling strategies. The first strategy is training separately in two independent models. A straightforward approach that one model for creating the sequential identifiers and the other for retrieval. We denote this framework as **Sep.** (*rep.* Separate models).

Considering the consistency of the output targets of the two models, we designed the second strategy that trains successively in a decoder-shared framework where the image and text encoders connect to a mutual decoder. For ease of description, we name the textual encoder-decoder as the T2id module, and similarly the image encoder-decoder as the I2id module. Taking the text-to-image retrieval task as an example, the training process is: I2id module $\rightarrow$ T2id module. Specifically, 1) we first train the I2id module and preserve the parameters to create the sequential identifiers for the image gallery. 2) Then we continue to use decoder parameters and train the T2id module for mapping text queries to the sequential identifiers. A visual illustration of this training process is shown in Figure 3 (II). Combining those two modules, we can accomplish the text-to-image retrieval task given ahead, which we call as **T2I** model. As for the image-to-text retrieval task, and vice versa, we reverse the training order and get the **I2T** model.

Note that both models can take on the bidirectional image-text retrieval tasks independently, although we design them for the two retrieval tasks separately. Experimental results are reported in Section 4.3 to illustrate their performances.

3.3 Index Sampling

Different from previous methods that rank a gallery of embeddings to get top-K results, CMI fulfills the matching and ranking operation in the decoding process, just like sampling from the latent space

of the learned model. Specifically, taking text-to-image retrieval as an example, we first put images into the visual encoder-decoder model to generate the ID gallery. Then a sentence query will be put into the textual encoder-decoder model. To obtain top-K predictions, we use semi-beam search in the sentence decoding process and a re-read strategy to improve the quantity of the predictions, which are introduced at length below. And a vivid illustration is shown in Figure 4.

Semi Beam Search. In the general seq-to-seq model like translation, beam search is a left-to-right truncated breadth-first search algorithm for the best prediction of the decoder at each time step. We express beam search as the following recursion:

$$\gamma_t = \underset{\hat{\gamma} \in \mathcal{B}_t}{\arg \operatorname{Topk}} p_\theta(\hat{\gamma} |\, x), t \in [1, .., T] \tag{15}$$

where x is the input, $\hat{\gamma}$ is the predicted code at the time step t and $\mathcal{B}_t$ is the code set at the time step t. And T is the whole time step (*i.e.*, code length). $p_\theta(\gamma|x)$ is the product of probability distributions over the output space whose size is defined in Equation 11.

However, the sequence of identifiers presents a coarse to fine semantical change following the hierarchical clustering tree. That means it is relatively easy to predict the prefix of a sequential identifier while not that easy in the suffix. In light of this observation, we propose the semi-beam search, a simple but efficient modification of beam search. We divide the decoder progress into two parts and apply beam search only in the suffix part, which we called Semi Beam Search. It can be formulated as:

$$\gamma_t = \underset{\hat{\gamma} \in \mathcal{B}_t}{\arg \operatorname{Topk}} p_\theta(\hat{\gamma} |\, x), t \in [\lceil \frac{T}{2} \rceil, .., T] \tag{16}$$

which means semi-beam search begins at the time step $\lceil \frac{T}{2} \rceil$ and only pays attention to the difficult and fine semantical code, which is applicable to our CMI.

Re-read. After getting IDs via the semi-beam search in decoding, we can read very quickly from the gallery by index. However, with some probability, we encounter a missing problem that query IDs are not existing in the gallery. As compensation, we re-read all the candidates in the partition (denoted as G') where the abortive ID is located and select the i_{th} candidate as an alternative that achieves the minimum difference:

$$\Gamma' = \underset{i}{\arg \min} |\mathrm{G}'[\,\mathrm{i}\,] - \Gamma| \tag{17}$$

An example of this process is given in Appendix 6.2.

4 Experiments

4.1 Experimental Settings

Please refer to Appendix 6.1.

4.2 Experimental Results

Accuracy, Speed, and Memory. We carry out bi-directional retrieval experiments on MS-COCO and Flickr30K. The comparison results with CME and CMH are shown in Table 1 and Table 4 (in Appendix 6.2). We can draw the following conclusions by analyzing: CME single-stream methods are leading in retrieval accuracy. CME two-stream methods occupy more than 32K bits for image and text embedding storage while our CMI only takes 30 bits for SIDs usage, which reduced memory consumption by at least 1000x. More importantly, CMI achieves better or comparable accuracy to CME two-stream methods except GPO pre-trained on 940M tagged images. What's more, CMI surpasses CMH on the three metrics with higher lookup speed, lower memory occupancy, and significant accuracy (*e.g.,* 50.4% upper in image-to-text top-1 retrieval results). Note that, CME single-stream methods matching via model reasoning online have no lookup (Speed) stage. We also have compared the inference (model reasoning + lookup) time of various methods.

Inference Time. Additionally, we test the once inference time of UNITER (Chen et al., 2020c) (*rep.* CME single-stream), GPO (*rep.* CME two-stream), DCMH (*rep.* CMH) and our CMI under different magnitude of candidates. The results in Figure 5 show that the time complexity of CMI

Method	Speed (ms)	Memory (bits)	Image → Text R@1	Image → Text R@5	Image → Text R@10	Text → Image R@1	Text → Image R@5	Text → Image R@10
CME Single-Stream								
BLIP (Li et al., 2022)	-	0	97.4	99.8	99.9	87.6	97.7	99.0
IMRAM (Chen et al., 2020a)	-	0	74.1	93.0	96.6	53.9	79.4	87.2
NCR (Huang, 2021)	-	0	77.3	94.0	97.5	59.6	84.4	89.9
SGRAF (Diao et al., 2021)	-	0	77.8	94.1	97.4	58.5	83.0	88.8
CMCAN (Zhang et al., 2022)	-	0	79.5	95.6	97.6	60.9	84.3	89.9
CME Two-Stream								
GPO (Chen et al., 2021)	∼25	32K	88.7	98.9	99.8	76.1	94.5	97.1
SCAN (Lee et al., 2018)	∼25	32K	67.4	90.3	95.8	48.6	77.7	85.2
VSRN (Li et al., 2019b)	∼50	64K	71.3	90.6	**96.0**	54.7	81.8	88.2
CVSE (Wang et al., 2020)	∼25	32K	70.5	88.0	92.7	54.7	**82.2**	88.6
CMH								
VSE++ (Faghri et al., 2018)	∼3	512	13.5	34.7	48.2	10.8	31.1	43.6
DJSRH (Su et al., 2019)	∼3	512	17.9	43.5	56.3	13.3	36.3	48.9
JDSH (Liu et al., 2020b)	∼3	512	13.6	35.6	49.4	9.8	29.1	42.6
UCCH (Hu et al., 2022)	∼3	512	22.8	48.1	61.0	16.9	41.8	54.9
CMI	∼0	30	**72.4**	**91.7**	94.2	**55.8**	81.5	**89.2**

BLIP and GPO are trained with large-scale datasets.

Table 1: Comparison of bi-directional retrieval results on Flickr30K 1K test set with Cross-modal Embedding (CME) and Cross-modal Hashing (CMH) methods.

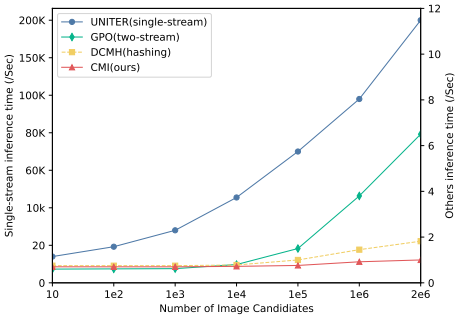


Figure 5: Comparison of inference time under different magnitude of candidates.

depth	length	Image-Text R@1	Image-Text R@10	Text-Image R@1	Text-Image R@10
2	6	71.6	93.1	53.7	88.4
3	6	**72.4**	**94.2**	**55.8**	**89.2**
4	6	72.0	93.8	55.1	88.8
3	5	71.8	93.8	53.5	88.9
3	6	**72.4**	**94.2**	**55.8**	**89.2**
3	7	70.3	93.2	53.8	88.0

Table 2: Ablation studies on the depth and length of SIDs with Flickr30K 1K test set.

is approximately sub-linear while CMH and CME two-stream methods are linear, which are more evident when the scale of the gallery is more than 10^5. What is more, the inference time of the CME single-stream is too long to compare with other methods under one scale axis.

Ranking and Lookup. We conduct bi-directional retrieval ranking experiments on NUS-WIDE. The comparison results with CMH are shown in Table 5 (in Appendix 6.2). CMI achieves better or comparable accuracy to CMH methods in three code lengths. We also report the mean of the image-to-text and text-to-image retrieval ranking results. CMI surpasses prior CMH methods, which demonstrate our balance performance in bi-directional retrieval. This may be because the SIDs of CMI can uniformly represent multi-modal instance pairs. And the hierarchical semantical help recalls similar candidates for the query. The P-R curves of lookup in CMH methods and our CMI are shown in Figure 6 (in Appendix 6.2), which also supports the conclusion mentioned above.

Qualitative results. Please refer to Appendix 6.2 for visualization results of bi-directional retrieval.

Framwork			Ranking		Image-Text		Text-Image	
Sep.	I2T	T2I	BS	**SBS**	R@1	R@10	R@1	R@10
✓			✓		69.3	86.5	53.6	80.4
✓				✓	69.3	89.4	53.6	83.1
	✓		✓		72.4	92.3	52.2	79.1
	✓			✓	**72.4**	**94.2**	52.2	81.2
		✓	✓		68.6	84.2	55.8	86.3
		✓		✓	68.6	88.7	**55.8**	**89.2**
	✓	✓	✓		72.4	92.3	55.8	86.3
	✓	✓		✓	72.4	94.2	55.8	89.2

Table 3: Ablation studies on framework and ranking strategy. 'Sep.' means two independent frameworks without shared decoder. I2T and T2I represent Image-to-Text and Text-to-Image framework respectively. 'BS' meas beam search while 'SBS' means semi beam search strategy. The ensemble results are marked as gray.

4.3 Ablation Study

Depth and Length of SIDs. We conduct ablation studies on the depth and length of SIDs. As Table 2 shows, the code style of SIDs has a limited impact on the accuracy. This is probably because SIDs is determined especially by the dataset so the memory size of SIDs is limited on it too.

Framework and Ranking Strategy. We first conduct ablation studies to analyze the individual impact of each model proposed in this paper and the effectiveness of the semi-beam search. The results are presented in Table 3. We note that the separate architecture model works pretty well with 72.4% for R@1 in sentence retrieval and 55.8% for R@1 in image retrieval. What's more, I2T and T2I achieve the comparable result on their respective retrieval task while the performance is relatively down on the opposite task, e.g using I2T model for the image retrieval task. Particularly for semi-beam search, it helps improve about 2% and 3% for R@1 in the sentence and image retrieval task respectively. It is profitable for the separate architecture model too, with an average 3% improvement on bi-directional retrieval tasks on R@10.

5 Conclusion

In this paper, we propose a new paradigm named cross-modal indexing (CMI) for the cross-modal retrieval trilemma to simultaneously satisfy high accuracy, fast speed, and low storage requirements. Compared to existing methods, the proposed CMI discard the unfavorable modality interaction, dense vector storage, and vector compression by directly mapping the query into the identifiers of relevant candidates. Specifically, we implement CMI with pre-defined sequential identifiers, encoder-decoder networks, and a beam search sampling strategy. By conducting extensive experiments on the most popular image-text benchmarks, we confirm that the proposed paradigm CMI reduces the time complexity from $\mathcal{O}(logN)$ to $\mathcal{O}(1)$ and substantially compresses the memory storage more than 1000x while performing favorably accuracy against the state-of-the-art methods. We hope we could raise the community's attention on the retrieval trilemma, and shed some light on future research in the new CMI paradigm.

Limitations. Even though the proposed CMI paradigm balances the retrieval trilemma with the lowest time complexity and the lowest memory storage, the accuracy somehow is unsatisfied compared to state-of-the-art methods, especially those pre-trained methods. Besides, the paper only conducts experiments on the most representative cross-modal retrieval across image and text domains, remaining video and audio modalities unexplored. In the future, we will further 1) exploit the potentialities of CMI and improve the accuracy performance, and 2) investigate more modalities of cross-modal retrieval with the proposed paradigm.

REFERENCES

Cong Bai, Chao Zeng, Qing Ma, Jinglin Zhang, and Shengyong Chen. Deep adversarial discrete hashing for cross-modal retrieval. *Proceedings of the 2020 International Conference on Multimedia Retrieval*, 2020.

Min Cao, Shiping Li, Juntao Li, Liqiang Nie, and Min Zhang. Image-text retrieval: A survey on recent research and development. *arXiv preprint arXiv:2203.14713*, 2022.

Wenming Cao, Wenshuo Feng, Qiubin Lin, Guitao Cao, and Zhihai He. A review of hashing methods for multimodal retrieval. *IEEE Access*, 8:15377–15391, 2020.

Hui Chen, Guiguang Ding, Xudong Liu, Zijia Lin, Ji Liu, and Jungong Han. Imram: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12652–12660, 2020a.

Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Chang Lian Wang. Learning the best pooling strategy for visual semantic embedding. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15784–15793, 2021.

Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*, pp. 1542–1553. PMLR, 2020b.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020c.

Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*, 2009.

Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio de Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8411–8420, 2021.

Cheng Deng, Zhaojia Chen, Xianglong Liu, Xinbo Gao, and Dacheng Tao. Triplet-based deep hashing network for cross-modal retrieval. *IEEE Transactions on Image Processing*, 27:3893–3903, 2018.

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.

Haiwen Diao, Ying Zhang, Lingyun Ma, and Huchuan Lu. Similarity reasoning and filtration for image-text matching. *AAAI*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *BMVC*, 2018.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *NIPS*, 2020.

Peng Hu, Xu Wang, Liangli Zhen, and Dezhong Peng. Separated variational hashing networks for cross-modal retrieval. *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

Peng Hu, Hongyuan Zhu, Jie Lin, Dezhong Peng, Yin-Ping Zhao, and Xi Peng. Unsupervised contrastive cross-modal hashing. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2022.

Zhenyu Huang. Learning with noisy correspondence for cross-modal matching. In *NeurIPS*, 2021.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *ArXiv*, abs/2004.00849, 2020.

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:117–128, 2011.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

Qing-Yuan Jiang and Wu-Jun Li. Deep cross-modal hashing. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3270–3278, 2017.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 201–216, 2018.

Chao Li, Cheng Deng, Ning Li, W. Liu, Xinbo Gao, and Dacheng Tao. Self-supervised adversarial hashing networks for cross-modal retrieval. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4242–4251, 2018.

Chao Li, Cheng Deng, Lei Wang, De Xie, and Xianglong Liu. Coupled cyclegan: Unsupervised hashing network for cross-modal retrieval. In *AAAI*, 2019a.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

Kai Li, Guo-Jun Qi, Jun Ye, and Kien A. Hua. Linear subspace ranking hashing for cross-modal retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1825–1838, 2017a.

Kunpeng Li, Yulun Zhang, K. Li, Yuanyuan Li, and Yun Raymond Fu. Visual semantic reasoning for image-text matching. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4653–4661, 2019b.

Shuang Li, Tong Xiao, Hongsheng Li, Wei Yang, and Xiaogang Wang. Identity-aware textual-visual matching with latent co-attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1890–1899, 2017b.

Yang Li, Yapeng Wang, Zhuang Miao, Jiabao Wang, and Rui Zhang. Contrastive self-supervised hashing with dual pseudo agreement. *IEEE Access*, 8:165034–165043, 2020.

Qiubin Lin, Wenming Cao, Zhiquan He, and Zhihai He. Mask cross-modal hashing networks. *IEEE Transactions on Multimedia*, 23:550–558, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Chunxiao Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang. Graph structured network for image-text matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10921–10930, 2020a.

Hong Liu, R. Ji, Yongjian Wu, Feiyue Huang, and Baochang Zhang. Cross-modality binary code learning via fusion similarity hashing. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6345–6353, 2017.

Song Liu, Shengsheng Qian, Yang Guan, Jiawei Zhan, and Long Ying. Joint-modal distribution-based similarity hashing for large-scale unsupervised deep cross-modal retrieval. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020b.

Xin Liu, Zhikai Hu, Haibin Ling, and Yiu ming Cheung. Mtfh: A matrix tri-factorization hashing framework for efficient cross-modal retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:964–981, 2021.

Xuanwu Liu, Guoxian Yu, Carlotta Domeniconi, Jun Wang, Yazhou Ren, and Maozu Guo. Ranking-based deep cross-modal hashing. In *AAAI*, 2019.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

Yulong Lu and Jianfeng Lu. A universal approximation theorem of deep neural networks for expressing probability distributions. *Advances in neural information processing systems*, 33:3094–3105, 2020.

Bo Luo, Xiaogang Wang, and Xiaoou Tang. World-wide-web-based image search engine using text and image content features. In *Internet Imaging IV*, volume 5018, pp. 123–130. International Society for Optics and Photonics, 2003.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

Zexuan Qiu, Qinliang Su, Zijing Ou, Jianxing Yu, and Changyou Chen. Unsupervised hashing with contrastive information bottleneck. In *IJCAI*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *international conference on machine learning*, pp. 2847–2854. PMLR, 2017.

Antonio Rubio, LongLong Yu, Edgar Simo-Serra, and Francesc Moreno-Noguer. Multi-modal joint embedding for fashion product retrieval. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 400–404. IEEE, 2017.

Shupeng Su, Zhisheng Zhong, and Chao Zhang. Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3027–3035, 2019.

Siqi Sun, Yen-Chun Chen, Linjie Li, Shuohang Wang, Yuwei Fang, and Jingjing Liu. Lightningdot: Pre-training visual-semantic embeddings for real-time image-text retrieval. *NAACL*, 2021.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

Haoran Wang, Ying Zhang, Zhong Ji, Yanwei Pang, and Lingyun Ma. Consensus-aware visual-semantic embedding for image-text matching. In *ECCV*, 2020.

Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. A comprehensive survey on cross-modal retrieval. *arXiv preprint arXiv:1607.06215*, 2016a.

Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5005–5013, 2016b.

Yun Wang, Tong Zhang, Xueya Zhang, Zhen Cui, Yuge Huang, Pengcheng Shen, Shaoxin Li, and Jian Yang. Wasserstein coupled graph learning for cross-modal retrieval. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1793–1802. IEEE, 2021.

Fei Yan and Krystian Mikolajczyk. Deep correlation for matching images and text. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3441–3450, 2015.

Huatian Zhang, Zhendong Mao, Kun Zhang, and Yongdong Zhang. Show your faith: Cross-modal confidence-aware network for image-text matching. In *AAAI*, 2022.

Jian Zhang, Yuxin Peng, and Mingkuan Yuan. Unsupervised generative adversarial cross-modal hashing. In *AAAI*, 2018a.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5575–5584, 2021.

Xi Zhang, Hanjiang Lai, and Jiashi Feng. Attention-aware deep adversarial hashing for cross-modal retrieval. In *ECCV*, 2018b.

6 APPENDIX

6.1 DETAILED EXPERIMENTAL SETTINGS

Datasets. CMI is evaluated on the three datasets: MS-COCO (Lin et al., 2014), Flickr30k (Plummer et al., 2015), NUS-WIDE (Chua et al., 2009). In MSCOCO and Flickr30k datasets, each image owns five annotated text descriptions, which we take as a multi-modal sample group. MSCOCO is a popular dataset for image-text matching and retrieval tasks, which contains 123,287 groups. In (Karpathy & Fei-Fei, 2015), it is split into 82,783 groups as the training set, 5000 validation set and 5000 test set. We follow the data split of (Faghri et al., 2018) to add 30,504 groups into the training set, which are originally abandoned in the validation set. Flickr30k contains 31,000 groups, which we split into 29,000 groups for the training set, 1000 validation set and 1000 test as (Karpathy & Fei-Fei, 2015; Faghri et al., 2018). Particularly for MSCOCO, the result of MSCOCO 1K is the average over five fold on the test set while MSCOCO 5K and Flickr30k are reported on the full test set. NUS-WIDE consists of 269,498 web images where each image corresponds to some text description of 81 concept categories. We select 186,557 image-text pairs belonging to 10 most frequent classes in our experiments.

Evaluation metrics. General cross-modal retrieval methods use the metric of recall at K (R@K, K=1,5,10). R@K is the percentage of correct matching in the top-K candidates. For a fair comparison, we obtain the top-K candidates using semi beam search with the beam width K. In addition, we introduce *Speed* and *Memory* in cross-modal retrieval task. *Memory* represent the memory occupancy of pre-computed representation, *i.e.*, dense vector in CME two-stream methods, hashing code in CMH, SID in our CMI. And *Speed* indicate the lookup time in the prepared gallery (1000 candidates in Flickr30k and 5000 in MSCOCO), *i.e.*, doc-product of vectors, ANN then XOR of hashing codes, index then ANN of SIDs, and no model reasoning time included here.

To compare with CMH methods thoroughly, we also utilize the widely-used mean Average Precision (mAP) and precision-recall (P-R) curve. mAP is the mean value of Average Precision (AP) scores for each query to measure the accuracy of the ranking results. P-R curve can measure the accuracy of the lookup. In CMH baselines, mAP is reported on all recall results. Semi beam search is limited here. Thus we sort the output logits of the last layer's left token and obtained the corresponding SIDs as CMI ranking results.

Implementation Details. We experiment with all Transformer (Vaswani et al., 2017) architecture. For the image encoder, we use the popular ViT (Dosovitskiy et al., 2021) pre-trained by CLIP (Radford et al., 2021). We closely follow CLIP implementations by adding an additional layer normalization before the transformer. For the text encoder, we naturally use the pre-trained base size Transformer in CLIP with 12 layers in depth, 512 for width and 8 attention heads. We closely follow CLIP tokenization method and bracket the text sequence with $[SOS]$ and $[EOS]$ tokens. Corresponding to $[EOS]$ token, the activations of the last layer in the transformer are taken as the representation of the input sequence. The mutual decoder is a light Transformer decoder with 2 layers in depth, 512 for width and 2 attention heads.

We set n=$\{2, 3, 4\}$ and m=6 to create the hierarchical clustering tree, which corresponds to 16bits, 32bits and 64bits of hashing code in CMH methods respectively. Except for special instructions, we set n=3, m=6 as default in the experiments. We add $[SOS]$ at the beginning of all identifiers and $[SP]$ at the blank place of short identifiers like the padding operation.

We implement the proposed method using PyTorch, and conduct the training and evaluation processes on two NVIDIA RTX 3090 GPU with 24 GB memory each. In all experiments, our model is optimized by Adam, and batch-size is set to 64. In addition to semi-beam search, we obtain predictions from the hidden state of the last layer for ranking results. The training progress is presented in Section 3.1. We start training the shared decoder with a learning rate 2e-4 for the first 5 epochs and then decay the learning rate by 0.1 for the rest of 10 epochs. After that, we keep the learning rate to finetune the image/text encoder-decoder model for 10 epochs and finetune the text/image encoder-decoder model for 5 epochs at last. As for the separate architecture model, we take the same training strategy. We choose the snapshot of the best performance on the validation for testing.

Table 4: Comparison of bi-directional retrieval results MSCOCO 5K test set with Cross-modal Embedding (CME) methods.

Method	**Speed** (ms)	**Memory** (bits)	**Image → Text** R@1	R@5	R@10	**Text → Image** R@1	R@5	R@10
CME Single-Stream								
BLIP (Li et al., 2022)	-	0	82.4	95.4	97.9	65.1	86.3	91.8
IMRAM (Chen et al., 2020a)	-	0	53.7	83.2	91.0	39.7	69.1	79.8
NCR (Huang, 2021)	-	0	58.2	84.2	91.5	41.7	71.0	81.3
SGRAF (Diao et al., 2021)	-	0	57.8	-	91.6	41.9	-	81.3
CMCAN (Zhang et al., 2022)	-	0	61.5	-	92.9	44.0	-	82.6
CME Two-Stream								
GPO (Chen et al., 2021)	∼120	32K	68.1	90.2	95.2	52.7	80.2	88.3
SCAN (Lee et al., 2018)	∼120	32K	50.4	**82.2**	90.0	38.6	69.3	80.4
VSRN (Li et al., 2019b)	∼260	64K	**53.0**	81.1	89.4	**40.5**	70.6	**81.1**
PCME (Chun et al., 2021)	∼120	32K	44.2	73.8	83.6	31.9	62.1	74.5
CMI	∼1	30	51.8	81.4	**90.6**	39.3	**71.5**	80.7

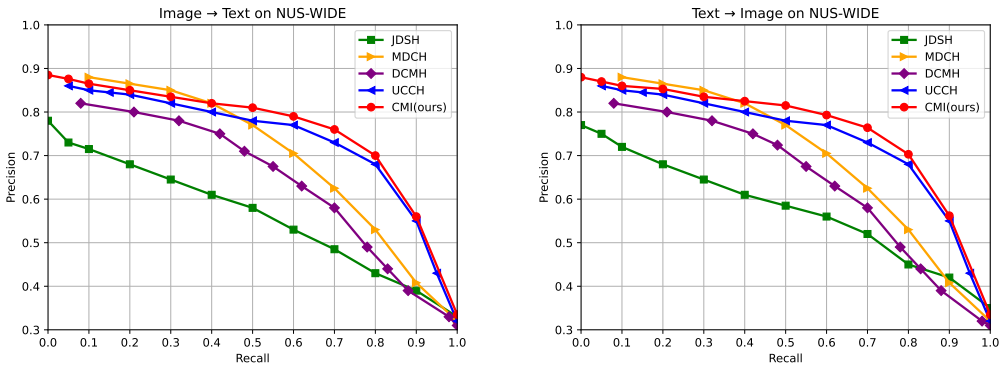
BLIP and GPO are trained with large-scale datasets.

Table 5: Comparison of bi-directional retrieval ranking mAP on NUS-WIDE with Cross-modal Hashing (CMH) methods. Namely, DCMH (Jiang & Li, 2017), JDSH (Liu et al., 2020b), MDCH (Lin et al., 2021), UCCH (Hu et al., 2022).

Method	16 bits			32 bits			64 bits		
	I-T	T-I	Mean	I-T	T-I	Mean	I-T	T-I	Mean
DCMH	0.5903	0.6389	0.6146	0.6031	0.6511	0.6271	0.6093	0.6571	0.6332
JDSH	0.6470	0.6490	0.6480	0.6560	0.6690	0.6625	0.6790	0.6890	0.6840
MDCH	0.6920	0.6654	0.6787	0.6994	0.6822	0.6908	0.7072	0.6915	0.6994
UCCH	0.6980	0.7010	0.6995	0.7080	0.7240	0.7160	**0.7370**	**0.7450**	**0.7410**
CMI(ours)	**0.7125**	**0.7267**	**0.7196**	**0.7218**	**0.7346**	**0.7282**	0.7285	0.7350	0.7315

6.2 Additional Experiments

Qualitative Results. Here we provide some image-to-text retrieval visual results in Figure 7 and text-to-image retrieval visual results in Figure 8, including correct and incorrect ones. We can find that our proposed SIDs are semantic relevant, and semi-beam search can provide multiple retrieval results. Even for the incorrect results, the generated SIDs are still similar to the correct ones, which demonstrates the effectiveness of our CMI framework.



(a) P-R curves of text retrieval.

(b) P-R curves of image retrieval.

Figure 6: Precision-Recall curves on NUS-WIDE. The code length of UCCH is 128 and others@64bits.

Figure 7: Examples of visual results in image to text retrieval task. A correct retrieval by the generated SID is noted with red. All result is reported by semi beam search with width 2. The first one is the top-1 result and the four row followed are the rest result in Top-5.

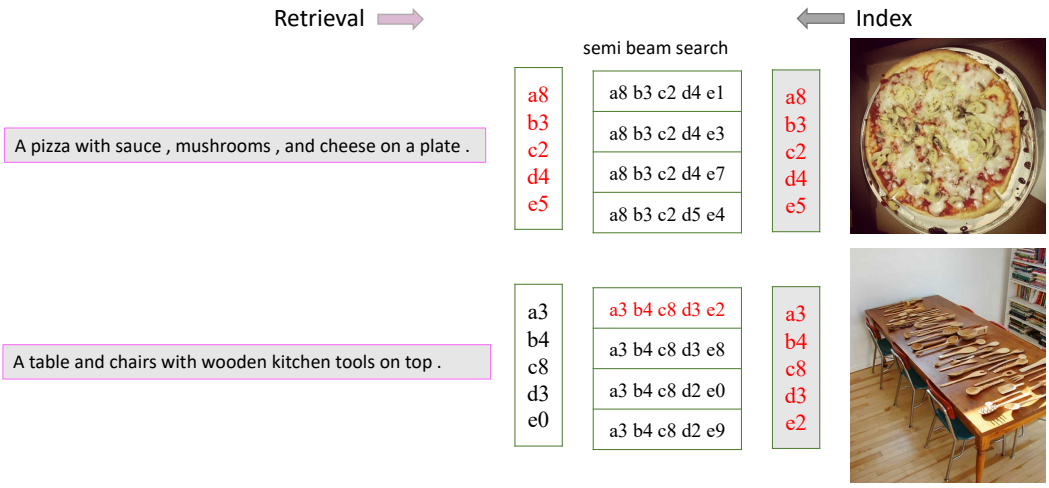


Figure 8: Examples of visual results in text-to-image retrieval task. A correct retrieval by the generated SID is noted with red. All result is reported by semi beam search with width 2. The first one is the top-1 result and the four row followed are the rest result in Top-5.

Re-read example. In Equation 17, only the leaf node layer that codes with embedding distance (Section 3.1) participates in this calculation. For example, the abortive ID is 000226 and the partition to which it belongs is G'=[0002a, ..., 0002z]. In Equation 17, $0002a - 000226$ means $a - 26$. Suppose the i_{th} candidate 000225 achieves the minimum of Equation 17, 000225 will be the prediction. Note that, each partition is very small and easy to access in the hierarchical and partitioned gallery, which has little impact on CMI sampling speed.