

# GENIUS: A Generative Framework for Universal Multimodal Search

## —Supplementary Material—

Sungyeon Kim<sup>1,2</sup> Xinliang Zhu<sup>1</sup> Xiaofan Lin<sup>1</sup> Muhammet Bastan<sup>1</sup> Douglas Gray<sup>1</sup> Suha Kwak<sup>2</sup>

<sup>1</sup> Amazon

<sup>2</sup> POSTECH

{sungyeon.kim, suha.kwak}@postech.ac.kr {xlzhu, xiaofanl, mbastan, dougray}@amazon.com

### A. Details of M-BEIR Dataset

The M-BEIR dataset [12] combines 10 datasets to support multimodal retrieval tasks, covering diverse domains such as image-caption retrieval, product search, news, and complex multimodal queries. As summarized in Table 1, it encompasses a total of 5.6M candidates. It supports eight distinct retrieval tasks, including retrieving images from text, text from images, and matching multimodal queries with corresponding multimodal responses. The dataset spans queries with varying levels of complexity, covering multiple domains such as fashion, news, and general-purpose data.

Each query instance consists of a query  $q$ , a set of related positive candidates  $c^+$ , and unrelated negative candidates  $c^-$ . To clarify the user’s intention, each query is paired with an additional intent description. All queries include at least one positive candidate while including negative candidates is optional.

**VisualNews.** The VisualNews dataset [8] was curated by randomly sampling 200K, 40K, and 40K image-caption pairs for training, validation, and testing, respectively. Tasks include retrieving captions ( $q_i \rightarrow c_t$ ) for a given image and retrieving images ( $q_t \rightarrow c_i$ ) for a given caption. The initial number of candidates of 2.5M entries was reduced to 1M in the M-BEIR dataset, consisting of 500K text and 500K image candidates.

**Fashion200K.** The Fashion200K dataset [4], comprising 200K images and 60K descriptions, was curated by selecting 30K image-description pairs for training. Tasks include retrieving product descriptions ( $q_i \rightarrow c_t$ ) for a given image and retrieving images ( $q_t \rightarrow c_i$ ) for a given product description. The number of candidates is 260K.

**COCO.** Using the Karpathy split [7], COCO data was converted to support tasks such as retrieving captions ( $q_i \rightarrow c_t$ ) from images and retrieving images ( $q_t \rightarrow c_i$ ) from captions. The dataset includes 113K training instances for image-to-caption retrieval, which was trimmed to 100K in the M-BEIR dataset for efficiency. The number of candidates for testing includes 25K text entries and 5K images, the same as the original test set of COCO.

**WebQA.** The WebQA dataset [1] links textual questions to images and their corresponding textual answers. Tasks include retrieving answers ( $q_i \rightarrow c_t$ ) based on questions and matching queries ( $q_i \rightarrow (c_i, c_t)$ ) with both images and textual explanations. The number of candidates comprises 400K image-text pairs and 540K text-only candidates.

**EDIS.** The EDIS dataset [9] connects captions to image-headline pairs. Tasks involve matching textual queries ( $q_i \rightarrow (c_i, c_t)$ ) with multimodal pairs consisting of images and their associated text. The number of candidates includes 1M image-headline pairs, and the training set consists of 26K instances.

**NIGHTS.** The NIGHTS dataset [3] pairs reference images with target images. The task focuses on retrieving images ( $q_i \rightarrow c_i$ ) based on a reference image. The dataset contains 16K, 2K, and 2K instances for training, validation, and testing, with a number of candidates of 40K images.

**FashionIQ.** FashionIQ [13] connects reference images and their textual descriptions to target images. Tasks include retrieving target images ( $q_i \rightarrow c_i$ ) based on reference images and associated descriptions. The dataset includes all images as the number of candidates, with 1.7K instances reserved for validation.

**CIRR.** CIRR [10] matches reference images and textual modifications to target images. The task involves retrieving target images ( $((q_i, q_t) \rightarrow c_i)$ ) that align with both the reference image and the specified textual modification. The number of candidates comprises all images, with validation and test sets derived from the dataset splits.

**OVEN.** The OVEN dataset [5] pairs images with text questions and their corresponding multimodal answers. Tasks include retrieving textual descriptions ( $((q_i, q_t) \rightarrow c_t)$ ) for a given query and matching multimodal responses ( $((q_i, q_t) \rightarrow (c_i, c_t))$ ). The dataset originally contained 6M candidates, which were reduced to a 1M number of candidates in the M-BEIR dataset, and training data was trimmed to 120K instances.

**InfoSeek.** InfoSeek [2] uses queries consisting of images and related questions paired with textual answers segmented into snippets. Tasks include retrieving text snippets

Task (query $\rightarrow$ candidate)	Dataset	Domain	# Query			# Rel./Query			# Candid.
			Train	Val	Test	Train	Val	Test	
1. $q_t \rightarrow c_i$	VisualNews [8]	News	99K	20K	20K	1.0	1.0	1.0	542K
	MSCOCO [7]	Misc.	100K	24.8K	24.8K	1.0	1.0	1.0	5K
	Fashion200K [4]	Fashion	15K	1.7K	1.7K	3.3	3.1	2.8	201K
2. $q_t \rightarrow c_t$	WebQA [1]	Wiki	16K	1.7K	2.4K	2.0	2.0	2.0	544K
3. $q_t \rightarrow (c_i, c_t)$	EDIS [9]	News	26K	3.2K	3.2K	2.6	2.6	2.6	1M
	WebQA [1]	Wiki	17K	1.7K	2.5K	1.4	1.4	1.4	403K
4. $q_i \rightarrow c_t$	VisualNews [8]	News	100K	20K	20K	1.0	1.0	1.0	537K
	MSCOCO [7]	Misc.	113K	5K	5K	5.0	5.0	5.0	25K
	Fashion200K [4]	Fashion	15K	4.8K	4.8K	1.0	1.0	1.0	61K
5. $q_i \rightarrow c_i$	NIGHTS [3]	Misc.	16K	2K	2K	1.0	1.0	1.0	40K
6. $(q_i, q_t) \rightarrow c_t$	OVEN [5]	Wiki	150K	50K	50K	8.5	10.0	9.9	676K
	InfoSeek [2]	Wiki	141K	11K	11K	6.8	6.7	6.5	611K
7. $(q_i, q_t) \rightarrow c_i$	FashionIQ [13]	Fashion	16K	2K	6K	1.0	1.0	1.0	74K
	CIRR [10]	Misc.	26K	2K	4K	1.0	1.0	1.0	21K
8. $(q_i, q_t) \rightarrow (c_i, c_t)$	OVEN [5]	Wiki	157K	14.7K	14.7K	17.8	17.5	17.7	335K
	InfoSeek [2]	Wiki	143K	17.6K	17.6K	9.1	7.5	7.5	481K
M-BEIR [12]		4 domains	1.1M	182K	190K	6.5	5.9	5.7	5.6M

Table 1. Summary of statistics of M-BEIR. Each row describes a task-specific retrieval setup, including the dataset, domain, the number of queries across Train/Validation/Test splits (# Query), the average number of relevant labels per query (# Rel./Query), and the total number of candidates (# Candid.).

$((q_i, q_t) \rightarrow c_t)$  and matching multimodal pairs  $((q_i, q_t) \rightarrow (c_i, c_t))$  with relevant queries. The processed dataset includes 140K instances each for text and multimodal retrieval tasks, with the number of candidates reduced to 1M in the M-BEIR dataset.

## B. Further Analysis

### B.1. Storage Efficiency Comparison

Efficient storage utilization is crucial for large-scale retrieval systems. Table 2 compares the per-data storage requirements of CLIP and GENIUS, highlighting the significant advantage of quantized representations.

CLIP, which operates on a 768-dimensional floating-point embedding, requires approximately 3 KB per data point when stored in 32-bit precision. This can lead to substantial storage costs, particularly in large-scale retrieval scenarios. In contrast, GENIUS leverages a compact quantization scheme, encoding each data point using a 2-bit code (for modality separation) and eight 12-bit codes selected from a  $2^{12}$ -sized codebook. This results in a total storage requirement of only  $2 + (8 \times 12) = 98$  bits, equivalent to 12.25 bytes per data point, which is over a 99% reduction compared to CLIP. For example, indexing one million data points would require around 3 GB with CLIP, whereas GENIUS would require only 12 MB. This drastic reduction in storage overhead makes GENIUS highly scalable and cost-efficient for deployment in real-world retrieval applications, especially those handling billions of data points.

### B.2. Training Efficiency

GENIUS offers high training efficiency. When training on 1.1 million samples using 4×RTX3090 GPUs, the CLIP encoder requires 91 hours. In comparison, GENIUS introduces an additional 0.4 hours for quantization and 2 hours for decoder training. As a result, on a per-sample basis, GENIUS is approximately 2.8 times more efficient than GRACE, which, according to reports, trains on 0.1 million samples in 24 hours for the MS-COCO dataset.

## C. Additional Experiments

### C.1. Impact of Contrastive Loss in Quantization

As shown in Table 3 of the main paper,  $\mathcal{L}_{cl}$  plays a crucial role, and its removal from the training of quantization (Eq. 8) leads to near-zero performance. To analyze how contrastive learning affects the embedding space, we conduct a UMAP visualization of the quantized feature  $\hat{z}$  before and after applying contrastive learning  $\mathcal{L}_{cl}$  (Eq. 3). Note that the quantized feature  $\hat{z}$  is the reconstructed feature using code embeddings derived from discrete IDs.

Fig. 1 illustrates that even though residual quantization loss (Eq. 7) is applied, removing contrastive learning results in misalignment between query and target features and causes target features to collapse. This degradation in representation leads to discrete IDs that fail to capture the relations between queries and targets effectively, making it difficult for the decoder to learn it. Furthermore, an excessive number of targets become mapped to a single ID, rendering the retrieval process ineffective and generating semantically

Model	Representation Format	Storage Cost per Data
CLIP [11]	768-dim floating-point vector (32-bit)	$768 \times 32 = 24,576 \text{ bits} = 3,072 \text{ bytes} \approx 3 \text{ KB}$
GENIUS	Quantized codes: 1 modality code (2-bit) + 8 semantic codes (12-bit each)	$2 + (8 \times 12) = 98 \text{ bits} \approx 12.25 \text{ bytes} (\sim 0.012 \text{ KB})$

Table 2. Comparison of storage efficiency between CLIP and GENIUS. GENIUS achieves a more than 99% reduction in storage requirements, significantly enhancing scalability for large-scale retrieval tasks.

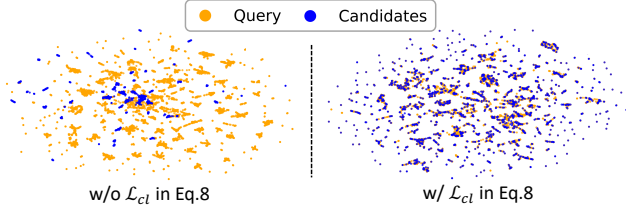


Figure 1. UMAP visualization of the quantized feature  $\hat{\mathbf{z}}$  before and after contrastive learning  $\mathcal{L}_{cl}$  of Eq. 3

inconsistent IDs.

In contrast, when contrastive loss is applied in Eq. 8, query-target alignment is preserved despite quantization. This ensures that the semantic information is well-represented within the discrete IDs. As a result, when training the decoder to map queries to targets, it can effectively capture the underlying relations, allowing it to generate meaningful discrete target IDs from queries.

## C.2. Impact of Modality Encoding

We analyze the impact of modality encoding by comparing different quantization strategies in Table 3: modality-decoupled quantization, classifier-based modality encoding, and residual quantization without a modality code.

Modality-decoupled quantization achieves the best performance among the three approaches. While classifier-based encoding successfully differentiates modalities, it does not integrate modality information within the quantization process. As a result, modality and semantic information are mixed within the discrete codes, limiting their representational capacity. In contrast, modality-decoupled quantization explicitly separates modality information by assigning the first code to modality while using the remaining codes for semantics, leading to a more structured and expressive representation.

The baseline without modality encoding, which does not explicitly separate modalities, further demonstrates that failing to encode modality weakens retrieval performance. These findings emphasize that modality-decoupled quantization provides a unified approach for handling multiple modalities in generative retrieval, offering a more effective discrete ID representation.

Method	COCO		WebQA		CIRR
	T2I	I2T	T2T	T2(I,T)	(I,T)2I
Modality-decoupled quantization	<b>55.4</b>	<b>82.7</b>	<b>28.3</b>	<b>47.1</b>	<b>20.5</b>
Classifier-based modality encoding	48.9	79.2	25.7	37.5	20.3
RQ w/o modality-code	20.2	73.2	25.9	34.3	18.3

Table 3. Ablation study on modality encoding approach (universal retrieval, R@5).

## C.3. Impact of Beam Search

We conduct an ablation study to examine the impact of beam size on retrieval performance and efficiency across various tasks. As shown in Table 4, increasing the beam size significantly improves Recall@5. For instance, on the COCO dataset for text-to-image retrieval, Recall@5 increases from 24.2% at a beam size of 1 to 68.2% at a beam size of 50. Similar trends are observed for image-to-text retrieval on COCO and image-to-image retrieval on CIRR. The improvement is even more pronounced on the WebQA dataset, which contains knowledge-intensive data in Wikipedia based on long sentence queries. Recall@5 for text-to-text retrieval increases from 5.1% at a beam size of 1 to 32.8% at a beam size of 50. This substantial gain is attributed to the expanded search space provided by larger beam sizes, allowing the model to handle better the complexity and richness of the queries in WebQA.

However, larger beam sizes increase the computational load, resulting in higher latency. Based on our measurements of the text-to-image retrieval task, retrieval speed decreases from 19.6 queries per second at a beam size of 30 to 11.9 queries per second at a beam size of 50. This trade-off between performance and efficiency is a fundamental consideration when deploying generative models using beam search. Selecting an appropriate beam size requires balancing the need for higher recall against the constraints of computational resources and application-specific latency requirements.

## C.4. Impact of Decoder Size

We analyze the effect of the decoder size on retrieval performance. Table 5 presents the results using T5 decoders of varying sizes: T5-small (30M parameters), T5-base (110M parameters), and T5-large (400M parameters). Increasing the decoder size generally enhances performance on tasks like COCO and WebQA. On COCO text-to-image retrieval,

Beam Size	COCO		WebQA		CIRR
	T2I	I2T	T2T	T2(I,T)	(I,T)2I
1	24.2	41.6	5.1	10.4	4.9
5	55.6	79.1	15.9	32.3	18.0
10	62.8	82.8	22.4	40.0	20.4
20	66.5	<b>83.7</b>	28.3	45.1	<b>21.1</b>
30	65.3	83.4	28.8	47.4	21.0
50	<b>68.2</b>	83.3	<b>32.8</b>	<b>50.0</b>	21.0

Table 4. Ablation over beam size (task-specific information retrieval, R@5). The default setting of our method is highlighted in grey box .

Decoder	# Params	COCO		WebQA		CIRR
		T2I	I2T	T2T	T2(I,T)	(I,T)2I
T5-small	30M	65.3	83.4	28.8	47.4	<b>21.0</b>
T5-base	110M	<b>67.9</b>	<b>83.5</b>	31.6	48.0	18.3
T5-large	400M	67.2	83.2	<b>32.4</b>	<b>50.4</b>	7.1

Table 5. Ablation over decoder size (task-specific information retrieval, R@5). The default setting of our method is highlighted in grey box .

$K \times M$	COCO		WebQA		CIRR
	T2I	I2T	T2T	T2(I,T)	(I,T)2I
$4096 \times 9$	<b>65.3</b>	<b>83.4</b>	28.8	<b>47.4</b>	21.0
$8192 \times 17$	59.5	81.8	<b>30.6</b>	44.8	<b>26.5</b>
$4096 \times 9$ (Shared)	18.6	19.3	0.2	1.7	3.3

Table 6. Ablation over codebook size  $K$  (except for the first level) and code level  $M$  (task-specific information retrieval, R@5). The default codebook size and level are underlined. In the shared configuration, codebooks are shared across all levels except the first. The default setting of our method is highlighted in grey box .

Recall@5 improves from 65.3% with T5-small to 67.9% with T5-base. On WebQA, performance increases consistently with decoder size, reaching 32.4% Recall@5 with T5-large, which is beneficial for handling complex sentences in WebQA. However, on the CIRR dataset, which involves complex relational reasoning in image-to-image retrieval, performance declines slightly with T5-base and drops sharply to 7.1% with T5-large. This suggests that larger models may overfit or struggle with optimization on certain tasks, especially those that do not benefit from increased model capacity. Therefore, we adopt T5-small as the default decoder for its effective trade-off between retrieval performance and computational efficiency.

### C.5. Further Analysis of Codebook Configuration

We further investigate the impact of codebook configurations, including codebook size ( $K$ ), code levels ( $M$ ) and shared codebook usage across levels in our modality-decoupled semantic quantization. Table 6 shows the results for different configurations. Increasing the codebook size and the number of code levels to  $K = 8192$ ,  $M = 17$

Method	Training Data	R@1	R@5	R@10
<b>Flickr30K</b>				
GRACE [6] (Numeric ID)	Flickr30K	22.5	28.9	29.4
GRACE [6] (String ID)	Flickr30K	30.5	39.0	40.4
GRACE [6] (Semantic ID)	Flickr30K	22.9	34.9	37.4
GRACE [6] (Structured ID)	Flickr30K	37.4	59.5	66.2
IRGen [14]	Flickr30K	49.0	68.9	72.5
<b>GENIUS</b>	M-BEIR	51.5 <sup>†</sup>	74.6 <sup>†</sup>	80.3 <sup>†</sup>
<b>GENIUS<sup>R</sup></b>	M-BEIR	63.7 <sup>†</sup>	80.4 <sup>†</sup>	83.2 <sup>†</sup>
<b>GENIUS</b>	Flickr30K	60.6	84.0	90.5
<b>GENIUS<sup>R</sup></b>	Flickr30K	<b>74.1</b>	<b>92.0</b>	<b>94.8</b>
<b>COCO</b>				
GRACE [6] (Numeric ID)	COCO	0.03	0.14	0.28
GRACE [6] (String ID)	COCO	0.12	0.37	0.88
GRACE [6] (Semantic ID)	COCO	13.3	30.4	35.9
GRACE [6] (Structured ID)	COCO	16.7	39.2	50.3
IRGen [14]	COCO	29.6	50.7	56.3
<b>GENIUS</b>	M-BEIR	40.0	65.5	76.8
<b>GENIUS<sup>R</sup></b>	M-BEIR	42.6	67.3	78.9
<b>GENIUS</b>	COCO	41.2	67.8	77.8
<b>GENIUS<sup>R</sup></b>	COCO	<b>46.1</b>	<b>74.0</b>	<b>82.7</b>

Table 7. Comparison of generative retrieval methods on text-to-image retrieval benchmarks. Results are reported as Recall@k (%). <sup>†</sup> indicates zero-shot performance, highlighting the ability of the model to generalize without task-specific fine-tuning.

does not necessarily improve performance. For instance, on COCO text-to-image retrieval, Recall@5 decreases from 65.3% to 59.5%. However, on CIRR, this configuration leads to a significant performance improvement, highlighting the varying impact of codebook size depending on task complexity and modality. Overly large and fine-grained codebook configurations, while occasionally beneficial, increase the complexity of training the decoder model.

When using a shared codebook, Recall@5 on COCO drops drastically to 18.6%. Similar declines are observed across other tasks, indicating that level-specific codebooks are crucial for capturing the unique characteristics of different semantics. These findings highlight the importance of carefully configuring the codebook to ensure effective quantization and retrieval performance.

## D. Additional Quantitative Results

We present performance evaluations for additional settings not covered in the main paper, including variations in beam size and comparisons with a broader range of baselines.

### D.1. Standard Generative Retrieval Benchmark

We evaluate GENIUS against prior generative retrieval methods, including GRACE and IRGen, on standard text-to-image benchmarks such as Flickr30K and COCO, as summarized in Table 7. Unlike GRACE and IRGen, which are specifically designed for text-to-image tasks, GENIUS



is originally trained on the M-BEIR benchmark in a multi-task setting, supporting diverse retrieval scenarios while also being capable of task-specific training. Note that Flickr30K is not included in the M-BEIR dataset.

On Flickr30K, GENIUS trained with M-BEIR achieves an impressive zero-shot Recall@5 of 74.1%, surpassing GRACE by over 15 percentage points, despite having never seen the dataset during training. When fine-tuned exclusively on Flickr30K and combined with re-ranking, GENIUS further improves its performance to a Recall@5 of 92.0%, setting a new state-of-the-art for generative retrieval on this benchmark. On COCO, GENIUS trained with M-BEIR achieves a Recall@5 of 65.5%, significantly outperforming GRACE (39.2%) and IRGen (50.7%). When trained solely on COCO, GENIUS improves further to a Recall@5 of 74.0%. These results highlight the generalization ability of GENIUS to unseen datasets within a multi-task learning framework. Although M-BEIR includes domains similar to Flickr30K (e.g., COCO), GENIUS achieves zero-shot performance that surpasses models specifically trained on the same domain. Furthermore, GENIUS excels in task-specific scenarios, achieving superior performance when trained on individual datasets and achieving state-of-the-art results.

## D.2. Dataset-Specific Retrieval

Table 8 summarizes the performance of GENIUS across various retrieval tasks, demonstrating its ability to outperform prior generative methods and achieve results close to state-of-the-art embedding-based baselines in specific tasks. For text-to-image retrieval on COCO, GENIUS achieves a Recall@5 of 65.5% with a beam size of 30, significantly surpassing IRGen at 50.7%. With embedding-based re-ranking, performance improves to 78.0%, narrowing the gap with CLIP-SF, which achieves 81.7%. In image-to-text retrieval on COCO, GENIUS achieves a Recall@5 of 91.1% with re-ranking and a beam size of 50, nearly matching the 92.3% of CLIP-SF.

For relational reasoning tasks in CIRR, GENIUS achieves a Recall@5 of 35.5% with a beam size of 30. Increasing the beam size to 50 and incorporating re-ranking raises performance to 39.5%, demonstrating its strength in addressing relational queries. On WebQA, which features knowledge-intensive and long-form queries, embedding-based re-ranking boosts Recall@5 for text-to-text retrieval from 36.3% to 44.6%, effectively leveraging additional search space to handle semantically complex data. GENIUS already shows superior performance compared to prior generative methods with beam search alone. Moreover, by combining larger beam sizes with embedding-based re-ranking, GENIUS often achieves performance levels that are competitive with embedding-based state-of-the-art methods.

## D.3. Universal Retrieval

The universal retrieval performance of GENIUS demonstrates its ability to handle diverse tasks effectively, as shown in Table 9. Increasing the beam size alone does not always result in significant performance improvements. However, embedding-based re-ranking plays a crucial role in refining candidate sets and enhancing retrieval performance, often enabling GENIUS to approach state-of-the-art performance.

For image-to-text retrieval on MSCOCO, Recall@5 improves from 82.7% with beam search alone to 90.6% with re-ranking at a beam size of 50, narrowing the gap with CLIP-SF (92.3%). This highlights the strength of re-ranking in prioritizing relevant candidates that may not rank highly within the initial beam output. Similarly, on the OVEN dataset for image and text pair-to-text retrieval, Recall@5 increases from 34.4% to 38.0% with re-ranking at a larger beam size, effectively closing the gap with CLIP-SF (39.2%). On NIGHTS, which involves image-to-image retrieval, re-ranking produces a substantial improvement, with Recall@5 jumping from 8.4% to 30.2% at the largest beam size. These results indicate that while GENIUS generates strong candidates through beam search, embedding-based re-ranking is essential to achieve competitive performance, especially at larger beam sizes where the expanded search space requires further refinement to prioritize relevance.

## E. More Visualizations of Quantization

To illustrate how our modality-decoupled semantic quantization operates, we provide further visualizations demonstrating its dual properties of modality separation and coarse-to-fine semantic refinement across subsequent levels. These examples highlight the ability of GENIUS to handle multimodal data through structured code, capturing progressively distinct semantic details.

At the **first level**, codes represent modality distinctions: 0 for images, 1 for text, and 2 for image-text pairs. This clear separation ensures that the retrieval system processes each modality appropriately, which forms the foundation for multimodal data handling.

The **second level encodes** broad semantic concepts, capturing *primary objects* or *key scenes* shared across multimodal data. As shown in Fig. 2, examples include 1782 (i.e., a cat), grouping examples featuring cats in various contexts, such as lying on tables, eating bananas, or curling on skateboards. Other examples include 1534 (i.e., teddy bears), highlighting scenes like picnics or playful activities, and 3260 (i.e., flying a kite), which captures shared actions across different settings. Similarly, 1640 (i.e., hotel room) clusters scenes with shared elements like beds and lamps. These groupings extend naturally to other domains, cate-

Fine-tuning		COCO		VisualNews		Fashion200K		Nights	EDIS
		T to I	I to T	T to I	I to T	T to I	I to T	I to I	T to (I,T)
Embedding-based Retrieval									
CLIP-SF [12]	Single Task	81.7	89.8	43.5	42.7	10.7	12.0	33.5	58.8
BLIP-FF [12]		77.3	86.0	20.0	22.4	17.1	15.6	30.4	38.2
CLIP-SF [12]	Unified Instruction	81.1	92.3	42.6	43.1	18.0	18.3	32.0	59.4
BLIP-FF [12]		67.5	89.9	23.4	22.8	26.1	28.9	33.0	50.9
Generative Retrieval									
GRACE [6]	Single Task	39.5	–	–	–	–	–	–	–
IRGen [14]		50.7	–	–	–	–	–	–	–
GENIUS ( $\mathcal{B} = 30$ )	Unified Instruction	65.5	83.4	17.5	17.5	13.6	17.0	8.4	35.6
GENIUS <sup><math>\mathcal{R}</math></sup> ( $\mathcal{B} = 30$ )		67.3	89.7	23.3	24.0	15.2	18.9	29.0	41.4
GENIUS ( $\mathcal{B} = 50$ )		68.1	83.2	18.5	18.7	13.7	12.8	8.2	37.0
GENIUS <sup><math>\mathcal{R}</math></sup> ( $\mathcal{B} = 50$ )		78.0	91.1	27.4	28.4	16.2	16.3	30.2	44.3
Fine-tuning		WebQA		OVEN		InfoSeek		FashionIQ	CIRR
		T to T	T to (I,T)	(I,T) to T	(I,T) to (I,T)	(I,T) to T	(I,T) to (I,T)	(I,T) to I	(I,T) to I
Embedding-based Retrieval									
CLIP-SF [12]	Single Task	81.7	76.3	45.4	66.2	23.5	47.4	25.9	52.0
BLIP-FF [12]		67.5	67.8	33.8	49.9	18.5	32.3	3.0	13.9
CLIP-SF [12]	Unified Instruction	84.7	78.7	45.5	67.6	23.9	48.9	24.4	44.6
BLIP-FF [12]		80.0	79.8	41.0	55.8	22.4	33.0	29.2	52.2
Generative Retrieval									
GENIUS ( $\mathcal{B} = 30$ )	Unified Instruction	28.8	47.4	34.9	34.6	12.4	15.1	12.8	21.0
GENIUS <sup><math>\mathcal{R}</math></sup> ( $\mathcal{B} = 30$ )		36.3	54.9	36.6	35.0	18.0	26.7	17.5	35.5
GENIUS ( $\mathcal{B} = 50$ )		32.5	49.7	36.6	36.4	11.2	14.6	13.2	20.7
GENIUS <sup><math>\mathcal{R}</math></sup> ( $\mathcal{B} = 50$ )		44.6	60.6	41.9	52.5	20.7	30.1	19.3	39.5

Table 8. **Task-specific Information Retrieval.** Recall@5 results of single-task and unified instruction fine-tuning methods on the M-BEIR dataset, except Fashion200K and FashionIQ, where Recall@10 is reported.  $\mathcal{B}$  represents the beam size, and  $\mathcal{R}$  indicates re-ranking based on embedding vectors within the predicted candidate set. I and T denote image and text modalities, respectively, and (I,T) indicates the retrieval direction for image-to-text or text-to-image tasks.

Task	Dataset	Embedding-based Retrieval				Generative Retrieval			
		CLIP <sub>SF</sub>	CLIP <sub>FF</sub>	BLIP <sub>SF</sub>	BLIP <sub>FF</sub>	<b>GENIUS</b> ( $\mathcal{B} = 30$ )	<b>GENIUS</b> <sup><math>\mathcal{R}</math></sup> ( $\mathcal{B} = 30$ )	<b>GENIUS</b> ( $\mathcal{B} = 50$ )	<b>GENIUS</b> <sup><math>\mathcal{R}</math></sup> ( $\mathcal{B} = 50$ )
1. $q_t \rightarrow c_i$	VisualNews	<b>42.6</b>	28.8	20.9	23.0	18.5	23.9	18.5	<b>27.3</b>
	MSCOCO	<b>77.9</b>	74.7	71.6	75.6	55.4	64.8	55.1	<b>68.0</b>
	Fashion200K	17.8	15.5	24.3	<b>25.4</b>	13.6	14.7	13.7	<b>16.2</b>
2. $q_t \rightarrow c_t$	WebQA	<b>84.7</b>	78.4	78.9	79.5	28.3	36.5	31.1	<b>42.9</b>
3. $q_t \rightarrow (c_i, c_t)$	EDIS	<b>59.4</b>	50.0	47.2	50.3	35.4	41.4	36.6	<b>44.1</b>
	WebQA	78.8	75.3	76.8	<b>79.7</b>	47.1	55.8	49.0	<b>59.7</b>
4. $q_i \rightarrow c_t$	VisualNews	<b>42.8</b>	28.6	19.4	21.1	17.3	23.2	18.4	<b>26.8</b>
	MSCOCO	<b>92.3</b>	89.0	88.2	88.8	82.7	89.4	82.7	<b>90.6</b>
	Fashion200K	17.9	13.7	24.3	<b>27.6</b>	12.2	14.8	12.8	<b>16.2</b>
5. $q_i \rightarrow c_i$	NIGHTS	32.0	31.9	<b>33.4</b>	33.0	8.4	28.8	8.1	<b>30.2</b>
6. $(q_i, q_t) \rightarrow c_t$	OVEN	<b>39.2</b>	34.7	35.2	38.7	34.4	37.1	34.6	<b>38.0</b>
	InfoSeek	<b>24.0</b>	17.5	16.7	19.7	11.1	16.6	10.4	<b>18.0</b>
7. $(q_i, q_t) \rightarrow c_i$	FashionIQ	24.3	20.5	26.2	<b>28.5</b>	12.8	17.4	18.9	<b>19.2</b>
	CIRR	43.9	40.9	43.0	<b>51.4</b>	20.5	34.9	20.1	<b>38.3</b>
8. $(q_i, q_t) \rightarrow (c_i, c_t)$	OVEN	<b>60.2</b>	55.8	51.8	57.8	36.9	40.9	36.5	<b>48.6</b>
	InfoSeek	<b>44.6</b>	36.8	25.4	27.7	14.3	25.7	14.2	<b>28.6</b>
Average		<b>48.9</b>	43.3	42.7	45.5	28.1	35.4	28.8	<b>38.3</b>

Table 9. **Universal Information Retrieval.** Recall@5 for various tasks on the M-BEIR dataset, retrieved from a global pool across diverse modalities.  $\mathcal{B}$  represents the beam size, and  $\mathcal{R}$  indicates re-ranking based on embedding vectors within the predicted candidate set.

gorizing items like dresses, trousers, and jackets based on shared object types.

The **third-level codes** refine semantics by focusing on *attributes* such as material, color, and patterns. Fig. 3 illustrates these details. In COCO, 3771 (*i.e.*, a bunch of) groups collections of items like stacked oranges, vegetables, or bananas, emphasizing grouping semantics. Similarly, 1443 (*i.e.*, green) identifies objects prominently featuring green, such as train, fire hydrants, and bananas. In Fashion200K, 1443 (*i.e.*, green) also highlights garments sharing the color green, while 1275 (*i.e.*, striped clothing) focuses on items with striped patterns, such as blazers and trousers. Lastly, 3559 (*i.e.*, velvet) captures items made of velvet material, regardless of the type of clothing, showcasing material-specific details.

The **fourth-level codes** capture *highly fine-grained semantics*, such as specific actions, positions, and intricate design features. Fig. 4 provides examples from COCO, including 675 (*i.e.*, leaning down), which groups scenes featuring subjects leaning, such as giraffes eating grass or people bending over. Similarly, 1412 (*i.e.*, in-bedroom) emphasizes indoor bedroom settings, capturing nuanced elements beyond generic room scenes. Furthermore, 643 (*i.e.*, carrying) captures actions involving carrying objects, such as individuals carrying suitcases or animals transporting items. In Fashion200K, codes like 190 (*i.e.*, sleeveless style), 817 (*i.e.*, biker style), and 826 (*i.e.*, bomber style) reflect fine-grained characteristics of garments, such as sleeveless cuts, biker styles, or specific jacket designs.

While the examples showcase the first four levels, the quantization process extends further to encode increasingly fine details, enriching semantic representation. Although these examples primarily showcase COCO and Fashion200K data, the quantization framework is designed to generalize across datasets. Shared semantics, such as 1443 (*i.e.*, green) in second-level remain consistent across different domains, highlighting the universal nature of the code structure. This capability ensures consistent capturing and alignment of similar semantics, irrespective of the dataset. These properties enable the decoder in our GENIUS framework to seamlessly map multimodal data to their corresponding codes. As a result, by leveraging this structured and interpretable quantization, GENIUS achieves not only high retrieval performance but also ensures generalization across a wide range of tasks, spanning various modalities and domains.

*, 1782 , *				A <b>cat</b> is on a table with a cloth on it.	A <b>house cat</b> is taking a bite from a banana.	A <b>cat</b> curled up on a skateboard in a living room.
*, 1534 , *				Several <b>teddy bears</b> appear to have a picnic on the grass.	<b>Teddy bears</b> do want to be ice hockey players.	A wicker picnic hamper with three <b>teddy bears</b> .
*, 3260 , *				A woman standing in a field <b>flying a kite</b> .	Two people are <b>flying a large character kite</b> on the grass.	A person sitting on the green grass <b>flying a kite</b> .
*, 1640 , *				The <b>hotel room</b> headboard is also a desk.	A <b>tan and white bed</b> some chairs, pillows and a lamp.	A house where you can see the <b>bedroom</b> and relaxing room.
*, 3748 , *				Multicolor gracie pintuck jersey <b>trouser</b> .	Blue cropped pleated <b>pants</b> .	Black matilda pocket side wide leg <b>trousers</b> .
*, 2703 , *				Purple shelburne slim fit genuine coyote fur <b>trim down parka</b>	Black fur trim bi-stretch <b>down jacket</b> .	Brown quilted <b>puffer jacket</b> .
*, 1283 , *				White crepe mini wrap <b>dress</b> .	Pink long sleeve shift <b>dress</b> ruffle front.	Multicolor one shoulder <b>dress</b> .

Figure 2. Examples of second-level codes in the modality-decoupled semantic quantization. This level captures coarse semantics, such as primary objects or key scenes, with rows representing scenes from COCO and Fashion200K datasets.















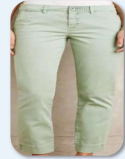








* , * , 3771 , *				A person standing next to <b>many</b> luggage bags	<b>A bunch of</b> ripe oranges are stacked neatly on top of each other	Vegetables <b>are stacked up</b> high on a market stand.
* , * , 2658 , *				<b>A fire burns</b> while a person rides a green motorcycle.	There is a stone oven pizza <b>near the</b> fire place.	Hot dogs on a skewer <b>roasting over</b> a fire.
* , * , 1909 , *				A horse eating <b>grass in a green</b> field.	Several zebras <b>eat the green</b> grass in the pasture.	A baby giraffe bending over to <b>graze on the</b> grass.
* , * , 1443 , *				<b>A green</b> train is going down the tracks in a rural setting.	A rusty <b>green</b> fire hydrant that is next to the curb.	A pile of <b>green</b> bananas sitting on top of a table.
* , * , 1443 , *				<b>Green</b> habitat crop pants.	<b>Green</b> dani water-resistant topper jacket.	<b>Green</b> annabelle convertible tulle column dress.
* , * , 1275 , *				Beige <b>stripe</b> cotton trousers.	Silver open back sleeveless shift dress <b>stripe</b> .	Yellow double-breasted <b>striped</b> blazer.
* , * , 3559 , *				Black arianna zip front tailored <b>velvet</b> trousers.	Multicolor <b>velvet</b> cross halter gown.	Blue <b>velvet</b> maxi dress.

Figure 3. Examples of third-level codes in the modality-decoupled semantic quantization. This level captures finer semantic attributes, such as object properties, material characteristics, or detailed patterns, across COCO and Fashion200K datasets.











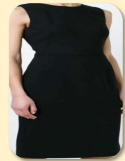







*,*,*, 675, *				A giraffe is <b>leaning down</b> to eat the grass.	A woman sitting at a desk next to a woman <b>bending over</b> .	A girl holding a cellphone <b>squatting down</b> on the corner of a street.
*,*,*, 1412, *				A very messy <b>bedroom</b> with many items laying all over it.	A black cat is laying <b>on a bed</b> .	A young child sits on <b>top of a bunk bed</b> .
*,*,*, 2837, *				Two giraffes look out from a zoo <b>towards the city skyline</b> .	<b>A city</b> full of buildings under a smoggy sky.	A clock below tall buildings in a <b>large city</b> .
*,*,*, 643, *				A person using cross country skis <b>pulling</b> a bundle behind them.	A man driving a motorcycle down the road <b>with</b> a box tied to the <b>back</b> of it.	A woman <b>carrying</b> a surfboard over her head on the beach.
*,*,*, 190, *				Silver open back <b>sleeveless</b> shift dress stripe.	Purple overlay chiffon <b>tank</b> dress.	Black double breasted <b>sleeveless</b> blazer.
*,*,*, 817, *				Green slim-fit <b>biker</b> trousers.	Beige shearling <b>biker</b> vest.	Multicolor boxy leather <b>moto</b> jacket.
*,*,*, 826, *				Blue amber tartan lined harrington <b>bomber</b> .	Black le velvet <b>bomber</b> jacket.	Black originals 3 stripe zip front supergirl <b>bomber</b> track top.

Figure 4. Examples of fourth-level codes in the modality-decoupled semantic quantization. This level captures highly fine-grained semantics, such as specific actions, positions, nuanced object details, or intricate clothing features.

## References

- [1] Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. Webqa: Multihop and multimodal qa. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [1](#), [2](#)
- [2] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. [1](#), [2](#)
- [3] Stephanie Fu, Netanel Yakir Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [1](#), [2](#)
- [4] Xintong Han, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis. Automatic spatially-aware fashion concept discovery. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017. [1](#), [2](#)
- [5] Hexiang Hu, Yi Luan, Yang Chen, Urvashi Khandelwal, Mandar Joshi, Kenton Lee, Kristina Toutanova, and Ming-Wei Chang. Open-domain visual entity recognition: Towards recognizing millions of wikipedia entities. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [1](#), [2](#)
- [6] Yongqi Li, Wenjie Wang, Leigang Qu, Liqiang Nie, Wenjie Li, and Tat-Seng Chua. Generative cross-modal retrieval: Memorizing images in multimodal language models for retrieval and beyond. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. [4](#), [6](#)
- [7] 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 *Proc. European Conference on Computer Vision (ECCV)*, 2014. [1](#), [2](#)
- [8] Fuwen Liu, Xiaojie Wang, Jianping Shi, Alan L Huang, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Visual news: Benchmark and challenges in news image captioning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. [1](#), [2](#)
- [9] Siqi Liu, Weixi Feng, Tsu-jui Fu, Wenhua Chen, and William Yang Wang. Edis: Entity-driven image search over multimodal web content. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4877–4894, 2023. [1](#), [2](#)
- [10] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. [1](#), [2](#)
- [11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proc. International Conference on Machine Learning (ICML)*, 2021. [3](#)
- [12] Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhua Chen. Uniir: Training and benchmarking universal multimodal information retrievers. In *Proc. European Conference on Computer Vision (ECCV)*, 2024. [1](#), [2](#), [6](#)
- [13] Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. Fashion iq: A new dataset towards retrieving images by natural language feedback. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [1](#), [2](#)
- [14] Yidan Zhang, Ting Zhang, Dong Chen, Yujing Wang, Qi Chen, Xing Xie, Hao Sun, Weiwei Deng, Qi Zhang, Fan Yang, et al. Irgen: Generative modeling for image retrieval. In *Proc. European Conference on Computer Vision (ECCV)*, 2024. [4](#), [6](#)