ADAPTIVE GLOBAL AND FINE-GRAINED PERCEPTUAL FUSION FOR MLLM EMBEDDINGS COMPATIBLE WITH HARD NEGATIVE AMPLIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal embeddings serve as a bridge for aligning vision and language, with the two primary implementations—CLIP-based and MLLM-based embedding models—both limited to capturing only global semantic information. Although numerous studies have focused on fine-grained understanding, we observe that complex scenarios currently targeted by MLLM embeddings often involve a hybrid perceptual pattern of both global and fine-grained elements, thus necessitating a compatible fusion mechanism. In this paper, we propose **A**daptive **G**lobal and **F**ine-grained perceptual **F**usion for MLLM **Embed**dings (AGFF-EMBED), a method that prompts the MLLM to generate multiple embeddings focusing on different dimensions of semantic information, which are then adaptively and smoothly aggregated. Furthermore, we adapt AGFF-EMBED with the Explicit Gradient Amplification (EGA) technique to achieve in-batch hard negatives enhancement without requiring fine-grained editing of the dataset. Evaluation on the MMEB and MMVP-VLM benchmarks shows that AGFF-EMBED comprehensively achieves state-of-the-art performance in both general and fine-grained understanding compared to other multimodal embedding models.

1 INTRODUCTION

The learning of embedding space has always been a critical issue in the multimodal field. Early CLIP (Radford et al., 2021) utilizes contrastive learning with massive amounts of image-caption pairs to map images and text into the same semantic space, thereby achieving strong zero-shot transfer capabilities. Subsequent works have proposed variants of CLIP by improving data engineering (Xu et al., 2024), scaling model parameters (Sun et al., 2023; 2024), and optimizing loss functions (Zhai et al., 2023; Tschannen et al., 2025), effectively promoting its maturity. However, CLIP-based models are limited to the simple form of single image-single caption pairs and fail to fully leverage the increasingly powerful capabilities of pretrained multimodal large language models (MLLMs) (Liu et al., 2023; Li et al., 2024; An et al., 2025; Zhu et al., 2025). Recently, VLM2Vec (Jiang et al., 2025c) extracts embeddings from image-text mixed content based on state-of-the-art MLLMs, achieving outstanding performance in a range of complex image-text-to-image-text matching tasks, such as classification, visual question answering, multimodal retrieval, and visual grounding.

One of the core limitations of both CLIP-based models and MLLM embeddings is that contrastive learning tends to focus on global semantic information while being insensitive to fine-grained differences (Zhong et al., 2022; Li et al., 2022; Wang et al., 2024b; 2025a;b; Yu et al., 2024; Maninis et al., 2024; Naeem et al., 2024; Jing et al., 2024; Shi et al., 2025; Jiang et al., 2025a;b; Yang et al., 2025; Hou et al., 2025; Xiao et al., 2025a; Xie et al., 2025). Although for MLLM embeddings, we can add prompts such as "Represent this type of fine-grained information: ..." to provide explicit guidance, this is not a universal solution. As shown in Figure 1, due to the diversity of image-text-to-image-text matching tasks, whether the query and target should focus on global or fine-grained semantic information varies case by case. For a photo of a dog or a brief textual response, global semantic information is sufficient, whereas for images containing multiple objects or screenshots of posters, the recognition of fine-grained semantic information becomes particularly critical. Therefore, we claim that **adaptively integrating global and fine-grained perception is a key issue for MLLM embeddings, rather than focusing solely on fine-grained perception.**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

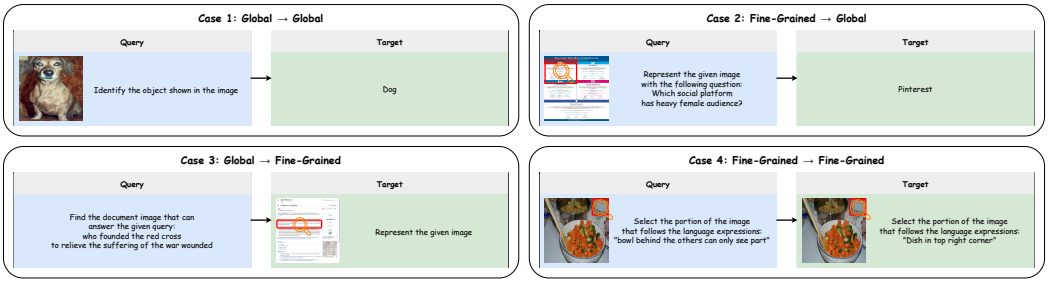

Figure 1: For image-text-to-image-text matching tasks, whether the query and target should focus on global or fine-grained semantic information varies case by case.

Some CLIP-based works make minor modifications to image-caption pairs, such as "a black cat wearing a yellow bow tie" and "a black cat wearing a red bow tie" to construct hard negative samples and improve the model's fine-grained understanding (Yuksekgonul et al., 2022; Patel et al., 2024; Wang et al., 2025c; Zhao et al., 2024). However, manually identifying fine-grained elements and performing large-scale edits is challenging and costly, especially for long-sequence image-text mixed data in MLLM embeddings. A common solution is to identify hard negative samples from existing training batches and amplify the weight of these data points (He et al., 2020; Chen et al., 2020; Meng et al., 2024; Lee et al., 2024; Lan et al., 2025a; Thirukovalluru et al., 2026). Recently, Xue et al. (2025) deeply explore the mechanism of negative samples in backpropagation and explicitly amplify their gradient contributions based on their difficulty. However, only smooth contrastive losses are compatible with explicit gradient amplification techniques, while sharp contrastive losses like MetaEmbed (Xiao et al., 2025b) are not suitable.

In this paper, we propose **A**daptive **G**lobal and **F**ine-grained perceptual **F**usion for MLLM **Embed**dings (AGFF-EMBED). In addition to the original embedding used for global perception, AGFF-EMBED employs learnable prompt tokens to guide the MLLM to spontaneously focus on fine-grained information across different dimensions and generate corresponding embeddings. To adapt to various cases in image-text–to-image-text matching tasks, we compute four types of similarities: global-to-global, fine-grained-to-global, global-to-fine-grained, and fine-grained-to-fine-grained. These similarities are aggregated using smoothed $\mathrm{logsumexp}$, which aims to enable the MLLM to softly favor more suitable patterns. We theoretically demonstrate that this smooth and adaptive fusion approach is compatible with the Explicit Gradient Amplification (EGA) (Xue et al., 2025) technique, thereby achieving hard negative enhancement without the need for additional datasets.

In summary, our contributions are as follows: (1) We propose AGFF-EMBED, a novel multimodal embedding framework that can adaptively integrate global and fine-grained information while being compatible with the EGA technique. (2) We guide the MLLM to generate global and fine-grained embeddings based on learnable prompt tokens, and appropriately aggregate their similarities using $\mathrm{logsumexp}$. (3) We conduct an in-depth analysis of the gradient expansion for AGFF-EMBED and theoretically derive the form of its gradient amplification for hard negatives. (4) The results on the MMEB and MMVP-VLM benchmarks demonstrate the outstanding performance of AGFF-EMBED in both general and fine-grained understanding, while the ablation study further validates the necessity of each module.

2 RELATED WORK

CLIP-based embedding models. CLIP (Radford et al., 2021) employs a dual-tower architecture with separate image and text encoders to generate corresponding embeddings. It performs contrastive learning on a vast collection of image-caption pairs from the web, achieving zero-shot performance on downstream tasks comparable to fully supervised models. Subsequent works further mature the CLIP framework. MetaCLIP (Xu et al., 2024) systematically constructs CLIP's training data and surpasses its performance. EVA-CLIP (Sun et al., 2023) and EVA-CLIP-18B (Sun et al., 2024) scale the parameters of CLIP to 8B and 18B respectively, further pushing the limits of its capabilities. SigLIP (Zhai et al., 2023) replaces the $\mathrm{softmax}$ normalization in CLIP with a $\mathrm{sigmoid}$ loss, decoupling

batch size from the loss function, while SigLIP-2 (Tschannen et al., 2025) builds upon SigLIP by integrating and forming a comprehensive training pipeline.

The limitation of CLIP lies in its focus on global semantic perception while overlooking fine-grained information. NegCLIP (Yuksekgonul et al., 2022), TripletCLIP (Patel et al., 2024), and CLIP-IN (Wang et al., 2025c) construct hard negatives with subtle differences and incorporate them into contrastive learning batches to guide the model in making distinctions. EqSim (Wang et al., 2023) introduces equivariant regularization for similar positive and negative sample pairs, making their relative positions in the embedding space more reasonable. DAC (Doveh et al., 2023) generates additional captions based on LLM-Expander and SAM-Expander, and employs a negative generator to create hard negatives for them. MosaiCLIP (Singh et al., 2023) generates scene graphs for images and captions separately, and perturbs their nodes to construct hard negative sub-graphs. CECLIP (Zhang et al., 2024b) generates hard negatives specifically for different attributes within captions. Additionally, RegionCLIP (Zhong et al., 2022), GLIP (Li et al., 2022), CLIPSelf (Wu et al., 2024), SILC (Naeem et al., 2024), FineCLIP (Jing et al., 2024), and FGCLIP (Xie et al., 2025) achieve fine-grained image-text alignment from the global to the regional level.

MLLM-based embedding models. VLM2Vec (Jiang et al., 2025c) and E5-V (Jiang et al., 2024) make full use of the powerful capabilities of MLLMs to extend the input of embedding models from single images or texts to mixed image-text content. They extract the final hidden state of the last token in the MLLM input as the embedding and perform contrastive learning on their proposed MMEB dataset. Furthermore, GME (Zhang et al., 2024c) and VLM2Vec-V2 (Meng et al., 2025) introduce additional modalities such as visual documents and videos. MegaPairs (Zhou et al., 2025) and mmE5 (Chen et al., 2025b) synthesize high-quality training data for MLLM embeddings. MoCA (Chen et al., 2025a) proposes a two-stage training framework for transforming MLLMs into embedding models, which involves modality-aware continual pre-training and heterogeneous contrastive fine-tuning. Recently, MetaEmbed (Xiao et al., 2025b) introduces a multi-vector retrieval mechanism, making retrieval costs scalable.

Similar to CLIP, a key point for improving the training efficiency and quality of MLLM embeddings lies in fully leveraging hard negatives. UniME (Gu et al., 2025a) filters false negatives and samples hard negatives for each batch during the instruction tuning stage. B3 (Thirukovalluru et al., 2026) constructs batches rich in hard negatives based on the ranking from a teacher model and METIS clustering. According to the similarity between negative samples and queries, i.e., the difficulty level, LLaVE (Lan et al., 2025a) and QQMM (Xue et al., 2025) weight the corresponding terms in the contrastive loss and the corresponding gradients during backpropagation, respectively.

3 METHOD

3.1 ADAPTIVE GLOBAL AND FINE-GRAINED PERCEPTUAL FUSION

To address the different cases presented in Figure 1, we drive the MLLM to adaptively generate and aggregate global and fine-grained embeddings, with its framework illustrated in Figure 2. We append the prompt "The above is the main content. Represent global information in the main content." after the input content to clearly delimit the scope for embedding and guide the MLLM to focus first on global information. Subsequently, similar to VLM2Vec (Jiang et al., 2025c), we use a special token <EMBED> to indicate the generation position of the global embedding $\mathbf{x}_0 \in \mathbb{R}^D$. For further extraction of fine-grained semantic information, we introduce N parallel fine-grained embedding modules after the global embedding module. Each of these modules consists of three components: first, a prompt "Represent this type of fine-grained information in the main content:" to explicitly guide the MLLM; second, M learnable <PROMPT$_{\mathtt{i}\times\mathtt{M}+\mathtt{j}}$> tokens, where $i = 1, \ldots, N$ denotes the module index and $j = 1, \ldots, M$ denotes the prompt index, aiming to allow the MLLM to autonomously determine the types of fine-grained information to focus on; and finally, a special <EMBED$_{\mathtt{i}}$> token to mark the position where the fine-grained embedding $\mathbf{x}_i \in \mathbb{R}^D$ of this module is generated. Thus, all fine-grained modules collectively produce N embeddings $\{\mathbf{x}_i\}_{i=1}^N$, each focusing on different fine-grained semantics.

Now we have the global embedding $\mathbf{x}_0^q$ and fine-grained embeddings $\{\mathbf{x}_i^q\}_{i=1}^N$ for the query, as well as the global embedding $\mathbf{x}_0^t$ and fine-grained embeddings $\{\mathbf{x}_i^t\}_{i=1}^N$ for the target. Corresponding to

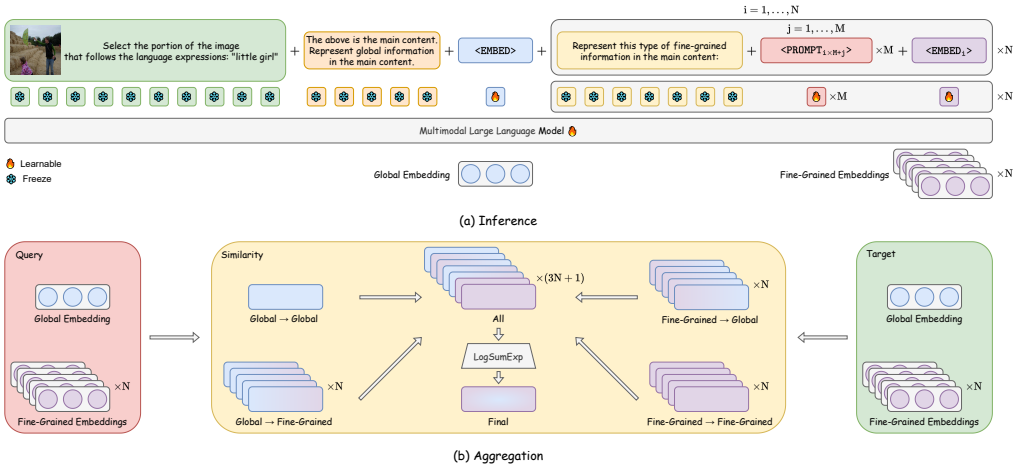

Figure 2: Framework of AGFF-EMBED. (a): MLLM generates a global embedding and N fine-grained embeddings. (b): Four types of similarity corresponding to four perception patterns are computed and aggregated via logsumexp.

the four cases in Figure 1, we compute the following four types of similarity:

$$\begin{cases} s^{\mathrm{g2g}}\left(X^q, X^t\right) = \mathbf{x}_0^q \cdot \mathbf{x}_0^t, \\ s_i^{\mathrm{f2g}}\left(X^q, X^t\right) = \mathbf{x}_i^q \cdot \mathbf{x}_0^t, \quad i = 1, \ldots, N, \\ s_i^{\mathrm{g2f}}\left(X^q, X^t\right) = \mathbf{x}_0^q \cdot \mathbf{x}_i^t, \quad i = 1, \ldots, N, \\ s_i^{\mathrm{f2f}}\left(X^q, X^t\right) = \mathbf{x}_i^q \cdot \mathbf{x}_i^t, \quad i = 1, \ldots, N, \end{cases} \tag{1}$$

where $X^q = [\mathbf{x}_0^q, \mathbf{x}_1^q, \ldots, \mathbf{x}_N^q]^\top \in \mathbb{R}^{(N+1)\times D}$ and $X^t = [\mathbf{x}_0^t, \mathbf{x}_1^t, \ldots, \mathbf{x}_N^t]^\top \in \mathbb{R}^{(N+1)\times D}$ are the complete embeddings for the query and target, respectively.

For a given positive pair, not all types of similarity align with its perceptual pattern, but the more compatible similarities should have higher values. An intuitive aggregation method is to take the maximum of all $(3N + 1)$ similarities (Xiao et al., 2025b). However, perceptual patterns are often not singular, requiring simultaneous attention to multiple global or fine-grained information, with varying degrees of emphasis. Moreover, the maximum function is sharp and lacks favorable gradient properties, making it unsuitable for explicit gradient amplification techniques to enhance hard negatives. Therefore, we adopt the logsumexp function for similarity aggregation, which, as a smooth approximation of the maximum, can adaptively capture multiple dominant perceptual patterns while preserving the differentiability and gradient stability of the loss function:

$$s^{\mathrm{final}} = \log\left[\exp\left(s^{\mathrm{g2g}}\right) + \sum_{i=1}^{N}\exp\left(s_i^{\mathrm{f2g}}\right) + \sum_{i=1}^{N}\exp\left(s_i^{\mathrm{g2f}}\right) + \sum_{i=1}^{N}\exp\left(s_i^{\mathrm{f2f}}\right)\right]. \tag{2}$$

Then we obtain the InfoNCE loss for training:

$$\mathcal{L} = -\log\frac{\exp\left[\phi\left(X^q, X^{t^+}\right)\right]}{\exp\left[\phi\left(X^q, X^{t^+}\right)\right] + \sum_{i=1}^{B-1}\exp\left[\phi\left(X^q, X^{t_i^-}\right)\right]}, \tag{3}$$

where $\phi(X^q, X^t) = \frac{1}{\tau}s^{\mathrm{final}}(X^q, X^t)$, τ is the temperature hyperparameter, B is the batch size, t^+ is the positive target, and t_i^- is the i-th negative target.

3.2 EXPLICIT HARD NEGATIVE GRADIENT AMPLIFICATION FOR AGFF-EMBED

A larger batch size can effectively improve the quality of contrastive learning by introducing richer negative samples. However, GPU memory often limit the increase in batch size, a challenge that is particularly pronounced for MLLM embeddings, as their inputs typically include multiple images

and long text sequences. To address this bottleneck, VLM2Vec (Jiang et al., 2025c) employs the GradCache technique (Gao et al., 2021) during backpropagation, decomposing the gradient of the loss function $\mathcal{L}$ with respect to the MLLM parameters Θ according to the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \Theta} = \frac{\partial \mathcal{L}}{\partial X^q}\frac{\partial X^q}{\partial \Theta} + \frac{\partial \mathcal{L}}{\partial X^{t^+}}\frac{\partial X^{t^+}}{\partial \Theta} + \sum_{i=1}^{B-1}\frac{\partial \mathcal{L}}{\partial X^{t_i^-}}\frac{\partial X^{t_i^-}}{\partial \Theta}. \tag{4}$$

In practice, the gradients of the loss function with respect to each embedding $\left\{\frac{\partial \mathcal{L}}{\partial X^q}, \frac{\partial \mathcal{L}}{\partial X^{t^+}}, \frac{\partial \mathcal{L}}{\partial X^{t_0^-}}, \ldots, \frac{\partial \mathcal{L}}{\partial X^{t_{B-1}^-}}\right\}$ are first computed separately by PyTorch's automatic differentiation and cached after detaching. Then, these gradients are dot-multiplied with their corresponding embeddings and backpropagated, which can be parallelized by splitting the full batch into smaller sub-batches.

Before delving into the analysis of cached gradients, we first denote the normalized probabilities of $s^{\mathrm{g2g}}, s_i^{\mathrm{f2g}}, s_i^{\mathrm{g2f}}, s_i^{\mathrm{f2f}}$ as $w^{\mathrm{g2g}}, w_i^{\mathrm{f2g}}, w_i^{\mathrm{g2f}}, w_i^{\mathrm{f2f}}$, respectively. For example:

$$w^{\mathrm{g2g}} = \frac{\exp\left(s^{\mathrm{g2g}}\right)}{\exp\left(s^{\mathrm{g2g}}\right) + \sum_{j=1}^{N}\exp\left(s_j^{\mathrm{f2g}}\right) + \sum_{j=1}^{N}\exp\left(s_j^{\mathrm{g2f}}\right) + \sum_{j=1}^{N}\exp\left(s_j^{\mathrm{f2f}}\right)}. \tag{5}$$

Additionally, the classification probabilities of the target samples are denoted as:

$$\begin{cases} p^+ = \dfrac{\exp\left[\phi\left(X^q, X^{t^+}\right)\right]}{\exp\left[\phi\left(X^q, X^{t^+}\right)\right] + \sum_{j=1}^{B-1}\exp\left[\phi\left(X^q, X^{t_j^-}\right)\right]}, \\ p_i^- = \dfrac{\exp\left[\phi\left(X^q, X^{t_i^-}\right)\right]}{\exp\left[\phi\left(X^q, X^{t^+}\right)\right] + \sum_{j=1}^{B-1}\exp\left[\phi\left(X^q, X^{t_j^-}\right)\right]}. \end{cases} \tag{6}$$

Then, for AGFF-EMBED, we expand $\frac{\partial \mathcal{L}}{\partial X^q}$ based on Equations (1) to (3):

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \mathbf{x}_0^q} = & \frac{1}{\tau}\sum_{i=1}^{B-1} p_i^- \left[w^{\mathrm{g2g}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_0^{t_i^-} + \sum_{j=1}^{N} w_j^{\mathrm{g2f}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_j^{t_i^-}\right. \\ & \left. -w^{\mathrm{g2g}}\left(X^q, X^{t^+}\right)\mathbf{x}_0^{t^+} - \sum_{j=1}^{N} w_j^{\mathrm{g2f}}\left(X^q, X^{t^+}\right)\mathbf{x}_j^{t^+}\right], \\ \frac{\partial \mathcal{L}}{\partial \mathbf{x}_j^q} = & \frac{1}{\tau}\sum_{i=1}^{B-1} p_i^- \left[w_j^{\mathrm{f2g}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_0^{t_i^-} + w_j^{\mathrm{f2f}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_j^{t_i^-}\right. \\ & \left. -w_j^{\mathrm{f2g}}\left(X^q, X^{t^+}\right)\mathbf{x}_0^{t^+} - w_j^{\mathrm{f2f}}\left(X^q, X^{t^+}\right)\mathbf{x}_j^{t^+}\right],\ j = 1, \ldots, N. \end{cases} \tag{7}$$

Similarly, $\frac{\partial \mathcal{L}}{\partial X^{t^+}}$ is derived as:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \mathbf{x}_0^{t^+}} = & \frac{1}{\tau}\left(p^+ - 1\right)\left[w^{\mathrm{g2g}}\left(X^q, X^{t^+}\right)\mathbf{x}_0^q + \sum_{j=1}^{N} w_j^{\mathrm{f2g}}\left(X^q, X^{t^+}\right)\mathbf{x}_j^q\right], \\ \frac{\partial \mathcal{L}}{\partial \mathbf{x}_j^{t^+}} = & \frac{1}{\tau}\left(p^+ - 1\right)\left[w_j^{\mathrm{g2f}}\left(X^q, X^{t^+}\right)\mathbf{x}_0^q + w_j^{\mathrm{f2f}}\left(X^q, X^{t^+}\right)\mathbf{x}_j^q\right],\ j = 1, \ldots, N. \end{cases} \tag{8}$$

Finally, $\frac{\partial \mathcal{L}}{\partial X^{t_i^-}}$ can be expressed as:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \mathbf{x}_0^{t_i^-}} = & \frac{1}{\tau}p_i^- \left[w^{\mathrm{g2g}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_0^q + \sum_{j=1}^{N} w_j^{\mathrm{f2g}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_j^q\right], \\ \frac{\partial \mathcal{L}}{\partial \mathbf{x}_j^{t_i^-}} = & \frac{1}{\tau}p_i^- \left[w_j^{\mathrm{g2f}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_0^q + w_j^{\mathrm{f2f}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_j^q\right],\ j = 1, \ldots, N. \end{cases} \tag{9}$$

For detailed derivation, please refer to Appendix A. The following conclusions can be drawn from Equations (7) to (9): (1) Negative targets with higher classification probabilities (i.e., harder negatives) contribute more to the gradient. (2) Similarities with higher normalized weights (i.e., more aligned perceptual patterns) contribute more to the gradient. It should be noted that the above metrics rely on the embeddings generated by the MLLM itself. Since the pre-trained MLLM has already distinguished a large number of positive and negative targets, its assessment of the hardness of negative targets is relatively reliable. However, its understanding of global or fine-grained perceptual patterns is learned from scratch, making it difficult to accurately identify dominant perceptual patterns

in the early stages of training. Therefore, at present, we only perform Explicit Gradient Amplification (EGA) (Xue et al., 2025) for hard negatives, reserving its application to perceptual patterns for future research as AGFF-EMBED further matures and scales up.

Similar to Xue et al. (2025), we define the hardness of a negative target based on its similarity difference from the positive target:

$$h_i^- = \exp\left\{\alpha \cdot \left[s^{\text{final}}\left(X^q, X^{t_i^-}\right) - s^{\text{final}}\left(X^q, X^{t^+}\right)\right]\right\}, \tag{10}$$

where α is a hyperparameter that controls the amplification magnitude. Then, the classification probabilities of negative targets are reassigned according to hardness:

$$\hat{p}_i^- = p_i^- \cdot h_i^-, \quad \bar{p}_i^- = \frac{\hat{p}_i^-}{\sum_{i=1}^{B-1} \hat{p}_i^-} \cdot \sum_{i=1}^{B-1} p_i^-. \tag{11}$$

Substituting the updated probabilities $\{\bar{p}_i^-\}_{i=1}^{B-1}$ back into Equations (7) to (9) yields cached gradients amplified by hard negatives.

4 EXPERIMENT

4.1 TRAINING AND EVALUATION ON MMEB

The MMEB (Massive Multimodal Embedding Benchmark) (Jiang et al., 2025c) is a comprehensive benchmark designed to holistically evaluate the capabilities of vision-language models across diverse tasks and data distributions. It comprises 36 datasets aggregated into four core meta-tasks: classification (10 datasets), visual question answering (VQA, 10 datasets), retrieval (12 datasets), and visual grounding (4 datasets). A key feature of MMEB is its explicit evaluation of model generalization by partitioning tasks into in-distribution (IND, 20 datasets) for training and out-of-distribution (OOD, 16 datasets) for evaluation. This structure provides a rigorous testbed for assessing a model's core competency across different visual-language understanding paradigms and its robustness to distributional shifts.

We fine-tune our AGFF-EMBED using the MMEB-train dataset based on the pretrained QQMM (Xue et al., 2025), which is also trained solely on the MMEB-train dataset without external data. AGFF-EMBED incorporates $N = 10$ fine-grained embedding modules, each equipped with $M = 10$ prompt tokens. Training is conducted for 40 steps with a batch size of 1024. In the MLLM, both the vision encoder and the projection layer are frozen, while the LLM is fine-tuned using LoRA with a rank of 8, a scaling factor of 16, a dropout rate of 0.05, and an initial learning rate of 5×10^{-5}. The learnable special tokens, including prompt tokens and embedding tokens, are assigned an initial learning rate of 0.05. The temperature τ and the amplification magnitude α are set to 0.02 and 20, respectively. Additional experimental results regarding the impact of hyperparameters are provided in Appendix C.

For fairness, we compare our AGFF-EMBED with multimodal embedding models that are also trained solely on the MMEB-train dataset. As shown in Table 1 (see Table 5 in Appendix B for detailed results), AGFF-Embed achieves an overall score of 74.9%, surpassing all CLIP-based embedding models and MLLM-based embedding models with comparable model sizes. Specifically, compared to QQMM (Xue et al., 2025) before fine-tuning, AGFF-EMBED improves the overall score by 2.4%. Furthermore, on both the IND and OOD subsets, AGFF-EMBED achieves state-of-the-art performance, demonstrating strong generalization capabilities. Analysis of the time costs for training and inference can be found in Appendix D.

4.2 ZERO-SHOT FINE-GRAINED PERCEPTION EVALUATION

Next, we evaluate the fine-grained comprehension capability of our AGFF-EMBED based on its zero-shot performance on the MMVP-VLM (Multimodal Visual Patterns for Vision Language Models) (Tong et al., 2024) benchmark. The MMVP-VLM benchmark consists of 9 subsets, each containing hard negatives with subtle edits targeting specific attributes, which can be easily distinguished by humans but are challenging for VLMs. These specific attributes include: Orientation (Ori.), Presence (Pre.), State (Sta.), Quantity (Qua.), Spatial (Spa.), Color (Col.), Structural Character (Str.), Text

Table 1: Results on the MMEB benchmark. Results for CLIP, BLIP2, MagicLens, SigLIP, OpenCLIP, UniIR, E5-V, and VLM2Vec are from Jiang et al. (2025c); results for EVA-CLIP, LLaVE, UniME, B3, and QQMM are from Xue et al. (2025); and results for UniME-V2, UME-R1, and ReMatch are from their original papers. For detailed results, please refer to Table 5 in Appendix B.

Model	Size	Per Meta-Task Score				Average Score		
		Classification	VQA	Retrieval	Grounding	IND	OOD	Overall
# of datasets		10	10	12	4	20	16	36
CLIP-Based Embedding Models								
CLIP (Radford et al., 2021)	0.4B	42.8	9.1	53.0	51.8	37.1	38.7	37.8
BLIP2 (Li et al., 2023)	1.2B	27.0	4.2	33.9	47.0	25.3	25.1	25.2
MagicLens (Zhang et al., 2024a)	0.4B	38.8	8.3	35.4	26.0	31.0	23.7	27.8
SigLIP (Zhai et al., 2023)	0.9B	40.3	8.4	31.6	59.5	32.3	38.0	34.8
OpenCLIP (Cherti et al., 2023)	0.4B	47.8	10.9	52.3	53.3	39.3	40.2	39.7
UniIR ($BLIP_{FF}$) (Wei et al., 2024)	unknown	42.1	15.0	60.1	62.2	44.7	40.4	42.8
EVA-CLIP (Sun et al., 2024)	8B	56.0	10.4	49.2	58.9	38.1	45.6	43.7
UniIR ($CLIP_{SF}$) (Wei et al., 2024)	unknown	44.3	16.2	61.8	65.3	47.1	41.7	44.7
MLLM-Based Embedding Models								
E5-V (Jiang et al., 2024)	7B	21.8	4.9	11.5	19.0	14.9	11.5	13.3
VLM2Vec (Jiang et al., 2025c)	7B	61.2	49.9	67.4	86.1	67.5	57.1	62.9
LLaVE (Lan et al., 2025a)	7B	65.7	65.4	70.9	91.9	75.0	64.4	70.3
UniME (Gu et al., 2025a)	7B	66.8	66.6	70.5	90.9	74.6	65.8	70.7
UniME-V2 (Gu et al., 2025b)	7B	65.3	67.6	72.9	90.2	74.8	66.7	71.2
UME-R1 (Lan et al., 2025b)	7B	67.1	69.2	71.9	84.9	76.1	65.1	71.3
B3 (Thirukovalluru et al., 2026)	7B	70.0	66.5	**74.1**	84.6	75.9	67.1	72.0
QQMM (Xue et al., 2025)	7B	69.9	70.0	72.1	86.0	77.2	66.6	72.5
ReMatch (Liu et al., 2025)	7B	65.8	**73.6**	**74.1**	**92.5**	78.1	68.2	73.7
AGFF-EMBED (Ours)	7B	**72.6**	72.6	73.8	90.0	**79.5**	**69.1**	**74.9**

(Tex.), and Camera Perspective (Cam.). Unlike generic multimodal retrieval tasks, the MMVP-VLM benchmark is designed to systematically evaluate the potential of VLMs to go beyond surface-level object recognition and achieve complex perceptual patterns and deep semantic understanding.

Table 2: Zero-shot performance on the MMVP-VLM benchmark. The attributes of the 9 subsets are abbreviated as follows: Orientation (Ori.), Presence (Pre.), State (Sta.), Quantity (Qua.), Spatial (Spa.), Color (Col.), Structural Character (Str.), Text (Tex.), and Camera Perspective (Cam.). Results for CLIP, DIVA, DFN, SigLIP2, and CLIP-IN are from Wang et al. (2025c); and results for LLaVE, UniME, UniME-V2, UME-R1, and QQMM are from our reproduction.

Model	Size	Ori.	Pre.	Sta.	Qua.	Spa.	Col.	Str.	Tex.	Cam.	Average
CLIP-Based Embedding Models											
CLIP (Radford et al., 2021)	0.4B	0.0	20.0	40.0	20.0	6.7	20.0	33.3	6.7	40.0	20.0
DIVA (Wang et al., 2024a)	0.4B	26.7	20.0	33.3	13.3	13.3	46.7	26.7	6.7	40.0	25.2
DFN (Fang et al., 2023)	1B	20.0	26.7	**73.3**	26.7	26.7	66.7	46.7	20.0	53.3	39.9
SigLIP2 (Tschannen et al., 2025)	0.9B	13.3	20.0	60.0	26.7	6.7	80.0	53.3	20.0	40.0	35.6
CLIP-IN (Wang et al., 2025c)	1B	20.0	26.7	**73.3**	26.7	33.3	66.7	46.7	26.7	53.3	41.5
MLLM-Based Embedding Models											
LLaVE (Lan et al., 2025a)	7B	53.3	40.0	66.7	63.3	43.3	73.3	53.3	30.0	**60.0**	53.7
UniME (Gu et al., 2025a)	7B	**60.0**	43.3	**73.3**	63.3	**56.7**	76.7	60.0	43.3	50.0	58.5
UniME-V2 (Gu et al., 2025b)	7B	46.7	**46.7**	**73.3**	56.7	50.0	73.3	63.3	43.3	**60.0**	57.0
UME-R1 (Lan et al., 2025b)	7B	53.3	**46.7**	66.7	60.0	43.3	**83.3**	53.3	40.0	**60.0**	56.3
QQMM (Xue et al., 2025)	7B	53.3	43.3	**73.3**	46.7	36.7	**83.3**	50.0	43.3	56.7	54.1
AGFF-EMBED (Ours)	7B	**60.0**	43.3	**73.3**	**73.3**	46.7	**83.3**	**66.7**	**50.0**	56.7	**61.5**

We endeavor to reproduce the zero-shot performance of recently open-sourced advanced MLLM-based embedding models on the MMVP-VLM benchmark, and compared them with our AGFF-EMBED in Table 2. It is reasonable that all MLLM-based embedding models consistently outperform CLIP-based embedding models due to the superior multimodal understanding potential of MLLMs. Building on this, AGFF-EMBED further achieves an average score of 61.5%, surpassing all baselines, including QQMM (Xue et al., 2025) before fine-tuning. Notably, on 6 subtasks (Ori., Sta., Qua., Col., Str., Tex.), AGFF-EMBED achieves state-of-the-art performance in each respective category. These

results demonstrate that the incorporation of fine-grained perception mechanisms unlocks the deep discriminative ability of MLLM embeddings for specific attributes.

4.3 Ablation Study

Finally, we investigate the necessity of each module in AGFF-Embed. Specifically, we will verify the following points: (1) Is the added perceptual pattern necessary? (2) Is the compatible EGA technique necessary? (3) Is the smooth $\mathrm{logsumexp}$ similarity aggregation necessary? We will conduct ablation studies on the MMEB and MMVP-VLM benchmarks to answer these questions.

Is the added perceptual pattern necessary? As shown in Equations (1) and (2), compared to traditional MLLM embeddings, our AGFF-Embed additionally incorporates fine-grained-to-global similarities $\{s_i^{\mathrm{f2g}}\}_{i=1}^N$, global-to-fine-grained similarities $\{s_i^{\mathrm{g2f}}\}_{i=1}^N$, and fine-grained-to-fine-grained similarities $\{s_i^{\mathrm{f2f}}\}_{i=1}^N$. To validate their necessity, we sequentially discard one type of perceptual pattern. In practice, we set the specified category of similarities to $-\infty$, so that they vanish in the final similarity after $\mathrm{logsumexp}$ aggregation, and the corresponding normalized probabilities $w_i^{\mathrm{f2g}}, w_i^{\mathrm{g2f}}, w_i^{\mathrm{f2f}}$ in the amplified gradients become 0. Other training configurations remain consistent with the full AGFF-Embed for fairness.

Table 3: Ablation study results on the MMEB benchmark.

Model	Per Meta-Task Score				Average Score		
	Classification	VQA	Retrieval	Grounding	IND	OOD	Overall
# of datasets	10	10	12	4	20	16	36
AGFF-Embed (full model)	72.6	**72.6**	**73.8**	90.0	**79.5**	**69.1**	**74.9**
Perceptual Pattern Exclusion							
w/o fine-grained → global	72.0	71.9	71.7	**90.1**	78.2	68.5	73.9
w/o global → fine-grained	**72.7**	71.7	72.6	89.8	78.5	**69.1**	74.3
w/o fine-grained → fine-grained	71.9	72.0	72.9	89.2	78.7	68.6	74.2
EGA Exclusion							
w/o EGA	72.5	72.2	73.4	89.3	79.0	**69.1**	74.6
$\mathrm{Logsumexp}$ *Similarity Aggregation Exclusion*							
$\max$ aggregation	69.2	68.2	69.4	88.1	74.2	67.2	71.1
mean-$\max$ aggregation (Xiao et al., 2025b)	71.0	70.3	71.5	87.7	76.9	67.6	72.8

Table 4: Ablation study results on the MMVP-VLM benchmark.

Model	Ori.	Pre.	Sta.	Qua.	Spa.	Col.	Str.	Tex.	Cam.	Average
AGFF-Embed (full model)	60.0	43.3	73.3	73.3	46.7	83.3	**66.7**	50.0	56.7	**61.5**
Perceptual Pattern Exclusion										
w/o fine-grained → global	46.7	33.3	73.3	**76.7**	46.7	**86.7**	50.0	53.3	**63.3**	58.9
w/o global → fine-grained	60.0	46.7	**76.7**	73.3	50.0	**86.7**	60.0	40.0	56.7	61.1
w/o fine-grained → fine-grained	60.0	43.3	70.0	66.7	**53.3**	80.0	60.0	43.3	60.0	59.6
EGA Exclusion										
w/o EGA	**63.3**	**50.0**	70.0	63.3	**53.3**	80.0	63.3	36.7	60.0	60.0
$\mathrm{Logsumexp}$ *Similarity Aggregation Exclusion*										
$\max$ aggregation	36.7	46.7	73.3	60.0	40.0	**86.7**	60.0	**56.7**	50.0	56.7
mean-$\max$ aggregation (Xiao et al., 2025b)	53.3	43.3	73.3	56.7	40.0	80.0	53.3	46.7	46.7	54.8

As shown in Table 3, due to the richness of test scenarios in the MMEB benchmark, the exclusion of any perceptual pattern leads to a decrease in overall performance. However, when we turn to the zero-shot performance on the MMVP-VLM benchmark, as shown in Table 4, an interesting phenomenon emerges: although the average score of the full AGFF-Embed surpasses that of AGFF-Embed without fine-grained-to global or fine-grained-to-fine-grained perception, it is on par with AGFF-Embed without global-to-fine-grained perception. This is reasonable when we delve into the characteristics of the MMVP-VLM benchmark: all queries are images containing key fine-grained

information, while all targets are brief textual captions. This unique evaluation scenario places greater emphasis on the MLLM's understanding of fine-grained details in queries and global information in targets, thereby making the exclusion of global-to-fine-grained perception inconsequential to overall performance.

Is the compatible EGA technique necessary? Explicit Gradient Amplification (EGA) (Xue et al., 2025) delves into the gradient expansion form of contrastive learning and leverages the unique GradCache mechanism of MLLM embeddings to achieve hard negative enhancement during backpropagation. We have theoretically adapted the EGA technique to our AGFF-EMBED in Section 3.2, and now we will evaluate its benefits through an ablation study. Specifically, we set the amplification magnitude α in Equation (10) to 0, which means the amplified gradients degenerate to the original gradients. All training configurations remain consistent with the full AGFF-Embed except for the exclusion of EGA. As shown in Tables 3 and 4, on the MMEB and MMVP-VLM benchmarks, although AGFF-EMBED without EGA achieves relatively good performance, it still marginally lags behind the full model.

Is the smooth $\mathrm{logsumexp}$ **similarity aggregation necessary?** As a smooth approximation of the maximum, $\mathrm{logsumexp}$ aims to adaptively capture multiple dominant perceptual patterns. To verify the superiority of this similarity aggregation approach, we replace it with the $\max$ operation for comparison:

$$s_{\max}^{\text{final}} = \max\{s^{\text{g2g}}, \max_{1\le i\le N} s_i^{\text{f2g}}, \max_{1\le i\le N} s_i^{\text{g2f}}, \max_{1\le i\le N} s_i^{\text{f2f}}\}, \tag{12}$$

where $s^{\text{g2g}}, \{s_i^{\text{f2g}}\}_{i=1}^N, \{s_i^{\text{g2f}}\}_{i=1}^N, \{s_i^{\text{f2f}}\}_{i=1}^N$ are defined by Equation (1). Recently, MetaEmbed (Xiao et al., 2025b) has proposed another method for multi-embedding aggregation, which we refer to as mean-max aggregation based on its formulation:

$$s_{\text{mean-max}}^{\text{final}}\left(X^q, X^t\right) = \sum_{i=0}^{N} \max_{0\le j\le N} \mathbf{x}_i^q \cdot \mathbf{x}_j^t, \tag{13}$$

where the definitions of X^q and X^t are consistent with those in Equation (1). Note that we do not directly compare our results with those reported by MetaEmbed (Xiao et al., 2025b), as it utilizes additional training data beyond the MMEB-train dataset. Instead, we only conduct ablation analysis focusing on its aggregation method.

Both $\max$ and mean-max aggregations lack favorable gradient properties and are not compatible with the EGA technique. Therefore, PyTorch's standard backpropagation is employed to update parameters, with other training configurations kept consistent with the full AGFF-EMBED. As shown in Tables 3 and 4, even compared to AGFF-EMBED without EGA, $\max$ and mean-max aggregation methods still perform significantly worse. Furthermore, if the EGA mechanism is introduced, this gap is further widened, which demonstrates the comprehensive superiority of $\mathrm{logsumexp}$ aggregation both in terms of the rationality of perceptual fusion and its compatibility with EGA.

5 CONCLUSION

In this paper, we propose the AGFF-EMBED framework, which aims to unlock the ability of MLLM embeddings to integrate multi-dimensional perception in complex scenarios. Leveraging the powerful language understanding of MLLMs, we guide the generation of global and fine-grained embeddings through clear prompt templates and learnable prompt tokens. Corresponding to four perception patterns, we compute their respective similarities and aggregate them via $\mathrm{logsumexp}$. This aggregation approach exhibits two distinct characteristics: it simultaneously captures multiple dominant perceptual patterns while maintaining an elegant gradient form. Exploiting the latter advantage, we adapt AGFF-EMBED with EGA technique to achieve hard negative enhancement within batches, eliminating the need for additional datasets. Evaluation results on the MMEB and MMVP-VLM benchmarks comprehensively demonstrate the state-of-the-art performance of AGFF-EMBED across both global and fine-grained understanding capabilities. The effectiveness of each module is further validated through ablation studies. In the future, we anticipate that AGFF-EMBED will serve as a new paradigm for MLLM embeddings.

AI USE STATEMENT

We used generative AI tools only to aid or polish the writing of this paper. The tools suggested revisions to improve clarity and readability, which the human authors reviewed and decided whether to adopt. The authors take full responsibility for the final content of this paper, including all AI-assisted writing.

REFERENCES

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, et al. LLaVA-OneVision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

Haonan Chen, Hong Liu, Yuping Luo, Liang Wang, Nan Yang, Furu Wei, and Zhicheng Dou. MoCa: Modality-aware continual pre-training makes better bidirectional multimodal embeddings. *arXiv preprint arXiv:2506.23115*, 2025a.

Haonan Chen, Liang Wang, Nan Yang, Yutao Zhu, Ziliang Zhao, Furu Wei, and Zhicheng Dou. mmE5: Improving multimodal multilingual embeddings via high-quality synthetic data. *arXiv preprint arXiv:2502.08468*, 2025b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607. PmLR, 2020.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

Sivan Doveh, Assaf Arbelle, Sivan Harary, Roei Herzig, Donghyun Kim, Paola Cascante-Bonilla, Amit Alfassy, Rameswar Panda, Raja Giryes, Rogerio Feris, et al. Dense and aligned captions (DAC) promote compositional reasoning in VL models. *Advances in Neural Information Processing Systems*, 36:76137–76150, 2023.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. Scaling deep contrastive learning batch size under memory limited setup. *arXiv preprint arXiv:2101.06983*, 2021.

Tiancheng Gu, Kaicheng Yang, Ziyong Feng, Xingjun Wang, Yanzhao Zhang, Dingkun Long, Yingda Chen, Weidong Cai, and Jiankang Deng. Breaking the modality barrier: Universal embedding learning with multimodal LLMs. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 2860–2869, 2025a.

Tiancheng Gu, Kaicheng Yang, Kaichen Zhang, Xiang An, Ziyong Feng, Yueyi Zhang, Weidong Cai, Jiankang Deng, and Lidong Bing. UniME-V2: MLLM-as-a-judge for universal multimodal embedding learning. *arXiv preprint arXiv:2510.13515*, 2025b.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Bohan Hou, Haoqiang Lin, Xuemeng Song, Haokun Wen, Meng Liu, Yupeng Hu, and Xiangyu Zhao. FiRE: Enhancing MLLMs with fine-grained context learning for complex image retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 803–812, 2025.

Jiachen Jiang, Jinxin Zhou, Bo Peng, Xia Ning, and Zhihui Zhu. Analyzing fine-grained alignment and enhancing vision understanding in multimodal language models. *arXiv preprint arXiv:2505.17316*, 2025a.

Song Jiang, Lijuan Jiao, Yuguang Li, Pei You, Hui Li, Ran Yang, Feng Zhu, Sehwan Ki, and Hyong-Euk Lee. Exploring MLLM in fine-grained visual quality comparison with quality token. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 3516–3526. IEEE, 2025b.

Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-V: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training vision-language models for massive multimodal embedding tasks. In *ICLR*, 2025c.

Dong Jing, Xiaolong He, Yutian Luo, Nanyi Fei, Guoxing Yang, Wei Wei, Huiwen Zhao, and Zhiwu Lu. FineCLIP: Self-distilled region-based CLIP for better fine-grained understanding. *Advances in Neural Information Processing Systems*, 37:27896–27918, 2024.

Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. LLaVE: Large language and vision embedding models with hardness-weighted contrastive learning. *arXiv preprint arXiv:2503.04812*, 2025a.

Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. UME-R1: Exploring reasoning-driven generative multimodal embeddings. *arXiv preprint arXiv:2511.00405*, 2025b.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved techniques for training LLMs as generalist embedding models. *arXiv preprint arXiv:2405.17428*, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pp. 19730–19742. PMLR, 2023.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10965–10975, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.

Qianying Liu, Xiao Liang, Zhiqiang Zhang, Zhongfei Qing, Fengfan Zhou, Yibo Chen, Xu Tang, Yao Hu, and Paul Henderson. ReMatch: Boosting representation through matching for multimodal retrieval. *arXiv preprint arXiv:2511.19278*, 2025.

Kevis-Kokitsi Maninis, Kaifeng Chen, Soham Ghosh, Arjun Karpur, Koert Chen, Ye Xia, Bingyi Cao, Daniel Salz, Guangxing Han, Jan Dlabal, et al. TIPS: Text-image pretraining with spatial awareness. *arXiv preprint arXiv:2410.16512*, 2024.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. Sfrembedding-mistral: enhance text retrieval with transfer learning. *Salesforce AI Research Blog*, 3(6):2, 2024.

Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, et al. VLM2Vec-V2: Advancing multimodal embedding for videos, images, and visual documents. *arXiv preprint arXiv:2507.04590*, 2025.

Muhammad Ferjad Naeem, Yongqin Xian, Xiaohua Zhai, Lukas Hoyer, Luc Van Gool, and Federico Tombari. SILC: Improving vision language pretraining with self-distillation. In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024.

Maitreya Patel, Abhiram Kusumba, Sheng Cheng, Changhoon Kim, Tejas Gokhale, Chitta Baral, and Yezhou Yang. TripletCLIP: Improving compositional reasoning of CLIP via synthetic vision-language negatives. *Advances in neural information processing systems*, 37:32731–32760, 2024.

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

Yuheng Shi, Xiaohuan Pei, Minjing Dong, and Chang Xu. Catching the details: Self-distilled roi predictors for fine-grained MLLM perception. *arXiv preprint arXiv:2509.16944*, 2025.

Harman Singh, Pengchuan Zhang, Qifan Wang, Mengjiao Wang, Wenhan Xiong, Jingfei Du, and Yu Chen. Coarse-to-fine contrastive learning in image-text-graph space for improved vision-language compositionality. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 869–893, 2023.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: Improved training techniques for CLIP at scale. *arXiv preprint arXiv:2303.15389*, 2023.

Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. EVA-CLIP-18B: Scaling CLIP to 18 billion parameters. *arXiv preprint arXiv:2402.04252*, 2024.

Raghuveer Thirukovalluru, Rui Meng, Ye Liu, Mingyi Su, Ping Nie, Semih Yavuz, Yingbo Zhou, Wenhu Chen, Bhuwan Dhingra, et al. Breaking the batch barrier (B3) of contrastive learning via smart batch mining. *Advances in Neural Information Processing Systems*, 38:23218–23241, 2026.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9568–9578, 2024.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Tan Wang, Kevin Lin, Linjie Li, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. Equivariant similarity for vision-language foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11998–12008, 2023.

Wei Wang, Zhaowei Li, Qi Xu, Linfeng Li, YiQing Cai, Botian Jiang, Hang Song, Xingcan Hu, Pengyu Wang, and Li Xiao. Advancing fine-grained visual understanding with multi-scale alignment in multi-modal models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14271–14290, 2025a.

Wenxuan Wang, Quan Sun, Fan Zhang, Yepeng Tang, Jing Liu, and Xinlong Wang. Diffusion feedback helps CLIP see better. *arXiv preprint arXiv:2407.20171*, 2024a.

Yeyuan Wang, Dehong Gao, Bin Li, Rujiao Long, Lei Yi, Xiaoyan Cai, Libin Yang, Jinxia Zhang, Shanqing Yu, and Qi Xuan. CoF: Coarse to fine-grained image understanding for multi-modal large language models. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025b.

Ziqi Wang, Chen Zhu, Zhi Zheng, Xinhang Li, Tong Xu, Yongyi He, Qi Liu, Ying Yu, and Enhong Chen. Granular entity mapper: Advancing fine-grained multimodal named entity recognition and grounding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3211–3226, 2024b.

Ziteng Wang, Siqi Yang, Limeng Qiao, and Lin Ma. VITRIX-CLIPIN: Enhancing fine-grained visual understanding in CLIP via instruction editing data and long captions. *arXiv preprint arXiv:2508.02329*, 2025c.

Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. UniIR: Training and benchmarking universal multimodal information retrievers. In *European Conference on Computer Vision*, pp. 387–404. Springer, 2024.

Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Change Loy. CLIPSelf: Vision transformer distills itself for open-vocabulary dense prediction. In *ICLR*, 2024.

Rui Xiao, Sanghwan Kim, Mariana-Iuliana Georgescu, Zeynep Akata, and Stephan Alaniz. FLAIR: VLM with fine-grained language-informed image representations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24884–24894, 2025a.

Zilin Xiao, Qi Ma, Mengting Gu, Chun-cheng Jason Chen, Xintao Chen, Vicente Ordonez, and Vijai Mohan. MetaEmbed: Scaling multimodal retrieval at test-time with flexible late interaction. *arXiv preprint arXiv:2509.18095*, 2025b.

Chunyu Xie, Bin Wang, Fanjing Kong, Jincheng Li, Dawei Liang, Gengshen Zhang, Dawei Leng, and Yuhui Yin. FG-CLIP: Fine-grained visual and textual alignment. *arXiv preprint arXiv:2505.05071*, 2025.

Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying CLIP data. In *ICLR*, 2024.

Youze Xue, Dian Li, and Gang Liu. Improve multi-modal embedding learning via explicit hard negative gradient amplifying. *arXiv preprint arXiv:2506.02020*, 2025.

Xuzheng Yang, Junzhuo Liu, Peng Wang, Guoqing Wang, Yang Yang, and Heng Tao Shen. New dataset and methods for fine-grained compositional referring expression comprehension via specialist-MLLM collaboration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Ya-Qi Yu, Minghui Liao, Jihao Wu, Yongxin Liao, Xiaoyu Zheng, and Wei Zeng. TextHawk: Exploring efficient fine-grained perception of multimodal large language models. *arXiv preprint arXiv:2404.09204*, 2024.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv preprint arXiv:2210.01936*, 2022.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

Kai Zhang, Yi Luan, Hexiang Hu, Kenton Lee, Siyuan Qiao, Wenhu Chen, Yu Su, and Ming-Wei Chang. MagicLens: Self-supervised image retrieval with open-ended instructions. *arXiv preprint arXiv:2403.19651*, 2024a.

Le Zhang, Rabiul Awal, and Aishwarya Agrawal. Contrasting intra-modal and ranking cross-modal hard negatives to enhance visio-linguistic compositional understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13774–13784, 2024b.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. GME: Improving universal multimodal retrieval by multimodal LLMs. *arXiv preprint arXiv:2412.16855*, 2024c.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. UltraEdit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024.

Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. RegionCLIP: Region-based language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16793–16803, 2022.

Junjie Zhou, Yongping Xiong, Zheng Liu, Ze Liu, Shitao Xiao, Yueze Wang, Bo Zhao, Chen Jason Zhang, and Defu Lian. MegaPairs: Massive data synthesis for universal multimodal retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19076–19095, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Detailed Derivation of Cached Gradients

To derive the final cached gradients, we first compute the derivatives of the intermediate variables. According to Equation (1), we have:

$$\frac{\partial s^{\mathrm{g2g}}\left(X^q, X^t\right)}{\partial \mathbf{x}_0^q} = \mathbf{x}_0^t, \quad \frac{\partial s^{\mathrm{g2g}}\left(X^q, X^t\right)}{\partial \mathbf{x}_0^t} = \mathbf{x}_0^q, \tag{14}$$

$$\frac{\partial s_i^{\mathrm{f2g}}\left(X^q, X^t\right)}{\partial \mathbf{x}_i^q} = \mathbf{x}_0^t, \quad \frac{\partial s_i^{\mathrm{f2g}}\left(X^q, X^t\right)}{\partial \mathbf{x}_0^t} = \mathbf{x}_i^q, \quad i = 1, \ldots, N, \tag{15}$$

$$\frac{\partial s_i^{\mathrm{g2f}}\left(X^q, X^t\right)}{\partial \mathbf{x}_0^q} = \mathbf{x}_i^t, \quad \frac{\partial s_i^{\mathrm{g2f}}\left(X^q, X^t\right)}{\partial \mathbf{x}_i^t} = \mathbf{x}_0^q, \quad i = 1, \ldots, N, \tag{16}$$

$$\frac{\partial s_i^{\mathrm{f2f}}\left(X^q, X^t\right)}{\partial \mathbf{x}_i^q} = \mathbf{x}_i^t, \quad \frac{\partial s_i^{\mathrm{f2f}}\left(X^q, X^t\right)}{\partial \mathbf{x}_i^t} = \mathbf{x}_i^q, \quad i = 1, \ldots, N. \tag{17}$$

From Equation (2), we obtain:

$$\frac{\partial s^{\mathrm{final}}}{\partial s^{\mathrm{g2g}}} = \frac{\exp\left(s^{\mathrm{g2g}}\right)}{\exp\left(s^{\mathrm{g2g}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{f2g}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{g2f}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{f2f}}\right)} = w^{\mathrm{g2g}}, \tag{18}$$

$$\frac{\partial s^{\mathrm{final}}}{\partial s_i^{\mathrm{f2g}}} = \frac{\exp\left(s_i^{\mathrm{f2g}}\right)}{\exp\left(s^{\mathrm{g2g}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{f2g}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{g2f}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{f2f}}\right)} = w_i^{\mathrm{f2g}}, \tag{19}$$

$$\frac{\partial s^{\mathrm{final}}}{\partial s_i^{\mathrm{g2f}}} = \frac{\exp\left(s_i^{\mathrm{g2f}}\right)}{\exp\left(s^{\mathrm{g2g}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{f2g}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{g2f}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{f2f}}\right)} = w_i^{\mathrm{g2f}}, \tag{20}$$

$$\frac{\partial s^{\mathrm{final}}}{\partial s_i^{\mathrm{f2f}}} = \frac{\exp\left(s_i^{\mathrm{f2f}}\right)}{\exp\left(s^{\mathrm{g2g}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{f2g}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{g2f}}\right) + \sum_{j=1}^N \exp\left(s_j^{\mathrm{f2f}}\right)} = w_i^{\mathrm{f2f}}, \tag{21}$$

which is precisely the definition of normalized probabilities as exemplified by Equation (5).

Noting that $\phi\left(X^q, X^t\right) = \frac{1}{\tau} s^{\mathrm{final}}\left(X^q, X^t\right)$, then we calculate its gradient:

$$\begin{aligned}
\frac{\partial \phi\left(X^q, X^t\right)}{\partial \mathbf{x}_0^q} &= \frac{1}{\tau}\left[\frac{\partial s^{\mathrm{final}}\left(X^q, X^t\right)}{\partial s^{\mathrm{g2g}}\left(X^q, X^t\right)} \frac{\partial s^{\mathrm{g2g}}\left(X^q, X^t\right)}{\partial \mathbf{x}_0^q} + \sum_{i=1}^N \frac{\partial s^{\mathrm{final}}\left(X^q, X^t\right)}{\partial s_i^{\mathrm{g2f}}\left(X^q, X^t\right)} \frac{\partial s_i^{\mathrm{g2f}}\left(X^q, X^t\right)}{\partial \mathbf{x}_0^q}\right] \\
&= \frac{1}{\tau}\left[w^{\mathrm{g2g}}\left(X^q, X^t\right) \mathbf{x}_0^t + \sum_{i=1}^N w_i^{\mathrm{g2f}}\left(X^q, X^t\right) \mathbf{x}_i^t\right],
\end{aligned} \tag{22}$$

$$\begin{aligned}
\frac{\partial \phi\left(X^q, X^t\right)}{\partial \mathbf{x}_i^q} &= \frac{1}{\tau}\left[\frac{\partial s^{\mathrm{final}}\left(X^q, X^t\right)}{\partial s_i^{\mathrm{f2g}}\left(X^q, X^t\right)} \frac{\partial s_i^{\mathrm{f2g}}\left(X^q, X^t\right)}{\partial \mathbf{x}_i^q} + \frac{\partial s^{\mathrm{final}}\left(X^q, X^t\right)}{\partial s_i^{\mathrm{f2f}}\left(X^q, X^t\right)} \frac{\partial s_i^{\mathrm{f2f}}\left(X^q, X^t\right)}{\partial \mathbf{x}_i^q}\right] \\
&= \frac{1}{\tau}\left[w_i^{\mathrm{f2g}}\left(X^q, X^t\right) \mathbf{x}_0^t + w_i^{\mathrm{f2f}}\left(X^q, X^t\right) \mathbf{x}_i^t\right], \quad i = 1, \ldots, N,
\end{aligned} \tag{23}$$

$$\begin{aligned}
\frac{\partial \phi\left(X^q, X^t\right)}{\partial \mathbf{x}_0^t} &= \frac{1}{\tau}\left[\frac{\partial s^{\mathrm{final}}\left(X^q, X^t\right)}{\partial s^{\mathrm{g2g}}\left(X^q, X^t\right)} \frac{\partial s^{\mathrm{g2g}}\left(X^q, X^t\right)}{\partial \mathbf{x}_0^t} + \sum_{i=1}^N \frac{\partial s^{\mathrm{final}}\left(X^q, X^t\right)}{\partial s_i^{\mathrm{f2g}}\left(X^q, X^t\right)} \frac{\partial s_i^{\mathrm{f2g}}\left(X^q, X^t\right)}{\partial \mathbf{x}_0^t}\right] \\
&= \frac{1}{\tau}\left[w^{\mathrm{g2g}}\left(X^q, X^t\right) \mathbf{x}_0^q + \sum_{i=1}^N w_i^{\mathrm{f2g}}\left(X^q, X^t\right) \mathbf{x}_i^q\right],
\end{aligned} \tag{24}$$

$$\begin{aligned}
\frac{\partial \phi\left(X^q, X^t\right)}{\partial \mathbf{x}_i^t} &= \frac{1}{\tau}\left[\frac{\partial s^{\mathrm{final}}\left(X^q, X^t\right)}{\partial s_i^{\mathrm{g2f}}\left(X^q, X^t\right)} \frac{\partial s_i^{\mathrm{g2f}}\left(X^q, X^t\right)}{\partial \mathbf{x}_i^t} + \frac{\partial s^{\mathrm{final}}\left(X^q, X^t\right)}{\partial s_i^{\mathrm{f2f}}\left(X^q, X^t\right)} \frac{\partial s_i^{\mathrm{f2f}}\left(X^q, X^t\right)}{\partial \mathbf{x}_i^t}\right] \\
&= \frac{1}{\tau}\left[w_i^{\mathrm{g2f}}\left(X^q, X^t\right) \mathbf{x}_0^q + w_i^{\mathrm{f2f}}\left(X^q, X^t\right) \mathbf{x}_i^q\right], \quad i = 1, \ldots, N.
\end{aligned} \tag{25}$$

We rewrite the InfoNCE loss in Equation (3) as:

$$\mathcal{L} = -\phi\left(X^q, X^{t^+}\right) + \log\left\{\exp\left[\phi\left(X^q, X^{t^+}\right)\right] + \sum_{i=1}^{B-1}\exp\left[\phi\left(X^q, X^{t_i^-}\right)\right]\right\}. \tag{26}$$

Finally, the cached gradients are expanded as follows:

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\mathbf{x}_0^q} &= -\frac{\partial\phi\left(X^q, X^{t^+}\right)}{\partial\mathbf{x}_0^q} + p^+\frac{\partial\phi\left(X^q, X^{t^+}\right)}{\partial\mathbf{x}_0^q} + \sum_{i=1}^{B-1} p_i^-\frac{\partial\phi\left(X^q, X^{t_i^-}\right)}{\partial\mathbf{x}_0^q} \\
&= \sum_{i=1}^{B-1} p_i^-\left[\frac{\partial\phi\left(X^q, X^{t_i^-}\right)}{\partial\mathbf{x}_0^q} - \frac{\partial\phi\left(X^q, X^{t^+}\right)}{\partial\mathbf{x}_0^q}\right] \\
&= \frac{1}{\tau}\sum_{i=1}^{B-1} p_i^-\left[w^{\mathrm{g2g}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_0^{t_i^-} + \sum_{j=1}^{N} w_j^{\mathrm{g2f}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_j^{t_i^-}\right. \\
&\quad \left. - w^{\mathrm{g2g}}\left(X^q, X^{t^+}\right)\mathbf{x}_0^{t^+} - \sum_{j=1}^{N} w_j^{\mathrm{g2f}}\left(X^q, X^{t^+}\right)\mathbf{x}_j^{t^+}\right],
\end{aligned} \tag{27}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\mathbf{x}_j^q} &= -\frac{\partial\phi\left(X^q, X^{t^+}\right)}{\partial\mathbf{x}_j^q} + p^+\frac{\partial\phi\left(X^q, X^{t^+}\right)}{\partial\mathbf{x}_j^q} + \sum_{i=1}^{B-1} p_i^-\frac{\partial\phi\left(X^q, X^{t_i^-}\right)}{\partial\mathbf{x}_j^q} \\
&= \sum_{i=1}^{B-1} p_i^-\left[\frac{\partial\phi\left(X^q, X^{t_i^-}\right)}{\partial\mathbf{x}_j^q} - \frac{\partial\phi\left(X^q, X^{t^+}\right)}{\partial\mathbf{x}_j^q}\right] \\
&= \frac{1}{\tau}\sum_{i=1}^{B-1} p_i^-\left[w_j^{\mathrm{f2g}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_0^{t_i^-} + w_j^{\mathrm{f2f}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_j^{t_i^-}\right. \\
&\quad \left. - w_j^{\mathrm{f2g}}\left(X^q, X^{t^+}\right)\mathbf{x}_0^{t^+} - w_j^{\mathrm{f2f}}\left(X^q, X^{t^+}\right)\mathbf{x}_j^{t^+}\right],\ j = 1, \ldots, N,
\end{aligned} \tag{28}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\mathbf{x}_0^{t^+}} &= -\frac{\partial\phi\left(X^q, X^{t^+}\right)}{\partial\mathbf{x}_0^{t^+}} + p^+\frac{\partial\phi\left(X^q, X^{t^+}\right)}{\partial\mathbf{x}_0^{t^+}} \\
&= \frac{1}{\tau}\left(p^+ - 1\right)\left[w^{\mathrm{g2g}}\left(X^q, X^{t^+}\right)\mathbf{x}_0^q + \sum_{j=1}^{N} w_j^{\mathrm{f2g}}\left(X^q, X^{t^+}\right)\mathbf{x}_j^q\right],
\end{aligned} \tag{29}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\mathbf{x}_j^{t^+}} &= -\frac{\partial\phi\left(X^q, X^{t^+}\right)}{\partial\mathbf{x}_j^{t^+}} + p^+\frac{\partial\phi\left(X^q, X^{t^+}\right)}{\partial\mathbf{x}_j^{t^+}} \\
&= \frac{1}{\tau}\left(p^+ - 1\right)\left[w_j^{\mathrm{g2f}}\left(X^q, X^{t^+}\right)\mathbf{x}_0^q + w_j^{\mathrm{f2f}}\left(X^q, X^{t^+}\right)\mathbf{x}_j^q\right], \quad j = 1, \ldots, N,
\end{aligned} \tag{30}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\mathbf{x}_0^{t_i^-}} &= p_i^-\frac{\partial\phi\left(X^q, X^{t_i^-}\right)}{\partial\mathbf{x}_0^{t_i^-}} \\
&= \frac{1}{\tau}p_i^-\left[w^{\mathrm{g2g}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_0^q + \sum_{j=1}^{N} w_j^{\mathrm{f2g}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_j^q\right],
\end{aligned} \tag{31}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\mathbf{x}_j^{t_i^-}} &= p_i^-\frac{\partial\phi\left(X^q, X^{t_i^-}\right)}{\partial\mathbf{x}_j^{t_i^-}} \\
&= \frac{1}{\tau}p_i^-\left[w_j^{\mathrm{g2f}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_0^q + w_j^{\mathrm{f2f}}\left(X^q, X^{t_i^-}\right)\mathbf{x}_j^q\right], \quad j = 1, \ldots, N,
\end{aligned} \tag{32}$$

where the classification probabilities of the target samples p^+ and p_i^- are defined by Equation (6). We utilize the following fact during the derivation:

$$p^+ + \sum_{i=1}^{B-1} p_i^- = 1. \tag{33}$$

B Detailed Results on MMEB

Table 5: Detailed results on the MMEB benchmark. The OOD datasets are highlighted with a yellow background.

	CLIP	OpenCLIP	SigLIP	BLIP2	MagicLens	E5-V	UniIR	VLM2Vec	UniME-V2	UME-R1	ReMatch	AGFF-Embed
Classification (10 tasks)												
ImageNet-1K	55.8	63.5	45.4	10.3	48.0	9.6	58.3	74.5	78.8	80.4	78.7	83.2
N24News	34.7	38.6	13.9	36.0	33.7	23.4	42.5	80.3	66.6	82.3	81.0	78.8
HatefulMemes	51.1	51.7	47.2	49.6	49.0	49.7	56.4	67.9	65.3	79.0	67.8	89.8
VOC2007	50.7	52.4	64.3	52.1	51.6	49.9	66.2	91.5	92.0	90.8	89.5	91.6
SUN397	43.4	68.8	39.6	34.5	57.0	33.1	63.2	75.8	78.7	80.3	78.7	83.8
Place365	28.5	37.8	20.0	21.5	31.5	8.6	36.5	44.0	42.9	46.8	45.4	49.6
ImageNet-A	25.5	14.2	42.6	3.2	8.0	2.0	9.8	43.6	48.0	53.9	48.4	59.1
ImageNet-R	75.6	83.0	75.0	39.7	70.9	30.8	66.2	79.8	89.3	90.1	83.5	90.1
ObjectNet	43.4	51.4	40.3	20.6	31.6	7.5	32.2	39.6	73.1	42.3	59.4	69.8
Country-211	19.2	16.8	14.2	2.5	6.2	3.1	11.3	14.7	19.8	25.0	25.4	30.4
All Classification	42.8	47.8	40.3	27.0	38.8	21.8	44.3	61.2	65.3	67.1	65.8	72.6
VQA (10 tasks)												
OK-VQA	7.5	11.5	2.4	8.7	12.7	8.9	25.4	69.0	71.9	71.7	72.6	75.4
A-OKVQA	3.8	3.3	1.5	3.2	2.9	5.9	8.8	54.4	71.4	58.7	63.6	72.2
DocVQA	4.0	5.3	4.2	2.6	3.0	1.7	6.2	52.0	92.6	93.8	95.8	95.4
InfographicsVQA	4.6	4.6	2.7	2.0	5.9	2.3	4.6	30.7	63.5	79.2	82.9	73.4
ChartQA	1.4	1.5	3.0	0.5	0.9	2.4	1.6	34.8	55.8	75.1	75.5	68.8
Visual7W	4.0	2.6	1.2	1.3	2.5	5.8	14.5	49.8	62.5	55.2	66.9	64.4
ScienceQA	9.4	10.2	7.9	6.8	5.2	3.6	12.8	42.1	54.0	53.7	59.5	65.3
VizWiz	8.2	6.6	2.3	4.0	1.7	2.6	24.3	43.0	53.7	51.6	53.5	54.9
GQA	41.3	52.5	57.5	9.7	43.5	7.8	48.8	61.2	69.5	69.3	77.9	70.0
TextVQA	7.0	10.9	1.0	3.3	4.6	8.2	15.1	62.0	84.5	83.5	87.6	86.3
All VQA	9.1	10.9	8.4	4.2	8.3	4.9	16.2	49.9	67.6	69.2	73.6	72.6
Retrieval (12 tasks)												
VisDial	30.7	25.4	21.5	18.0	24.8	9.2	42.2	80.9	84.2	80.7	85.8	83.7
CIRR	12.6	15.4	15.1	9.8	39.1	6.1	51.3	49.9	65.5	55.3	59.8	60.4
VisualNews_t2i	78.9	74.0	51.0	48.1	50.7	13.5	74.3	75.4	77.3	76.8	79.2	79.3
VisualNews_i2t	79.6	78.0	52.4	13.5	21.1	8.1	76.8	80.0	79.2	82.0	83.1	83.2
MSCOCO_t2i	59.5	63.6	58.3	53.7	54.1	20.7	68.5	75.7	79.1	78.3	81.2	81.6
MSCOCO_i2t	57.7	62.1	55.0	20.3	40.0	14.0	72.1	73.1	75.2	71.4	76.0	79.4
NIGHTS	60.4	66.1	62.9	56.5	58.1	4.2	66.2	65.5	68.1	68.1	69.5	71.2
WebQA	67.5	62.1	58.1	55.4	43.0	17.7	89.6	87.6	90.6	90.9	90.1	91.5
FashionIQ	11.4	13.8	20.1	9.3	11.2	2.8	40.2	16.2	26.4	23.4	29.6	23.7
Wiki-SS-NQ	55.0	44.6	55.1	28.7	18.7	8.6	12.2	60.2	71.2	72.5	69.5	67.7
OVEN	41.1	45.0	56.0	39.5	1.6	5.9	69.4	56.5	68.0	71.4	73.2	77.2
EDIS	81.0	77.5	23.6	54.4	62.6	26.8	79.2	87.8	88.2	92.0	92.5	86.4
All Retrieval	53.0	52.3	31.6	33.9	35.4	11.5	61.8	67.4	72.9	71.9	74.1	73.8
Visual Grounding (4 tasks)												
MSCOCO	33.8	34.5	46.4	28.9	22.1	10.8	46.6	80.6	78.2	72.7	84.2	83.6
RefCOCO	56.9	54.2	70.8	47.4	22.8	11.9	67.8	88.7	94.6	91.4	95.8	92.9
RefCOCO-matching	61.3	68.3	50.8	59.5	35.6	38.9	62.9	84.0	91.4	91.1	94.0	94.0
Visual7W-pointing	55.1	56.3	70.1	52.0	23.4	14.3	71.3	90.9	93.8	84.2	96.0	87.7
All Visual Grounding	51.8	53.3	59.5	47.0	26.0	19.0	65.3	86.1	90.2	84.9	92.5	89.6
Final Score (36 tasks)												
All	37.8	39.7	34.8	25.2	27.8	13.3	44.7	62.9	71.2	71.3	73.7	74.9
All IND	37.1	39.3	32.3	25.3	31.0	14.9	47.1	67.5	74.8	76.1	78.1	79.5
All OOD	38.7	40.2	38.0	25.1	23.7	11.5	41.7	57.1	66.7	65.1	68.2	69.1

C Additional Experiments on the Impact of Hyperparameters

Additionally, we analyze the impact of key hyperparameters in AGFF-Embed, namely the number of fine-grained embedding modules N, the number of prompt tokens M, and the amplification magnitude α. The main experiments in Sections 4.1 and 4.2 adopt the default settings of $M = N = 10$ and $\alpha = 20$. Subsequently, we adjust $M = N = \{3, 5\}$ and $\alpha = \{5, 10\}$ separately. The evaluation results on MMEB are presented in Table 6, and those on MMVP-VLM are shown in Table 7. Whether in general or fine-grained scenarios, AGFF-Embed equipped with fine-grained embedding modules of varying scales M, N achieves performance improvements of varying degrees compared to the pre-finetuned QQMM, which demonstrates the strong robustness of the perceptual fusion mechanism. Furthermore, due to the relatively low amplification magnitude, AGFF-Embed with $\alpha = 5$ shows no gain compared to the version without EGA. However, when α is increased to 10, the performance improvement becomes evident.

Table 6: Hyperparameter impact analysis on the MMEB benchmark.

Model	Per Meta-Task Score				Average Score		
	Classification	VQA	Retrieval	Grounding	IND	OOD	Overall
# of datasets	10	10	12	4	20	16	36
$M = N = 10$, $\alpha = 20$ (default)	**72.6**	**72.6**	73.8	**90.0**	**79.5**	**69.1**	**74.9**
M, N Modification							
$M = N = 5$, $\alpha = 20$	72.1	71.7	72.8	88.9	78.6	68.4	74.1
$M = N = 3$, $\alpha = 20$	71.8	71.7	72.9	89.2	78.3	68.8	74.1
$M = N = 0$, $\alpha = 20$ (QQMM)	69.9	70.0	72.1	86.0	77.2	66.6	72.5
α Modification							
$M = N = 10$, $\alpha = 10$	72.1	72.3	**73.9**	89.5	79.1	**69.1**	74.7
$M = N = 10$, $\alpha = 5$	72.3	72.4	73.5	89.1	79.1	68.9	74.6
$M = N = 10$, $\alpha = 0$ (w/o EGA)	72.5	72.2	73.4	89.3	79.0	**69.1**	74.6

Table 7: Hyperparameter impact analysis on the MMVP-VLM benchmark.

Model	Ori.	Pre.	Sta.	Qua.	Spa.	Col.	Str.	Tex.	Cam.	Average
$M = N = 10$, $\alpha = 20$ (default)	60.0	43.3	**73.3**	**73.3**	46.7	83.3	**66.7**	**50.0**	56.7	**61.5**
M, N Modification										
$M = N = 5$, $\alpha = 20$	56.7	**50.0**	66.7	63.3	50.0	80.0	63.3	43.3	60.0	59.3
$M = N = 3$, $\alpha = 20$	56.7	43.3	70.0	70.0	50.0	76.7	60.0	43.3	56.7	58.5
$M = N = 0$, $\alpha = 20$ (QQMM)	53.3	43.3	**73.3**	46.7	36.7	83.3	50.0	43.3	56.7	54.1
α Modification										
$M = N = 10$, $\alpha = 10$	56.7	46.7	70.0	70.0	46.7	**86.7**	63.3	**50.0**	**63.3**	**61.5**
$M = N = 10$, $\alpha = 5$	56.7	43.3	66.7	66.7	50.0	83.3	63.3	46.7	**63.3**	60.0
$M = N = 10$, $\alpha = 0$ (w/o EGA)	**63.3**	**50.0**	70.0	63.3	**53.3**	80.0	63.3	36.7	60.0	60.0

D TIME COST ANALYSIS

Table 8: Training and inference time on the MMEB benchmark.

Model	Training Time per Sample (s)	Inference Time per Sample (s)
QQMM ($M = N = 0$)	2.41	0.658
AGFF-EMBED ($M = N = 3$)	2.58	0.693
AGFF-EMBED ($M = N = 5$)	2.70	0.716
AGFF-EMBED ($M = N = 10$)	3.08	0.783

During training and evaluation on MMEB, we record the total time cost T_{total} and calculate the per-sample training and inference time:

$$T_{\text{sample}} = \frac{T_{\text{total}}}{N \cdot B}, \tag{34}$$

where N is the number of batches, and B is the batch size. In Table 8, we report the per-sample time cost of AGFF-EMBED with different numbers of fine-grained embedding modules N and different numbers of prompt tokens M, as well as that of the original QQMM. We observe that although increasing M, N introduces additional time overhead, it is not substantial. Even in the case of $M = N = 10$, it only increases the training time by approximately 28% and the inference time by about 19% compared to the original QQMM. In fact, for AGFF-EMBED, the additional cost introduced by the fine-grained embedding modules and similarity aggregation is marginal compared to image processing.