Comparison of multimodal LLM embedding models in the task of cross-modal recipe retrieval

Keisuke Gomi
The University of Electro-Communications
Chofu-shi, Tokyo, Japan
gomi-k@mm.inf.uec.ac.jp

Keiji Yanai
The University of Electro-Communications
Chofu-shi, Tokyo, Japan
yanai@cs.uec.ac.jp

***Abstract*—Cross-modal retrieval between food images and recipe texts is an important task that consists of image-to-recipe retrieval, which searches for the corresponding recipe from a food image, and recipe-to-image retrieval, which searches for the corresponding food image from a recipe text. Recent multimodal LLM embedding models have achieved strong performance in general image–text retrieval, but their effectiveness for food image–recipe retrieval on Recipe1M has not been sufficiently investigated. In this paper, we apply SIMMER, which had previously been used only with the VLM2Vec family, to GME and Qwen3-VL-Embedding, and compare four multimodal LLM embedding models on Recipe1M. In addition to directly applying SIMMER, we examine performance changes caused by modifying its training configuration and incorporating SMART, an inference-time method that requires no additional training. Experimental results show that both GME and Qwen3-VL-Embedding achieve meaningful zero-shot retrieval performance, while fine-tuning them with SIMMER on Recipe1M leads to substantial performance improvements. Qwen3-VL-Embedding-8B with batch size 512 gives the strongest image-to-recipe R@1, and its SMART variant gives the best recipe-to-image results. At 10k, the SMART variant improves R@1 over SIMMER (V1-7B) from 65.5% to 69.3% for image-to-recipe retrieval and from 61.5% to 68.6% for recipe-to-image retrieval.**

***Index Terms*—multimodal large language model, recipe retrieval, cross-modal retrieval, contrastive learning**

I. Introduction

Food is central to everyday life, and recipe-sharing services and social media now provide large collections of food images and recipe texts. This growth makes cross-modal recipe retrieval important: systems should retrieve a matching recipe from a food image and a matching image from a recipe text. Such bidirectional retrieval can support practical applications including dietary logging, nutritional management, and cooking assistance.

In parallel, Multimodal Large Language Models (MLLMs) have made rapid progress in computer vision and natural language processing. Representative models, including LLaVA [1] and Qwen2-VL [2], demonstrate strong capabilities for jointly understanding visual and textual inputs. Recent studies have further investigated the use of MLLMs as embedding models. Approaches such as VLM2Vec [3] and MM-Embed [4] suggest that the multimodal knowledge learned through large-scale pre-training can be transferred to the construction of effective multimodal embeddings.

Following this trend of using MLLMs as embedding models, SIMMER [5] was proposed for cross-modal food image–recipe retrieval. Unlike conventional dual-encoder architectures, SIMMER encodes both images and recipe texts using a single unified MLLM, thereby leveraging the shared multimodal representation space of the underlying model. However, SIMMER has been applied only to models from the VLM2Vec family [3], [6], and its generalizability to other MLLM-based embedding models remains unexplored.

In this study, we extend SIMMER to MLLM-based embedding models outside the VLM2Vec family. Specifically, we adopt two model families, GME [7] and Qwen3-VL-Embedding [8], each of which provides two model sizes, resulting in four models in total. Beyond directly applying SIMMER, we further investigate whether retrieval performance can be improved by modifying part of the SIMMER training configuration and by incorporating SMART [9], an inference-time method that does not require additional training. We validate the effectiveness of this study through experiments on Recipe1M [10], a large-scale benchmark dataset for food image–recipe retrieval.

The main contributions of this paper are summarized as follows:

- We show that SIMMER, which had previously been applied only to the VLM2Vec family, is also effective for other MLLM-based embedding models.
- We show that larger-batch training and SMART yield competitive results among the evaluated configurations.
- We apply SMART to cross-modal recipe retrieval and confirm its effectiveness, particularly for recipe-to-image retrieval.

II. Related work

Cross-modal retrieval between food images and recipe texts has become an actively studied task since the release of the Recipe1M dataset [10]. Compared to generic image–text retrieval, a key characteristic of this problem is that recipe texts are inherently structured, typically consisting of a title, a list of ingredients, and a sequence of cooking instructions. Accordingly, progress in this area has been closely tied to (i) how each modality is encoded and (ii) how cross-modal alignment objectives exploit this structured text.

Early approaches learned a shared embedding space for images and recipes using separate visual and textual encoders. Salvador et al. [10] introduced a joint embedding model with a ranking objective and semantic regularization, establishing the standard Recipe1M retrieval protocol. Subsequent representative methods improved cross-modal alignment through adversarial learning [11], stochastic latent-variable modeling [12], and disentangled recipe representations [13]. These methods established the standard food image–recipe retrieval setting, but their recipe encoders generally relied on limited language pre-training and had difficulty representing long, compositional recipe semantics.

Later studies emphasized richer recipe modeling and stronger cross-modal alignment. The Hierarchical Transformer (H-T) [14] encoded recipe components and introduced a self-supervised objective over titles, ingredients, and instructions. Other representative methods used Transformer decoders with cross-modal regularization [15] or improved Transformer-based training regimes [16], [17].

Recent work has increasingly adapted large-scale vision–language pre-training and foundation models to structured recipes. Representative examples include adaptation of vision–language pre-training to recipe text [18], foundation-model-driven data augmentation [19], fine-grained component alignment [20], and bias mitigation for food images [21].

Most recently, SIMMER [5] introduced an MLLM-based embedding approach to food image–recipe retrieval by using a unified MLLM encoder for both food images and recipe texts. However, its experiments were limited to the VLM2Vec family [3], [6], leaving open how effective other multimodal LLM embedding models are for food image–recipe retrieval on Recipe1M. This study addresses this gap by comparing additional multimodal LLM embedding models under the Recipe1M retrieval setting.

III. Method

A. Overview

In this study, we apply SIMMER [5] to MLLM-based embedding models beyond the VLM2Vec family [3], [6] used in the original framework. Specifically, we use GME [7] and Qwen3-VL-Embedding [8] as additional base models. GME is an embedding model built on Qwen2-VL [2], the same backbone family as VLM2Vec, whereas Qwen3-VL-Embedding is a recent MLLM-based embedding model based on Qwen3-VL [22]. In addition to applying SIMMER to these alternative base models, we modify the training configuration and incorporate SMART [9] into the retrieval pipeline. The modification to the training configuration is described in Section IV-C2.

B. Problem Formulation

Let $\{(v_i, r_i)\}_{i=1}^{N}$ denote a training set of paired food images and recipes, where v_i represents the image and r_i represents its associated recipe text. The recipe text is treated as a structured input composed of a title, an ingredient list, and ordered cooking instructions. The objective is to obtain a shared encoder $\Phi(\cdot)$ that projects images and recipes into the same embedding space, so that the embeddings of a matched pair are assigned high similarity while those of mismatched pairs are separated. In this study, this encoder is implemented with a Multimodal Large Language Model (MLLM), which can process both visual and textual inputs within a single model interface.

C. Prompt Design

In VLM2Vec, which was used as the base model in SIMMER, there is no explicit separation between the task instruction and the data to be encoded. For example, when an instruction and a recipe text are provided together, they are simply concatenated into a single input string. In contrast, GME and Qwen3-VL-Embedding, which we use as base models in this study, follow a different input format that explicitly separates the task instruction from the target content. Specifically, the task instruction is provided as the system message, while the text or image to be encoded is provided as the user message.

To enable a fair comparison with SIMMER, we preserve the original SIMMER prompt templates as much as possible while adapting them to the input format required by the newly applied models. Because SIMMER formulates contrastive learning as a unidirectional retrieval task, the same data point is given different prompt templates depending on whether it is used as a query or as a candidate. The resulting prompt templates for image-to-recipe retrieval and recipe-to-image retrieval are shown in Table I and Table II, respectively.

Table I shows the prompt template for image-to-recipe retrieval. When an image is used as the query, the instruction for retrieving a recipe is placed in the system message, and the image to be encoded is placed in the user message. When a recipe is used as a candidate in SIMMER, no explicit instruction is provided, and only the recipe text is given to the model. Therefore, we do not provide a specific instruction in the system message for the candidate side. As a result, the model-specific default instruction, denoted by `{default_instruction}`, is used when the instruction field is left unspecified. The default instruction is "You are a helpful assistant." for GME and "Represent the user's input." for Qwen3-VL-Embedding. In the recipe text, `{title}` denotes the recipe title, `{ingredients}` denotes the comma-separated list of ingredients, and `{instructions}` denotes the sequence of cooking steps joined by spaces.

Table II shows the prompt template for recipe-to-image retrieval. When a recipe is used as the query, the retrieval instruction is placed in the system message, and the recipe text to be encoded is placed in the user message, as in the image-to-recipe setting. Unlike the candidate side in image-to-recipe retrieval, however, the image candidate is encoded with an explicit system instruction: "Represent the given food image for recipe prediction." This difference between image-to-recipe and recipe-to-image retrieval arises from the original SIMMER prompt design, where text inputs are encoded

TABLE I: Prompt for image-to-recipe retrieval.

Target	Role	Message content
Query	system	Find a cooking recipe describing the given food image.
	user	`{image}`
Candidate	system	`{default_instruction}`
	user	A cooking recipe: Title: `{title}`, Ingredients: `{ingredients}`, Instructions: `{instructions}`

TABLE II: Prompt for recipe-to-image retrieval.

Target	Role	Message content
Query	system	Find me a food image that matches the given cooking recipe.
	user	Title: `{title}`, Ingredients: `{ingredients}`, Instructions: `{instructions}`
Candidate	system	Represent the given food image for recipe prediction.
	user	`{image}`

without an additional instruction, whereas image inputs are encoded together with an instruction text.

D. Training Objectives

To achieve the objective described in Section III-B, we fine-tune the MLLM as illustrated in Figure 1. We use the hidden-state vector of the last token from the final layer of the MLLM as the embedding vector and train the model with the information noise-contrastive estimation (InfoNCE) loss [23]. Unlike CLIP [24], which uses a bidirectional contrastive loss, our training objective uses a unidirectional InfoNCE loss. Bidirectional retrieval is achieved by training with both the image-to-recipe dataset and the recipe-to-image dataset.

Concretely, given a mini-batch of B query–candidate pairs $\{(q_i, c_i)\}_{i=1}^{B}$, the matched pair is treated as the positive and all other in-batch candidates serve as negatives. The InfoNCE loss is defined as:

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\mathrm{sim}(\mathbf{e}_{q_i}, \mathbf{e}_{c_i})/\tau)}{\sum_{j=1}^{B} \exp(\mathrm{sim}(\mathbf{e}_{q_i}, \mathbf{e}_{c_j})/\tau)} \tag{1}$$

where $\mathbf{e}_{q_i}$ and $\mathbf{e}_{c_i}$ are the embeddings of the query and the candidate, respectively, $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity, and τ is a temperature hyperparameter.

E. SMART

When a single-vector embedding model is used for cross-modal food image–recipe retrieval, a food image containing multiple dishes must still be represented by a single embedding vector. This can make retrieval difficult because the visual characteristics of multiple dishes are compressed into one global representation. To address this issue, we use

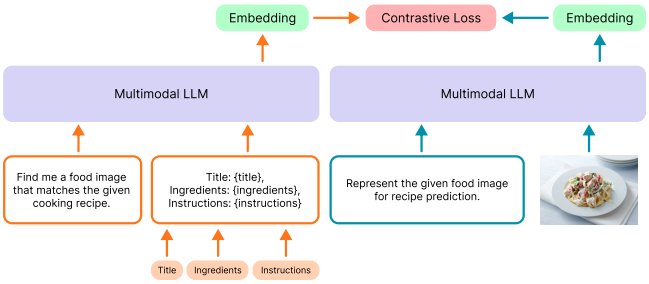


Fig. 1: Overview of the contrastive learning procedure used for fine-tuning.

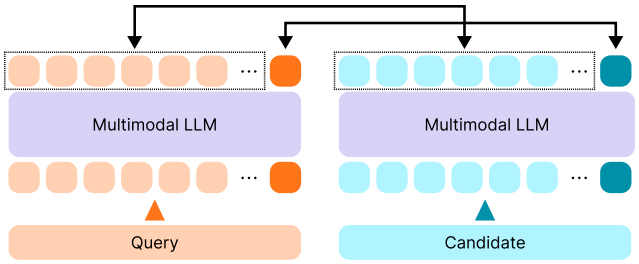


Fig. 2: SMART scoring with late interaction between hidden-state vectors.

SMART [9], a method that converts a single-vector embedding model into a multi-vector embedding model. Although SMART has several variants, we adopt its training-free variant, which only changes the similarity scoring procedure at inference time.

As in many MLLM-based embedding models, including the models considered in this study, the hidden-state vector of the last token from the final layer is used as the embedding vector. As shown in Figure 2, SMART additionally uses hidden-state vectors from tokens other than the last token, which are typically discarded in single-vector embedding models, to compute the similarity score. Specifically, the similarity score is computed using the MAXSIM late-interaction score [25], defined as follows:

$$\mathrm{sim}_{\mathrm{late}}(q, c) = \frac{1}{|M_q|}\sum_{i \in M_q} \max_{j \in M_c} \mathrm{sim}(h_{q,i}, h_{c,j}) \tag{2}$$

where $h_{q,i}$ and $h_{c,j}$ denote the hidden-state vectors of the i-th query token and the j-th candidate token, respectively. M_q and M_c denote the sets of token indices excluding the token used as the single-vector embedding.

We then combine this late-interaction score with the conventional single-vector similarity score, defined as $\mathrm{sim}_{\mathrm{single}}(q, c) = \mathrm{sim}(\mathbf{e}_q, \mathbf{e}_c)$, to obtain the final score:

$$\mathrm{sim}_{\mathrm{hybrid}}(q, c) = \mathrm{sim}_{\mathrm{single}}(q, c) + \mathrm{sim}_{\mathrm{late}}(q, c) \tag{3}$$

IV. Experiments

A. Experimental Setup

1) Dataset: We conduct the evaluation on Recipe1M [10], a large-scale food dataset containing more than one million recipes and their associated images. Because image annotations are not available for every recipe, we restrict the experiments to recipes with paired food images, yielding 238,408 training pairs, 51,119 validation pairs, and 51,304 test pairs. Each recipe entry includes a title, a set of ingredient strings, and a sequence of cooking instruction strings.

2) Evaluation Metrics: We use the standard retrieval metrics for this task: median rank (medR) and Recall@k (R@k, $k = 1, 5, 10$). The medR score is computed as the median rank assigned to the correct item in the retrieval list, with smaller values indicating better retrieval quality. Recall@k is the fraction of queries whose ground-truth match is included among the top-k retrieved candidates.

Results are reported using the two commonly used evaluation protocols, namely the 1k and 10k settings. For the 1k setting, we randomly sample 1,000 image–recipe pairs from the test set, compute the metrics on the sampled subset, and average the results over 10 independent repetitions to reduce sampling variance. The 10k setting evaluates retrieval over a larger pool of 10,000 candidate pairs and therefore provides a more difficult test case.

3) Implementation Details: We employ four base models: GME-2B and GME-7B [7], and Qwen3-VL-Embedding-2B and Qwen3-VL-Embedding-8B [8]. Their embedding dimensions are 1536, 3584, 2048, and 4096, respectively.

Each model is fine-tuned for 2,000 steps with the Adam optimizer [26] and a learning rate of 1×10^{-4}. We optimize the InfoNCE loss using a temperature of $\tau = 0.02$. Unless otherwise stated, the global batch size is set to 128. To fine-tune the models in a parameter-efficient manner, we use Low-Rank Adaptation (LoRA) [27] with rank 16, scaling factor $\alpha = 64$, and dropout rate 0.05.

B. Comparison with Existing Methods

Table III compares existing methods with the models newly trained in this study. Qwen3-VL-Embedding-8B with batch size 512 and SMART achieves the strongest recipe-to-image performance and competitive image-to-recipe performance. Compared with SIMMER (V1-7B), this SMART variant improves 10k image-to-recipe R@1 from 65.5% to 69.3%, corresponding to a gain of 3.8 percentage points. On the 10k recipe-to-image retrieval setting, it improves R@1 from 61.5% to 68.6%, yielding a larger gain of 7.1 percentage points.

On the other hand, simply applying SIMMER to newer base models does not consistently surpass the original SIMMER (V1-7B). On general-purpose multimodal embedding benchmarks such as MMEB [3], [6], Qwen3-VL-Embedding-8B is reported to perform better than VLM2Vec-7B. Nevertheless, one possible reason for the present result is that our training configuration and prompt design remain close to those used by SIMMER for VLM2Vec. This suggests that further improvements may be obtained by designing model-specific training configurations and prompts.

C. Ablation Study

1) Effect of Fine-tuning: To examine the effect of fine-tuning on Recipe1M, we compare the four models trained in our experiments with their zero-shot counterparts. The results are shown in Table IV. The zero-shot setting uses the pre-trained embedding models directly, whereas the SIMMER setting fine-tunes them on Recipe1M. Fine-tuning consistently improves retrieval performance for all four models. In the 1k image-to-recipe retrieval setting, R@1 increases from 49.5% to 79.0% for GME-2B, from 53.0% to 82.6% for GME-7B, from 57.0% to 83.0% for Qwen3-VL-Embedding-2B, and from 60.2% to 86.0% for Qwen3-VL-Embedding-8B. A similar trend is observed for recipe-to-image retrieval, where the 1k R@1 scores improve from 43.7% to 76.6%, 45.8% to 80.4%, 47.2% to 81.2%, and 49.6% to 84.6%, respectively. Although Recall is lower in the 10k setting because the retrieval pool is larger, the gains from fine-tuning remain consistent across both retrieval directions. The median rank also improves substantially: in the 10k zero-shot setting, many results have a median rank of 4.0 or higher, while most SIMMER results reach 1.0. These results indicate that GME and Qwen3-VL-Embedding models already provide non-trivial zero-shot cross-modal retrieval ability, but task-specific fine-tuning is important for strong performance on Recipe1M food image–recipe retrieval.

2) Effect of Batch Size: SimCLR [28] reports that batch size is an important factor in contrastive learning with in-batch negatives, where larger batches provide a larger set of negative examples. TNLBT [16] also demonstrates the effectiveness of contrastive learning with large batch sizes for cross-modal food image–recipe retrieval.

Motivated by these observations, we evaluate retrieval performance while varying the training batch size. We use Qwen3-VL-Embedding-8B as the target model for this analysis. Although the default SIMMER training setting uses a batch size of 128, we additionally evaluate larger batch sizes of 256 and 512. The results are shown in Table V.

Increasing the batch size from 128 to 256 improves performance across all evaluation metrics. However, further increasing the batch size from 256 to 512 does not consistently improve all metrics. Among the 12 Recall values across the 1k and 10k settings, six improve, one remains unchanged, and five decrease when the batch size is increased from 256 to 512.

We attribute this mixed trend to the adverse effect of false negative and noisy negative samples in the dataset, as discussed by Wu et al. [29]. When the batch size is increased from 128 to 256, the benefit of using more negative samples appears to dominate. In contrast, when the batch size is increased from 256 to 512, the benefit of additional negatives may become saturated, making the influence of false or noisy negatives more visible during training.

TABLE III: Experimental results on image-to-recipe and recipe-to-image retrieval. The best results are presented in **bold**, while the next best results are presented in underlined.

Setting	Method	Venue	Image-to-Recipe				Recipe-to-Image			
			medR↓	R@1↑	R@5↑	R@10↑	medR↓	R@1↑	R@5↑	R@10↑
1k	Salvador et al. [10]	CVPR'17	5.2	24.0	51.0	65.0	5.1	25.0	52.0	65.0
	H-T [14]	CVPR'21	1.0	60.0	87.6	92.9	1.0	60.3	87.6	93.2
	T-Food [15]	CVPRW'22	1.0	72.3	90.7	93.4	1.0	72.6	90.6	93.4
	Yang et al. [17]	ICMRW'24	1.0	81.8	95.9	97.8	1.0	81.2	96.0	97.9
	FARM [20]	WACV'24	1.0	73.7	90.7	93.4	1.0	73.6	90.8	93.5
	DAR [19]	ECCV'24	1.0	77.3	95.3	97.7	1.0	77.1	95.4	97.9
	Wang et al. [21]	MM'25	1.0	79.1	94.6	97.0	1.0	78.3	95.0	97.2
	SIMMER (V1-2B) [5]	-	1.0	84.1	97.3	98.8	1.0	81.5	96.6	98.4
	SIMMER (V1-7B) [5]	-	1.0	87.5	**98.0**	**99.2**	1.0	85.1	97.6	**99.1**
	SIMMER (V2) [5]	-	1.0	83.8	96.9	98.7	1.0	81.7	96.5	98.3
	Results obtained in this study									
	SIMMER (GME-2B)	-	1.0	79.0	95.5	97.8	1.0	76.6	94.4	97.3
	SIMMER (GME-7B)	-	1.0	82.6	96.4	98.2	1.0	80.4	96.2	98.1
	SIMMER (Qwen3-VL-Embedding-2B)	-	1.0	83.0	96.7	98.3	1.0	81.2	96.0	97.9
	SIMMER (Qwen3-VL-Embedding-8B)	-	1.0	86.0	97.5	98.9	1.0	84.6	97.1	98.7
	SIMMER (Qwen3-VL-Embedding-8B) + Batch Size 512 + SMART	-	1.0	**88.2**	**98.0**	99.1	1.0	**88.7**	**98.2**	**99.1**
10k	Salvador et al. [10]	CVPR'17	41.9	-	-	-	39.2	-	-	-
	H-T [14]	CVPR'21	4.0	27.9	56.4	68.1	4.0	28.3	56.5	68.1
	T-Food [15]	CVPRW'22	2.0	43.4	70.7	79.7	2.0	44.6	71.2	79.7
	Yang et al. [17]	ICMRW'24	1.0	56.5	81.0	87.6	1.0	55.7	80.2	87.1
	FARM [20]	WACV'24	2.0	44.9	71.8	80.0	2.0	44.3	71.5	80.0
	DAR [19]	ECCV'24	2.0	47.8	75.9	84.3	2.0	47.4	75.5	84.1
	Wang et al. [21]	MM'25	1.0	51.7	78.2	85.9	1.0	52.2	78.4	86.0
	SIMMER (V1-2B) [5]	-	1.0	59.7	83.5	89.9	1.0	55.8	81.1	88.0
	SIMMER (V1-7B) [5]	-	1.0	65.5	87.4	92.5	1.0	61.5	85.0	91.0
	SIMMER (V2) [5]	-	1.0	59.1	83.3	89.8	1.0	55.7	81.0	88.0
	Results obtained in this study									
	SIMMER (GME-2B)	-	1.0	53.3	78.1	85.7	1.6	49.7	75.1	83.4
	SIMMER (GME-7B)	-	1.0	58.8	81.7	88.1	1.0	52.8	78.5	86.2
	SIMMER (Qwen3-VL-Embedding-2B)	-	1.0	58.7	82.1	88.8	1.0	55.6	79.6	86.6
	SIMMER (Qwen3-VL-Embedding-8B)	-	1.0	63.9	85.3	90.8	1.0	61.9	84.2	90.1
	SIMMER (Qwen3-VL-Embedding-8B) + Batch Size 512 + SMART	-	1.0	**69.3**	**88.7**	**93.2**	1.0	**68.6**	**88.9**	**93.5**

In SIMMER, component-aware data augmentation is introduced to improve robustness to incomplete recipes by generating three variants from each image–recipe pair and adding them to the training data before training. This offline augmentation strategy can effectively introduce additional samples that may act as false negatives. Therefore, applying component-aware data augmentation online during training may reduce the number of false negatives in each batch and further improve retrieval performance.

3) Effect of SMART: We apply SMART [9] at inference time to Qwen3-VL-Embedding-8B trained with a batch size of 512. The results are shown in Table V. Comparing Recall before and after applying SMART, we observe slight decreases for image-to-recipe retrieval, whereas all Recall scores improve for recipe-to-image retrieval. For recipe-to-image retrieval, R@1 improves from 85.5% to 88.7% in the 1k setting, corresponding to a gain of 3.2 percentage points. In the 10k setting, R@1 improves from 63.5% to 68.6%, corresponding to a gain of 5.1 percentage points.

We attribute this asymmetric effect across retrieval directions to the characteristics of both the Recipe1M dataset and SMART. Recipe1M pairs recipe texts with food images, but a recipe text usually describes a single dish, whereas a food image may contain multiple dishes. In addition, as shown by the definition in Equation 2, the SMART similarity score is asymmetric. It does not merely select a candidate that is semantically close to the query, but instead favors candidates that contain the query. Under these characteristics, SMART can benefit recipe-to-image retrieval because a candidate image only needs to contain an image region that matches the recipe query. In contrast, for image-to-recipe retrieval, if the query image contains multiple dishes, the scoring behavior effectively favors recipes that cover all dishes in the image. This mismatch may explain why SMART does not improve image-to-recipe retrieval.

V. Conclusion

In this study, we compared MLLM-based embedding models for cross-modal food image–recipe retrieval on Recipe1M. In particular, we applied SIMMER, which had previously been used only with the VLM2Vec family, to GME and Qwen3-VL-Embedding, and evaluated their retrieval performance.

TABLE IV: Effect of fine-tuning. Comparison between zero-shot inference and SIMMER on Recipe1M.

Setting	Model	Training setting	Image-to-Recipe				Recipe-to-Image			
			medR↓	R@1↑	R@5↑	R@10↑	medR↓	R@1↑	R@5↑	R@10↑
1k	GME-2B	Zero-shot	1.6	49.5	77.5	85.0	2.0	43.7	72.1	81.3
		SIMMER	1.0	79.0	95.5	97.8	1.0	76.6	94.4	97.3
	GME-7B	Zero-shot	1.1	53.0	80.5	87.9	2.0	45.8	74.8	83.7
		SIMMER	1.0	82.6	96.4	98.2	1.0	80.4	96.2	98.1
	Qwen3-VL-Embedding-2B	Zero-shot	1.0	57.0	82.3	88.9	2.0	47.2	72.9	81.3
		SIMMER	1.0	83.0	96.7	98.3	1.0	81.2	96.0	97.9
	Qwen3-VL-Embedding-8B	Zero-shot	1.0	60.2	85.0	91.0	1.5	49.6	74.6	83.0
		SIMMER	1.0	86.0	97.5	98.9	1.0	84.6	97.1	98.7
10k	GME-2B	Zero-shot	7.2	23.0	45.3	55.7	11.3	19.0	38.8	48.8
		SIMMER	1.0	53.3	78.1	85.7	1.6	49.7	75.1	83.4
	GME-7B	Zero-shot	6.0	25.5	48.4	58.9	9.3	20.7	41.4	51.6
		SIMMER	1.0	58.8	81.7	88.1	1.0	52.8	78.5	86.2
	Qwen3-VL-Embedding-2B	Zero-shot	4.9	29.2	52.6	62.9	9.1	23.0	42.9	52.0
		SIMMER	1.0	58.7	82.1	88.8	1.0	55.6	79.6	86.6
	Qwen3-VL-Embedding-8B	Zero-shot	4.0	32.6	56.7	66.5	7.8	25.5	45.2	54.3
		SIMMER	1.0	63.9	85.3	90.8	1.0	61.9	84.2	90.1

TABLE V: Effect of batch size and SMART for Qwen3-VL-Embedding-8B on Recipe1M.

Setting	Batch size	SMART	Image-to-Recipe				Recipe-to-Image			
			medR↓	R@1↑	R@5↑	R@10↑	medR↓	R@1↑	R@5↑	R@10↑
1k	128		1.0	86.0	97.5	98.9	1.0	84.6	97.1	98.7
	256		1.0	87.9	98.1	99.1	1.0	86.6	97.8	98.9
	512		1.0	89.3	98.4	99.3	1.0	85.5	97.3	98.9
	512	✓	1.0	88.2	98.0	99.1	1.0	88.7	98.2	99.1
10k	128		1.0	63.9	85.3	90.8	1.0	61.9	84.2	90.1
	256		1.0	67.5	87.6	92.7	1.0	64.6	86.2	91.8
	512		1.0	69.7	89.2	93.7	1.0	63.5	85.2	90.8
	512	✓	1.0	69.3	88.7	93.2	1.0	68.6	88.9	93.5

The experimental results show that both GME and Qwen3-VL-Embedding achieve meaningful zero-shot retrieval performance, while fine-tuning them with SIMMER on Recipe1M yields substantial improvements. These results indicate that the knowledge encoded in general-purpose MLLM-based embedding models is useful for food image–recipe retrieval, but that task-specific fine-tuning remains important for achieving strong performance on a domain-specific benchmark such as Recipe1M.

At the same time, simply applying SIMMER to newer base models did not consistently outperform the original SIMMER (V1-7B). This suggests that the training configuration and prompt design of SIMMER are closely aligned with the assumptions of the VLM2Vec family, and that embedding models with different input formats and model characteristics, such as GME and Qwen3-VL-Embedding, may require more suitable configurations. We therefore further investigated the effect of changing the batch size and incorporating SMART for Qwen3-VL-Embedding-8B. As a result, the model fine-tuned with batch size 512 achieved the strongest R@1 for image-to-recipe retrieval, while the model using SMART at inference time achieved the strongest recipe-to-image performance. In the 10k setting, the SMART variant improved R@1 over SIMMER (V1-7B) from 65.5% to 69.3% for image-to-recipe retrieval and from 61.5% to 68.6% for recipe-to-image retrieval.

The ablation studies show that increasing the batch size from 128 to 256 consistently improves performance, whereas increasing it further to 512 does not lead to consistent gains. We attribute this behavior to the saturation of the benefit from additional in-batch negatives and the relatively larger influence of false negative and noisy negative samples in Recipe1M. We also find that SMART clearly improves recipe-to-image retrieval, while image-to-recipe retrieval is slightly degraded. This asymmetric effect can be explained by both the asymmetric late-interaction score used in SMART and the characteristics of Recipe1M, where a food image may contain multiple dishes whereas a recipe text typically describes a single dish.

Future work includes exploring prompt designs and training configurations tailored to each MLLM-based embedding model. Because offline component-aware data augmentation may increase false negative samples, online augmentation and improved negative sampling strategies may enable more stable contrastive learning.

REFERENCES

[1] F. Li, R. Zhang, H. Zhang, Y. Zhang, B. Li, W. Li, Z. Ma, and C. Li, "LLaVA-NeXT-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models," *arXiv preprint arXiv:2407.07895*, 2024.

[2] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, "Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution," *arXiv preprint arXiv:2409.12191*, 2024.

[3] Z. Jiang, R. Meng, X. Yang, S. Yavuz, Y. Zhou, and W. Chen, "VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks," in *Proceedings of International Conference on Learning Representations*, 2025.

[4] S.-C. Lin, C. Lee, M. Shoeybi, J. Lin, B. Catanzaro, and W. Ping, "MM-Embed: Universal Multimodal Retrieval with Multimodal LLMs," in *Proceedings of International Conference on Learning Representations*, 2025.

[5] K. Gomi and K. Yanai, "SIMMER: Cross-Modal Food Image–Recipe Retrieval via MLLM-Based Embedding," *arXiv preprint arXiv:2604.15628*, 2026.

[6] R. Meng, Z. Jiang, Y. Liu, M. Su, X. Yang, Y. Fu, C. Qin, Z. Chen, R. Xu, C. Xiong, Y. Zhou, W. Chen, and S. Yavuz, "VLM2Vec-V2: Advancing Multimodal Embedding for Videos, Images, and Visual Documents," *arXiv preprint arXiv:2507.04590*, 2025.

[7] X. Zhang, Y. Zhang, W. Xie, M. Li, Z. Dai, D. Long, P. Xie, M. Zhang, W. Li, and M. Zhang, "Bridging Modalities: Improving Universal Multimodal Retrieval by Multimodal Large Language Models," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 9274–9285.

[8] M. Li, Y. Zhang, D. Long, K. Chen, S. Song, S. Bai, Z. Yang, P. Xie, A. Yang, D. Liu, J. Zhou, and J. Lin, "Qwen3-VL-Embedding and Qwen3-VL-Reranker: A Unified Framework for State-of-the-Art Multimodal Retrieval and Ranking," *arXiv preprint arXiv:2601.04720*, 2026.

[9] J. Zhang, H. J. Lee, S. Ganguly, T.-E. Kam, D. Kim, and Y. J. Lee, "Your Embedding Model is SMARTer Than You Think," *arXiv preprint arXiv:2605.24938*, 2026.

[10] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba, "Learning Cross-Modal Embeddings for Cooking Recipes and Food Images," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3068–3076.

[11] H. Wang, D. Sahoo, C. Liu, E.-p. Lim, and S. C. H. Hoi, "Learning Cross-Modal Embeddings With Adversarial Networks for Cooking Recipes and Food Images," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 564–11 573.

[12] H. Fu, R. Wu, C. Liu, and J. Sun, "MCEN: Bridging Cross-Modal Gap between Cooking Recipes and Dish Images with Latent Variable Model," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 558–14 568.

[13] Y. Sugiyama and K. Yanai, "Cross-Modal Recipe Embeddings by Disentangling Recipe Contents and Dish Styles," in *Proceedings of ACM International Conference on Multimedia*, 2021, pp. 2501–2509.

[14] A. Salvador, E. Gundogdu, L. Bazzani, and M. Donoser, "Revamping Cross-Modal Recipe Retrieval with Hierarchical Transformers and Self-supervised Learning," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 470–15 479.

[15] M. Shukor, G. Couairon, A. Grechka, and M. Cord, "Transformer Decoders with MultiModal Regularization for Cross-Modal Food Retrieval," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022, pp. 4566–4577.

[16] J. Yang, J. Chen, and K. Yanai, "Transformer-Based Cross-Modal Recipe Embeddings with Large Batch Training," in *Proceedings of International Conference on Multimedia Modeling*, 2023, pp. 471–482.

[17] ——, "Improving Cross-Modal Recipe Embeddings with Cross Decoder," in *Proceedings of ACM Workshop on Intelligent Cross-Data Analysis and Retrieval*, 2024, pp. 1–4.

[18] M. Shukor, N. Thome, and M. Cord, "Vision and Structured-Language Pretraining for Cross-Modal Food Retrieval," *Computer Vision and Image Understanding*, vol. 247, no. C, 2024.

[19] F. Song, B. Zhu, Y. Hao, and S. Wang, "Enhancing Recipe Retrieval with Foundation Models: A Data Augmentation Perspective," in *Proceedings of European Conference on Computer Vision*, 2024, pp. 111–127.

[20] M. Wahed, X. Zhou, T. Yu, and I. Lourentzou, "Fine-Grained Alignment for Cross-Modal Recipe Retrieval," in *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, 2024, pp. 5572–5581.

[21] Q. Wang, C.-W. Ngo, Y. Cao, and E.-P. Lim, "Mitigating Cross-modal Representation Bias for Multicultural Image-to-Recipe Retrieval," in *Proceedings of ACM International Conference on Multimedia*, 2025, pp. 6223–6231.

[22] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin, X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, and K. Zhu, "Qwen3-VL Technical Report," *arXiv preprint arXiv:2511.21631*, 2025.

[23] A. van den Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," *arXiv preprint arXiv:1807.03748*, 2018.

[24] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in *Proceedings of International Conference on Machine Learning*, 2021, pp. 8748–8763.

[25] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," in *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 39–48.

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of International Conference on Learning Representations*, 2015.

[27] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *Proceedings of International Conference on Learning Representations*, 2022.

[28] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *Proceedings of International Conference on Machine Learning*, vol. 119, 2020, pp. 1597–1607.

[29] C. Wu, F. Wu, and Y. Huang, "Rethinking InfoNCE: How Many Negative Samples Do You Need?" in *Proceedings of International Joint Conference on Artificial Intelligence*, 2022, pp. 2509–2515.