INSTRUCTION FOLLOWING IN MULTIMODAL EMBEDDINGS: MEASURABLE, DIAGNOSABLE, CONTROLLABLE

Anonymous authors
Paper under double-blind review

ABSTRACT

Current multimodal embedding models map images and text into a unified vector space and achieve strong performance on general image-text retrieval and visual question answering (VQA). However, in practical applications such as e-commerce, agent-driven Q&A, and medical diagnosis, users perform cross-modal retrieval with diverse perspectives and intentions, requiring multimodal embedding models to dynamically adjust their representations according to natural language instructions. Existing benchmarks primarily focus on behavioral-level performance evaluation and fail to quantitatively measure the instruction-following capability of multimodal embedding models. To address this limitation, we propose **IFME-Bench** (Instruction-Following for Multimodal Embeddings Benchmark) and the **4STEP** (4-Sequential Testing for Embedding Progression) evaluation pipeline. As the core component of 4STEP, IFME-Bench integrates eight measurable metrics and pioneers end-to-end instruction-following diagnosis from the behavioral layer to the embedding layer under multimodal settings. Crucially, our analysis identifies a verifiable mapping between IFME-Bench diagnostic metrics and training hyperparameters, demonstrating the feasibility of controlling the model's specific instruction-following sub-capabilities via hyperparameter optimization during training. To facilitate future research, we release two baseline models trained on Qwen3-VL-Embedding-2B: **GF-Embedding** (General instruction-following embedding), which achieves a favorable balance between general embedding quality and instruction-following performance, and **SF-Embedding** (Specialized instruction-following embedding), which outperforms all evaluated models in terms of nearly all metrics of IFME-Bench.

1 INTRODUCTION

Multimodal embedding models map images and text into a unified vector space, providing foundational representations for image-text retrieval, visual question answering (VQA), and image classification. VLM2Vec (Jiang et al., 2025) further introduced instructions into the embedding process, enabling a single image to produce different embeddings conditioned on natural language instructions. Leveraging this ability to dynamically shift embedding focus based on instructions, such models are increasingly deployed in e-commerce, agent-based Q&A, and medical diagnosis (Wang et al., 2022). As an example in medical imaging diagnosis, Figure 1 shows how the same CT scan maps to different text directions in an instruction-following setting. Given an upper-chest CT scan with multiple lesions, a doctor focusing on the left upper lobe issues the instruction "Aid the diagnosis of left lung lesion". The model should steer the image embedding toward the corresponding disease description for that region, distinguishing it from conditions in the right lung or trachea. When the model's instruction-following capability is insufficient, the embedding may falsely drift toward a description of the right lung lesion instead.

Despite its critical importance, evaluating whether these embedding models truly follow instructions remains an underexplored challenge. On the text retrieval side, FollowIR (Weller et al., 2025) evaluates whether models can execute multi-constraint retrieval instructions. On the multimodal side, CtrlBench (Schneider et al., 2025) verifies whether a model can use instructions to select the correct answer among plausible candidates. Yet both only reveal *whether* the retrieval result is correct,

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.

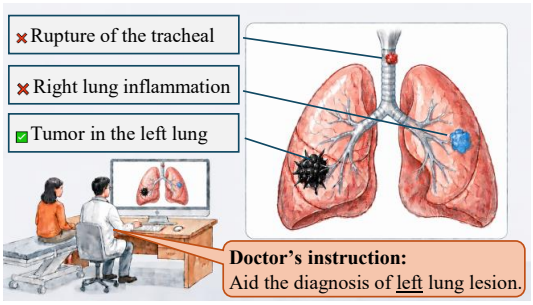


Figure 1: An example of instruction following, where different instructions are matched to lesions in different regions of the same lung CT image.

they treat the instruction-following chain as a black box, unable to answer the question of where instruction-following capability comes from.

To address the lack of interpretable diagnosis in existing benchmarks for multimodal embeddings, we propose IFME-Bench (Instruction-Following for Multimodal Embeddings Benchmark), a novel benchmark designed to measure and interpret instruction-following behaviors, serving as the core of our newly constructed 4STEP (4-Sequential Testing for Embedding Progression) pipeline. 4STEP progressively analyzes instruction-following quality from shallow to deep: Step 1 verifies base capability via MMEB-V2-Sub (Meng et al., 2026), which consists of the VQA and classification tasks from MMEB-V2 and is used to evaluate the basic alignment capability of embedding models, while Steps 2–4, powered exclusively by IFME-Bench, assess instruction-following performance, instruction sensitivity, and embedding-level representation shifts. Additionally, we train two baseline models and compare them against existing models, revealing that current multimodal embedding models, while maintaining general-purpose embedding quality, have room for improvement in instruction following; existing instruction-following models, conversely, often sacrifice general capability for limited instruction-following gains with poor generalization. To address these limitations, our core thesis is that instruction-following capability is ***measurable***, ***diagnosable***, and ***controllable***, and that evaluation results can directly guide training improvements. Specifically, we make three contributions:

- **4STEP Evaluation Pipeline and IFME-Bench Benchmark** : We propose the first full-chain diagnostic framework for instruction following in multimodal embeddings, which leverages eight metrics of IFME-Bench to translate black-box evaluation scores into localizable, fine-grained capability metrics, making instruction-following capability ***measurable*** and ***diagnosable***.
- **Two Baseline Models: GF-Embedding and SF-Embedding**: We construct two types of baselines targeting different trade-offs: GF-Embedding (General instruction-following embedding) balances general-purpose embeddings with instruction following, while SF-Embedding (Specialized instruction-following embedding) specializes in instruction following, providing the community with reference points across the performance spectrum.
- **Benchmark-Guided Optimization**: We demonstrate that IFME-Bench diagnostic metrics can be directly mapped to training hyperparameters, transforming the benchmark from a passive scoring tool into an active training guide that directs developers to resolve issue, making specific sub-capabilities ***controllable***.

2 Related Work

2.1 Multimodal Embedding Models

CLIP (Radford et al., 2021) pioneered contrastive visual-language pretraining on large-scale image-text pairs, and BLIP-2 (Li et al., 2023) added a language model for stronger cross-modal grounding. SigLIP (Zhai et al., 2023) replaced the contrastive objective with a sigmoid loss for better large-batch alignment. UniIR (Wei et al., 2024) unified eight retrieval tasks under instruction-guided multimodal

retrievers, VLM2Vec (Jiang et al., 2025) adapted vision-language models as embedding learners and proposed the MMEB benchmark to unify evaluation across retrieval, VQA, and classification; its successor VLM2Vec-V2 (Meng et al., 2026) extended MMEB to video and visual documents. These works target *general alignment* across cross-modal matching tasks but do not evaluate instruction-driven regulation of embedding representations.

2.2 Instruction-Following Embedding Evaluation

Instruction following was first systematized in generative LLMs, where instruction tuning (Wei et al., 2022; Ouyang et al., 2022; Wang et al., 2023) and constraint-following benchmarks (Jiang et al., 2024b) established the prevailing paradigm. InBedder (Peng et al., 2024) evaluated text embedding sensitivity to instruction variants in clustering, and FollowIR (Weller et al., 2025) constructed a benchmark and training set for multi-constraint retrieval instructions. In the multimodal domain, GeneCIS (Vaze et al., 2023) proposed a conditional image similarity benchmark, Ctrl-Bench (Schneider et al., 2025) tested whether instructions select the correct answer among plausible candidates, and MagicLens (Zhang et al., 2024) studied retrieval under open-ended instructions. MIA-Bench (Qian et al., 2025) evaluated instruction following in multimodal LLMs, and VIRTUE (Wang et al., 2026) introduced visually interactive prompts. More recently, MMEB-V3 (Huang et al., 2026) introduced a 190-task full-modality benchmark with OmniSET to disentangle semantic alignment from modality compliance. On the training side, InstEmb (Gao et al., 2026) improves instruction-following embeddings via look-ahead token distillation. A critical gap remains: text-side evaluations remain unimodal, and multimodal works report task performance without diagnosing the embedding level. This paper advances evaluation from the behavioral to the embedding layer through 4STEP and IFME-Bench.

3 4STEP and IFME-Bench

3.1 Overall Framework

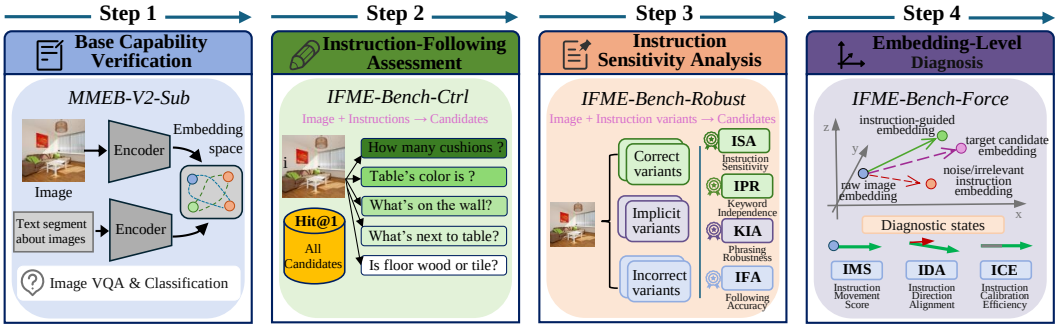


Figure 2: The 4STEP four-level progressive architecture: base capability verification (MMEB-V2-Sub) → instruction-following performance (IFME-Bench-Ctrl) → instruction sensitivity analysis (IFME-Bench-Robust) → embedding-level diagnosis (IFME-Bench-Force).

As shown in Figure 2, the 4STEP evaluation framework decomposes the evaluation of instruction-following capability in multimodal embedding models into four progressive steps. Step 1 verifies the base cross-modal alignment capability via image VQA and classification tasks from MMEB-V2 (Meng et al., 2026) (i.e., MMEB-V2-Sub), confirming whether the model accurately aligns image and text representations. Building upon this base, Steps 2 through 4 utilize our proposed IFME-Bench for in-depth diagnosis. Step 2 evaluates instruction-following performance via IFME-Bench-Ctrl, determining whether instructions effectively guide the model's retrieval. Step 3 examines instruction sensitivity via IFME-Bench-Robust, testing whether the semantic matching stems from deep semantic understanding rather than superficial keyword coincidence. Finally, Step 4 performs quantitative diagnosis using IFME-Bench-Force by analyzing instruction-induced representation shifts, investigating whether the instruction effectively aligns the embeddings toward the target representation with a sufficient shift magnitude. The specific diagnostic metrics employed in each step are summarized in Table 1.

Table 1: IFME-Bench diagnostic metric system.

Metric	Capability interpretation	Metric	Capability interpretation
Hit@1	Overall instruction-following effectiveness	ISA	Semantic sensitivity to correct instructions
KIA	Generalization without keyword leakage	IFA	Instruction- vs. image-driven retrieval
IPR	Robustness to instruction paraphrasing	IMS	Movement beyond the noise level
IDA	Shift alignment toward target direction	ICE	Efficiency of positive-sample pulling

3.2 IFME-Bench Dataset Construction

IFME-Bench consists of three specialized subsets: (1) **IFME-Bench-Ctrl** for instruction-following assessment (Step 2), (2) **IFME-Bench-Robust** for instruction sensitivity analysis (Step 3), and (3) **IFME-Bench-Force** for embedding-level diagnosis (Step 4). All are built through a unified two-stage process.

Base data production. To construct the base data, we sample 1,000 rich-object images from ADE20K (Zhou et al., 2017). For each image, we select five annotated objects from the source image and use GPT-5.6-Sol to generate five distinct (instruction, answer) pairs targeting different objects. Since all five candidate answers describe objects actually present in the image, retrieval cannot succeed by image-text matching alone and requires following the instruction, yielding approximately 5,000 evaluation triples (image, instruction, answer). We denote the set of all instructions as $\mathcal{V} = \{v_i\}$.

Instruction variant expansion. To support both instruction sensitivity analysis (Step 3) and embedding-level representation diagnosis (Step 4), we expand the base instructions $\mathcal{V}$ by constructing three types of instruction variants derived from each original instruction using GPT-5.6-Sol. (1) **Correct variants** $\mathcal{V}_{\textbf{corr}}$: We generate four paraphrases of the original instruction that alter the sentence structure (e.g., converting a wh-question into a yes/no question or an imperative) while preserving the original meaning. (2) **Implicit variants** $\mathcal{V}_{\textbf{impl}}$: Based on the original instruction, we generate four ambiguous rewrites that semantically target the same object but deliberately omit explicit object names. This prevents keyword leakage. (3) **Incorrect variants** $\mathcal{V}_{\textbf{inco}}$: We directly reuse the instructions of the other four objects associated with the same image.

Data instruction verification. To ensure reliability, we randomly sampled 50% of the entries for manual inspection against our evaluation criteria, obtaining a validation accuracy of 98.4%.

3.3 Step 1: Base Capability Verification

Robust instruction-following presupposes basic multimodal alignment. Therefore, Step 1 establishes this necessary baseline by evaluating embedding capabilities on MMEB-V2-Sub. These two task families directly probe whether image representations are truly grounded in textual semantics, and thus serve as an indicator of the model's base embedding alignment capability. The model encodes an image I and its instruction v into a query vector $\mathbf{q}_v = \text{Emb}(I, v)$ and candidate answers as $\mathcal{C} = \{\mathbf{c}_i\}$, predicting the top-ranked candidate to finish the VQA and classification via cosine similarity $\text{sim}(\mathbf{q}_v, \mathbf{c}_i) = \frac{\mathbf{q}_v^\top \mathbf{c}_i}{\|\mathbf{q}_v\|\|\mathbf{c}_i\|}$.

3.4 Step 2: Instruction-Following Assessment

Building on the image-text alignment verified in Step 1, Step 2 employs IFME-Bench-Ctrl, which follows a "one image, five instructions, five candidates" scheme similar to CtrlBench. Specifically, each image is paired with five distinct instructions, and the five candidate answers of the same image form its per-image pool. Since all candidates are semantically reasonable for the same image, selecting the correct one requires precise instruction following. The model jointly embeds the image and instruction as query vector $\mathbf{q}_v$, computes cosine similarity with the candidate answers, and selects the highest match as the prediction. Hit@1 is computed over the global pool of all 5,000

answers, enlarging the candidate range to raise the difficulty, whereas the variant-based metrics ISA, KIA, and IFA in Step 3 use the per-image pool, so that instruction sensitivity is measured more rigorously. We compute Hit@1 to assess the model's instruction-following performance. Under uniform random prediction over the global pool, the expected Hit@1 is 0.02%.

3.5 Step 3: Instruction Sensitivity Analysis

In Step 3, we adopt IFME-Bench-Robust to quantify the model's instruction sensitivity from the dimensions of generalization and robustness. During evaluation, original instructions are replaced by variants, while the five candidate answers of each image constitute a fixed retrieval pool and the matching target remains the original answer. For the Implicit and Correct variants, each variant type yields 20,000 test instances (1,000 images × 5 candidate answers × 4 instructions). Based on the retrieval performance across these extensive test instances, IFME-Bench-Robust defines four metrics for fine-grained diagnosis of the instruction sensitivity:

- **ISA (Instruction Sensitivity Accuracy)**: As illustrated in Eq. 1, ISA computes the global average Hit@1 over the full set of Correct instruction variants. Since paraphrased instructions share the same retrieval intent, ISA evaluates the model's fundamental understanding of the base instruction, independent of phrasing fluctuations. Under uniform random prediction over the five candidates, the expected ISA is 20%.
- **KIA (Keyword Independence Accuracy)**: As defined in Eq. 2, KIA computes the global average Hit@1 over all implicit instruction variants. By evaluating instructions that contain no explicit object names, KIA forces the model to infer the target purely from semantic context. This metric directly measures the model's independent retrieval capability without relying on surface-level keyword matching.
- **IPR (Instruction Phrasing Robustness)**: According to Eq. 3, IPR is computed as one minus the average of the per-sample standard deviations. Specifically, for k-th sample, we calculate the standard deviation of similarities between the query vector (encoded with each correct instruction variant) $\mathbf{q}_v$ and the target answer embedding. Averaging these standard deviations across all N samples and subtracting the result from one yields IPR. This metric measures robustness to instruction paraphrasing; a higher IPR indicates that the model's retrieval similarity remains stable regardless of how the instruction is phrased.
- **IFA (Instruction Following Accuracy)**: As shown in Eq. 4, IFA is computed as one minus the global average Hit@1 over the full set of Incorrect variants. If an embedding model ignores textual instructions, it will erroneously retrieve the target answer despite the mismatched instruction. Under uniform random prediction over the five candidates, the expected IFA is 80%; scores substantially above this baseline indicate an instruction-driven departure rather than random deviation.

The formulas for these four metrics are listed below:

$$\text{ISA} = \frac{1}{|\mathcal{V}_{\text{corr}}|} \sum_{v \in \mathcal{V}_{\text{corr}}} \mathbb{I}\big[\arg\max_{\mathbf{c} \in \mathcal{C}_k} \text{sim}(\mathbf{q}_v, \mathbf{c}) = \mathbf{c}^*\big] \tag{1}$$

$$\text{KIA} = \frac{1}{|\mathcal{V}_{\text{impl}}|} \sum_{v \in \mathcal{V}_{\text{impl}}} \mathbb{I}\big[\arg\max_{\mathbf{c} \in \mathcal{C}_k} \text{sim}(\mathbf{q}_v, \mathbf{c}) = \mathbf{c}^*\big] \tag{2}$$

$$\text{IPR} = 1 - \frac{1}{N} \sum_{k=1}^{N} \text{std}\left(\{\text{sim}(\mathbf{q}_v, \mathbf{c}^*) \mid v \in \mathcal{V}_{\text{corr}}^{(k)}\}\right) \tag{3}$$

$$\text{IFA} = 1 - \frac{1}{|\mathcal{V}_{\text{inco}}|} \sum_{v \in \mathcal{V}_{\text{inco}}} \mathbb{I}\big[\arg\max_{\mathbf{c} \in \mathcal{C}_k} \text{sim}(\mathbf{q}_v, \mathbf{c}) = \mathbf{c}^*\big] \tag{4}$$

where $\mathcal{V}_{\text{corr}}$, $\mathcal{V}_{\text{impl}}$, and $\mathcal{V}_{\text{inco}}$ denote the full sets of Correct, Implicit, and Incorrect instruction variants, respectively; $\mathcal{C}$ is the global candidate pool of all $5{,}000$ answer embeddings over which Hit@1 is computed, $\mathcal{C}_k$ denotes the five-candidate pool of the k-th image over which ISA, KIA, and IFA are computed, $\mathbf{q}_v$ is the query vector encoded with instruction v, N is the total number of test images, and $\mathcal{V}_{\text{corr}}^{(k)}$ is the set of Correct instruction variants for the k-th image, and $\mathbf{c}^*$ is the target answer embedding.

3.6 STEP 4: EMBEDDING-LEVEL DIAGNOSIS

In Step 4, we introduce IFME-Bench-Force to evaluate the model's instruction-induced representation shift in the embedding space through three progressive stages: (1) whether the instruction produces an effective shift beyond the noise level; (2) whether the shift direction correctly aligns with the target answer; and (3) whether the instruction successfully pulls the image embedding closer to the target answer. These three stages constitute a comprehensive instruction-diagnosis chain, systematically decomposing the internal mechanism of instruction following within the embedding space. Accordingly, IFME-Bench-Force defines the three Embedding-Level diagnostic metrics:

- **IMS (Instruction Movement Score)**: As shown in Eq. 5, IMS is the average difference between the target similarity and the noise baseline. The target similarity is the cosine similarity between the instruction-conditioned query $\mathbf{q}_v$ and the target answer $\mathbf{c}^*$; the noise baseline averages $\text{sim}(\mathbf{q}_{vn}, \mathbf{c}^*)$ over noise instructions. Consequently, an IMS significantly greater than 0 confirms that the instruction substantively shifts embedding beyond generic descriptions, whereas IMS approaching 0 indicates a loss of instruction-following capability. The baseline is zero by construction, since the noise instruction carries no directional information.
- **IDA (Instruction Direction Alignment)**: IDA precisely measures how well the instruction aligns the image embedding with the target answer direction in the embedding space. As illustrated in Eq. 6, we define IDA as the average of the cosine similarity between the instruction shift direction $\frac{\mathbf{q}_v - \mathbf{e}_I}{\|\mathbf{q}_v - \mathbf{e}_I\|}$ and the target direction $\frac{\mathbf{c}^* - \mathbf{e}_I}{\|\mathbf{c}^* - \mathbf{e}_I\|}$ across all instructions in $\mathcal{V}$. Random directions in high dimension are near-orthogonal, so the chance level of IDA is approximately zero.
- **ICE (Instruction Calibration Efficiency)**: As shown in Eq. 7, ICE is the average ratio of the instruction-induced gain to the maximum gain achievable if the pictures were perfectly aligned with the target answer. ICE is zero when the instruction yields no gain over the image-only embedding, and negative when it actively drives the picture away from the target answer.

Taken together, the eight metrics of IFME-Bench render instruction-following capability ***measurable*** and ***diagnosable*** across both behavioral and representational levels.

The formulas for these three metrics are listed below:

$$\text{IMS} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \left\{ \text{sim}(\mathbf{q}_v, \mathbf{c}^*) - \left[\frac{1}{M} \sum_{j=1}^{M} \text{sim}(\mathbf{q}_{vn}^{(j)}, \mathbf{c}^*) \right] \right\} \tag{5}$$

$$\text{IDA} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \text{sim} \left(\frac{\mathbf{q}_v - \mathbf{e}_I}{\|\mathbf{q}_v - \mathbf{e}_I\|}, \frac{\mathbf{c}^* - \mathbf{e}_I}{\|\mathbf{c}^* - \mathbf{e}_I\|} \right) \tag{6}$$

$$\text{ICE} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{\text{sim}(\mathbf{q}_v, \mathbf{c}^*) - \text{sim}(\mathbf{e}_I, \mathbf{c}^*)}{1 - \text{sim}(\mathbf{e}_I, \mathbf{c}^*)} \tag{7}$$

where $\mathbf{q}_v$ is the query vector encoded with instruction v, $\mathbf{q}_{vn}$ is the query encoded with a noise instruction, $\mathbf{e}_I$ is the image embedding without instruction conditioning, M is the number of noise instructions per target answer, and $\mathbf{c}^*$ is the target answer embedding.

4 BASELINES AND EXPERIMENTS

4.1 TRAINING OF BASELINES

We construct two baseline embedding models on a dedicated training dataset, implemented with the ms-swift (Zhao et al., 2025) framework.

Training data construction and long-tail handling. We construct (image, instruction, answer) triplets to train the model to align image embeddings with instruction-conditioned text embeddings.

We leverage the Visual Genome (VG) dataset (Krishna et al., 2017), whose fine-grained region descriptions and "one-image, multiple-objects" structure provide the semantics essential for instruction diversification. For each image, we select annotated objects and use their original region descriptions as target answers to minimize synthetic noise. Corresponding instructions are generated via Doubao-Seed-2.0-lite.

However, VG also suffers from a severe long-tail distribution. To prevent the model from overfitting to high-frequency background classes (e.g., "sky" or "ground"), we employ inverse-frequency sampling. Each object category c is assigned a sampling weight $w_c = \frac{1}{\log(f_c+1)}$ based on its global frequency f_c. This strategy effectively balances the distribution, reducing the Top-100 object concentration from 47.5% to 40.1% and suppressing ubiquitous background classes to below 2% of their original frequency. This ensures the training signal focuses on discriminative object semantics, providing a robust baseline for instruction-following evaluation.

Training Loss. We employ LoRA (Hu et al., 2022) to fine-tune Qwen3-VL-Embedding-2B using an InfoNCE contrastive loss (van den Oord et al., 2018), with explicit hard negatives from the other instructions of the same image, in addition to standard in-batch negatives, forcing the model to distinguish different instructions for the same image:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(s_{i,i}/\tau)}{\exp(s_{i,i}/\tau)+\sum_{j\neq i}\exp(s_{i,j}/\tau)} \tag{8}$$

where $s_{i,j} = \text{sim}(\text{Enc}(I_i, \text{instr}_i), \text{Enc}(\text{answer}_j))$ denotes the cosine similarity between the image-text joint embedding of sample i and the j-th answer embedding, τ is the temperature parameter, and B is the batch size.

4.2 Benchmark-Guided Optimization

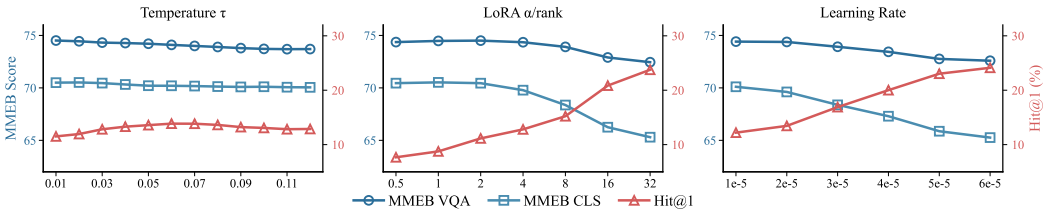


Figure 3: Trends of general embedding capability (MMEB_VQA, MMEB_CLS) and instruction-following ability (IFME-Bench-Ctrl Hit@1) under systematic hyperparameter sweeps.

Hyperparameter Choice. Our baseline ($\tau = 0.01$, LoRA rank $= 8$ with scaling factor $\alpha = 16$, batch size $= 128$, and learning rate $= 1 \times 10^{-5}$) follows the commonly adopted configuration for instruction fine-tuning of multimodal LLM-based embedding models (Jiang et al., 2024a; Liu et al., 2023). We then conduct a hyperparameter sweep over three core hyperparameters (τ, LoRA α/rank, learning rate) around this baseline. We stress that this is a baseline choice for benchmark study rather than a hyperparameter search.

Figure 3 reveals a *seesaw effect*: MMEB_VQA and MMEB_CLS move oppositely to IFME-Bench-Ctrl Hit@1 as hyperparameters vary. τ shows a shallow plateau on IFME-Bench-Ctrl Hit@1 with a mild peak near 0.07. Based on these observations, we select two representative hyperparameter combinations summarized in Table 2 to construct two baseline models:

- **GF-Embedding** (General instruction-following embedding) directly adopts the baseline configuration, minimizing disturbance to the pretrained base. It preserves high MMEB general embedding quality while achieving a modest instruction-following gain.
- **SF-Embedding** (Specialized instruction-following embedding) uses τ=0.07 together with a high α, trading a modest amount of general embedding quality for substantially stronger instruction following.

Both models are released solely as reproducible reference points for future users, rather than as a claim of a generally superior embedding backbone.

Table 2: Training hyperparameter configurations for the two baseline models.

Hyperparameter	GF-Embedding	SF-Embedding
Temperature τ	0.01	0.07
LoRA rank	8	8
LoRA scaling α	16	256
Batch size	128	128
Learning rate	1×10^{-5}	6×10^{-5}

Hyperparameter-Metric Analysis. Hyperparameter changes inevitably shift metric scores. To quantify the per-metric sensitivity that lets IFME-Bench directly guide hyperparameter optimization, we introduce the Relative Sensitivity Index (RSI), which normalizes a metric's relative gain by the average relative gain across all metrics within the same hyperparameter sweep:

$$RSI_{i,H} = \frac{\Delta_{i,H}}{\overline{\Delta_H}}, \qquad \Delta_{i,H} = \frac{M_{i,\text{end}} - M_{i,\text{start}}}{M_{i,\text{start}}} \tag{9}$$

where H indexes the swept hyperparameter, i an IFME-Bench metric, and $M_{i,\text{start}}$, $M_{i,\text{end}}$ its scores at the two sweep endpoints, with $\overline{\Delta_H}$ the mean relative gain across the eight IFME-Bench metrics in sweep H. We sweep $\tau \in [0.01, 0.12]$, and $[0.01, 0.07]$, α/rank $\in [0.5, 32]$, and learning rate $\in [1 \times 10^{-5}, 6 \times 10^{-5}]$. By construction, $\text{RSI}_{i,H}$ depends only on the metric values at the two endpoints of sweep H, not on the trajectory between them. Figure 4 illustrates this for τ: its two rows, computed over $[0.01, 0.12]$ and $[0.01, 0.07]$ respectively, identify the same two metrics, IMS and ICE, as those most strongly driven by τ.

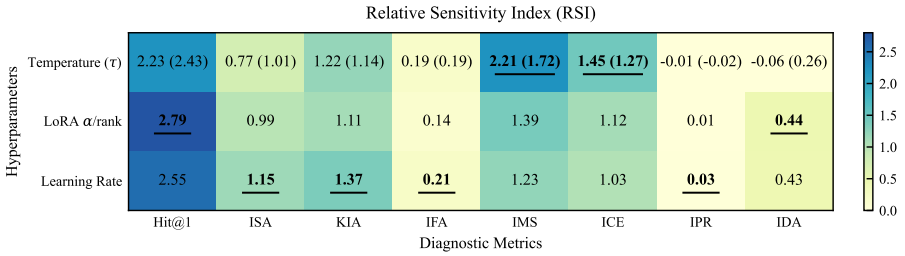


Figure 4: RSI of metrics across the τ, LoRA α/rank, and learning-rate sweeps. Values outside parentheses for τ are computed over the full sweep $[0.01, 0.12]$ and those inside over the monotone rising segment $[0.01, 0.07]$; the winning hyperparameter per metric is identical under either range.

Figure 4 reports the RSI matrix for eight core metrics. Each row reveals how efficiently a benchmark metric responds to a given hyperparameter. A higher RSI indicates greater efficiency; to strengthen a specific capability, prioritize the hyperparameter with the highest RSI in that column, enabling more efficient control of instruction-following ability via training hyperparameters. The RSI matrix yields two actionable insights. First, IMS and ICE respond most strongly to τ: a larger τ flattens the softmax distribution, leaving the explicit instruction-level hard negatives as the dominant training signal. Second, IPR exhibits a near-zero RSI across all sweeps, indicating that training barely changes what IPR measures. Through RSI, IFME-Bench closes the loop from diagnosis to targeted action, making instruction-following capability ***controllable***.

4.3 Comparison Results

We compare our baselines against SOTA multimodal embedding models: GME (Zhang et al., 2025), Ops-MM-embedding-v1 (OpenSearch-AI Team, 2025), Qwen3-VL-Embedding (Li et al., 2026), UME-R1 (Lan et al., 2026) and ABC (Schneider et al., 2025). Table 3 summarizes the results.

SF-Embedding leads on instruction-following dimensions, setting the best score on nearly all metrics. GF-Embedding marginally surpasses the 2B Qwen3-VL-Embedding on MMEB while achieving a substantial improvement in instruction following, suggesting that enhancing instruction following need not come at the cost of base embedding quality. Because GF-Embedding and SF-Embedding are both fine-tuned from Qwen3-VL-Embedding-2B, their IPR stays within a few points of the base model, echoing the near-zero IPR in Figure 4. This demonstrates that fine-tuning barely affects the robustness of embedding models, which is instead largely determined by the original weight distribution of the base model. Base models exceed the 20% chance level on ISA and KIA by only 13–19 points, indicating near-chance instruction sensitivity, whereas SF-Embedding reaches 92.25 and 85.16 on the same metrics; the two metrics thus separate models over a dynamic range of more than 70 points.

ABC also achieves strong instruction-following scores, but its MMEB_VQA is only 31.0, revealing a deficit in base embedding ability. This observation demonstrates the necessity for 4STEP to jointly evaluate general capability and instruction-following ability. UME-R1-2B illustrates the orthogonality of IPR and Hit@1, its IPR of 99.31 indicates that its retrieval similarity is stable across phrasings, while its Hit@1 of 0.68 indicates weak absolute alignment strength. The two metrics are complementary rather than contradictory. Collectively, these findings confirm that our benchmark exposes instruction-following deficits invisible to aggregate metrics, and that the GF/SF-Embedding baselines balance overall quality and instruction-following specialization.

Table 3: Comprehensive 4STEP evaluation across all models. all IFME-Bench metrics are reported as percentages, obtained by scaling the raw values by 100. best in each column is bold.

Model	Size	MMEB_VQA	MMEB_CLS	Hit@1	ISA	KIA	IPR	IFA	IMS	IDA	ICE
GME	2B	29.9	54.4	9.94	37.21	28.37	98.10	83.33	3.02	30.15	3.97
Ops-MM-embedding-v1	2B	65.1	68.1	12.68	36.90	29.11	98.85	84.28	6.42	29.16	7.99
UME-R1	2B	62.8	64.8	0.68	34.84	30.55	**99.31**	84.94	2.75	25.06	6.32
Qwen3-VL-Embedding	2B	74.3	70.3	5.00	38.59	33.02	95.29	83.97	8.08	40.08	12.49
ABC	7B	31.0	56.1	31.72	86.48	79.81	97.80	97.25	5.30	58.12	21.15
GME	7B	34.7	57.7	3.78	37.01	29.17	97.51	83.25	2.97	36.23	0.95
Ops-MM-embedding-v1	7B	69.6	69.7	11.64	48.96	41.05	98.94	87.48	2.91	32.16	4.45
UME-R1	7B	69.2	67.1	24.70	73.94	64.41	98.17	95.06	15.03	47.80	7.10
Qwen3-VL-Embedding	8B	**81.1**	**74.2**	10.56	43.97	33.37	96.29	85.05	15.43	37.61	3.83
GF-Embedding (ours)	2B	74.6	70.5	11.24	55.33	46.28	95.05	88.33	15.86	53.85	23.59
SF-Embedding (ours)	2B	71.9	64.7	**32.84**	**92.25**	**85.16**	96.56	**98.32**	**36.77**	**71.18**	**44.24**

5 CONCLUSION

This paper establishes three core goals for instruction following in multimodal embedding models: making it ***measurable***, ***diagnosable***, and ***controllable***. Leveraging eight metrics spanning the behavioral and representational levels, the 4STEP pipeline and IFME-Bench conduct detailed and comprehensive evaluations of instruction-following capabilities. They map the opaque instruction-following ability into decomposed fine-grained assessments, enabling the measurement and diagnosis of the underlying mechanisms of instruction following. RSI further translates these diagnostics into training hyperparameters, showing how efficiently a benchmark metric responds to a given hyperparameter, closing the loop from diagnosis to targeted optimization and allowing the instruction-following ability to be deliberately controlled during training. We also construct two baseline models, GF-Embedding and SF-Embedding, providing the community with reference benchmarks covering both generalist and specialist instruction-following capabilities.

Although this work builds a comprehensive framework for diagnosing instruction-following capabilities of multimodal embeddings, IFME-Bench mainly focuses on static natural images and single-step retrieval instructions. Future work will extend the 4STEP evaluation pipeline to modalities such as video and audio, and support more complex multi-turn reasoning instructions to evaluate model performance across a broader range of real-world scenarios.

AI USE STATEMENT

In this work, we used generative AI tools for synthetic data generation, method implementation, and translation. We have not used generative AI tools for theoretical models, mathematical claims or proof ingredients, hypotheses, methodology or experiment design, dataset cleaning, qualitative analysis, or interpreting results, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for code editing, figure generation, readability editing, and reference formatting.

We have reviewed all AI-assisted work. Specifically, GPT-5.6-Sol was used as the data generator for IFME-Bench: it produced (i) the five (instruction, answer) pairs per image in the base data and (ii) the Correct and Implicit instruction variants. Doubao-Seed-2.0-lite was used to generate the instruction–answer pairs in the Visual Genome training data. The generation protocol, the variant definitions, and the evaluation design were specified by the authors rather than proposed by the model. The resulting data were verified by the authors on a 50% sample with 98.4% accuracy, and the noise instructions used for the IMS baseline were written by the authors. LLM-assisted code was tested by the authors against the reported results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Tianhao Gao, Jun Fang, Xiaohui Zhang, Zhiyuan Liu, Chao Liu, Pengzhang Liu, and Qixia Jiang. InstEmb: Instruction-following embeddings through glimpses of the future. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022.

Haohang Huang, Xuan Lu, Mingyi Su, Xuan Zhang, Ziyan Jiang, Ping Nie, Kai Zou, Tomas Pfister, Wenhu Chen, Wei Zhang, Xiaoyu Shen, and Rui Meng. MMEB-V3: Measuring the performance gaps of omni-modality embedding models. *arXiv preprint arXiv:2604.23321*, 2026.

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*, 2024a.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. FollowBench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4667–4688. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.acl-long.257.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training vision-language models for massive multimodal embedding tasks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. doi: 10.1007/s11263-016-0981-7.

Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. UME-R1: Exploring reasoning-driven generative multimodal embeddings. In *The Fourteenth International Conference on Learning Representations*, 2026.

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang 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.

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

Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Raghuveer Thirukovalluru, Xuan Zhang, Zeyuan Chen, Ran Xu, Caiming Xiong, Yingbo Zhou, Wenhu Chen, and Semih Yavuz. VLM2Vec-V2: Advancing multimodal embedding for videos, images, and visual documents. *Transactions on Machine Learning Research*, 2026.

OpenSearch-AI Team. Ops-MM-Embedding-v1. `https://huggingface.co/OpenSearch-AI/Ops-MM-embedding-v1-2B`, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Letian Peng, Yuwei Zhang, Zilong Wang, Jayanth Srinivasa, Gaowen Liu, Zihan Wang, and Jingbo Shang. Answer is all you need: Instruction-following text embedding via answering the question. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 459–477. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.27.

Yusu Qian, Hanrong Ye, Jean-Philippe Fauconnier, Peter Grasch, Yinfei Yang, and Zhe Gan. MIA-Bench: Towards better instruction following evaluation of multimodal LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Benjamin Schneider, Florian Kerschbaum, and Wenhu Chen. ABC: Achieving better control of visual embeddings using VLLMs. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Sagar Vaze, Nicolas Carion, and Ishan Misra. GeneCIS: A benchmark for general conditional image similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6862–6872, 2023.

Wei-Yao Wang, Kazuya Tateishi, Qiyu Wu, Shusuke Takahashi, and Yuki Mitsufuji. VIRTUE: Visual-interactive text-image universal embedder. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.754.

Zifeng Wang, Zhenbang Wu, Dinesh Agarwal, and Jimeng Sun. MedCLIP: Contrastive learning from unpaired medical images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3876–3887. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.256.

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 *Proceedings of the European Conference on Computer Vision*, volume 15145 of *Lecture Notes in Computer Science*, pp. 387–404. Springer, 2024. doi: 10.1007/978-3-031-73021-4_23.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations*, 2022.

Orion Weller, Benjamin Chang, Sean MacAvaney, Kyle Lo, Arman Cohan, Benjamin Van Durme, Dawn Lawrie, and Luca Soldaini. FollowIR: Evaluating and teaching information retrieval models to follow instructions. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11926–11942. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.597.

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. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 59403–59420. PMLR, 2024.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Bridging modalities: Improving universal multimodal retrieval by multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9274–9285, 2025.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. SWIFT: A scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 29733–29735, 2025. doi: 10.1609/aaai.v39i28.35383.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 633–641, 2017.

APPENDIX

A NOISE INSTRUCTIONS FOR IMS

IMS uses five noise instructions to establish a baseline. The five noise instructions are:

Table 4: Noise instructions used in IMS baseline computation.

Noise Instructions
"Identify the scene shown in the image."
"Represent the given image."
"Identify the object shown in the image."
"Find an image caption describing the given image."
"Represent the given image for classification."

B STEPWISE HYPERPARAMETER COMBINATION ABLATION

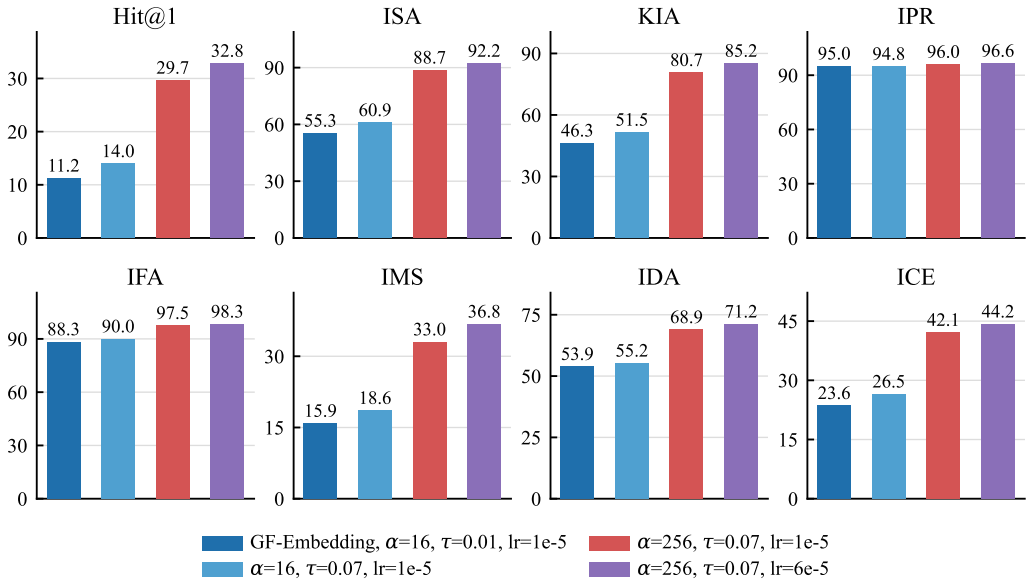


Figure 5: Stepwise ablation of the hyperparameter combination. each panel reports one IFME-Bench metric for the GF-Embedding baseline and three progressively modified configurations (α=16, τ=0.07, lr=1×10^{-5}; α=256, τ=0.07, lr=1×10^{-5}; α=256, τ=0.07, lr=6×10^{-5}).

We investigate whether the single-hyperparameter effects identified in the sweeps can be stacked. Starting from the GF-Embedding baseline, we first change τ to 0.07, then enlarge α (16→256), and finally raise the learning rate to 6×10^{-5} (Figure 5). Each step yields consistent gains across nearly all metrics, e.g., Hit@1 rises from 11.24 to 13.96 after setting τ, to 29.66 after enlarging α, and to 32.84 after increasing the learning rate, mirroring the respective sweeps. This confirms that favorable single-hyperparameter choices are additive under combination, and can be stacked to assemble an effective configuration.

C IFME-BENCH DATA VERIFICATION

Figure 6 illustrates the construction of a single IFME-Bench sample, from the source ADE20K image to its instruction–answer pairs and the derived instruction variants. Each image is annotated with five objects, each paired with one instruction and one candidate answer, and each base instruction is expanded into four Correct and four Implicit variants.

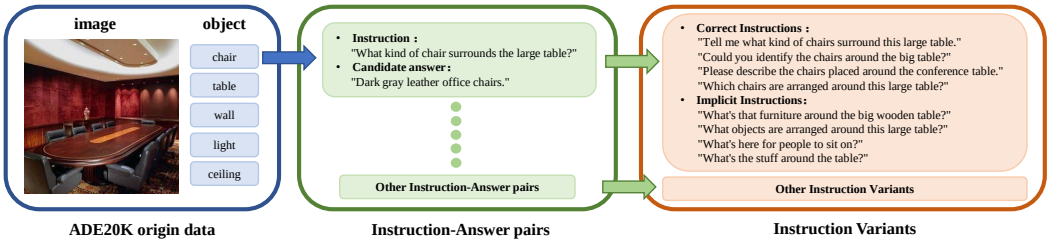


Figure 6: One IFME-Bench sample: from the ADE20K source image to its instruction–answer pairs and instruction variants.

We conducted manual inspection on 50% of the generated IFME-Bench data. For each image, we first check whether the five candidate answers in the base data refer to five distinct objects that truly exist in the image, and whether the current instruction can only be uniquely answered by its corresponding candidate answer. Next, we verify that both the Correct variants and the Implicit variants can be answered by the current candidate answer. Finally, we check whether the Implicit variants contain no vocabulary identical to the objects in the original dataset. Only when all conditions are satisfied is a sample deemed data-accurate. We built a human annotation interface to facilitate this review process, shown in Figure 7.

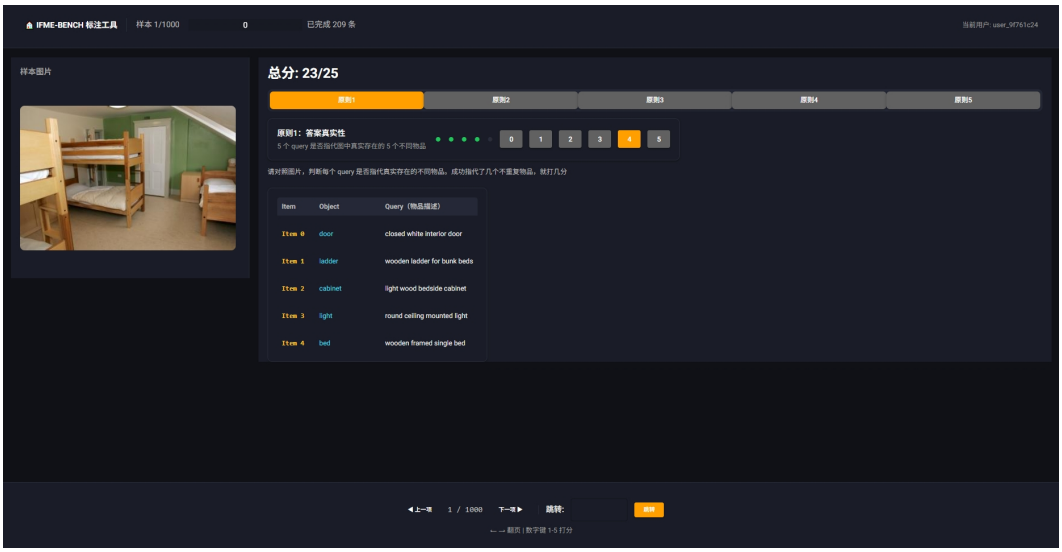


Figure 7: Human review interface for IFME-Bench data verification.

D DATA GENERATION PROMPTS

All instructions, candidate answers, and instruction variants for IFME-Bench were generated using GPT-5.6-Sol: the base data were generated from 1,000 images sampled from ADE20K, and the instruction variants were subsequently generated from the base data. The training dataset was constructed from Visual Genome region descriptions, with instructions generated by Doubao-Seed-

2.0-lite. The full prompts and output templates are documented below; in these prompts, "query" denotes the candidate answer.

D.1 IFME-BENCH BASE DATA GENERATION PROMPT

```
<Image> This image contains the following objects: {objects_str}

Your task is to generate exactly 5 short English descriptions
(queries) about objects in this image. Specifically: select 5
different objects from above. For each query, write an English
question (instruction) where that query is the answer.

- Each query must be 4-8 words, specific and grounded in the
  image content

- Each instruction must be 5-15 words, natural and answerable
  by the corresponding query

- All output must be in English

Output your answer as JSON in the following template exactly:
{"obj_1": <OBJECT NAME 1>, "query_1": <QUERY 1>,
 "instruction_1": <INSTRUCTION 1>,
 "obj_2": <OBJECT NAME 2>, "query_2": <QUERY 2>,
 "instruction_2": <INSTRUCTION 2>,
 "obj_3": <OBJECT NAME 3>, "query_3": <QUERY 3>,
 "instruction_3": <INSTRUCTION 3>,
 "obj_4": <OBJECT NAME 4>, "query_4": <QUERY 4>,
 "instruction_4": <INSTRUCTION 4>,
 "obj_5": <OBJECT NAME 5>, "query_5": <QUERY 5>,
 "instruction_5": <INSTRUCTION 5>}

Output ONLY the JSON, no other text.
```

D.2 IFME-BENCH INSTRUCTION VARIANTS GENERATION PROMPT

```
<Image> This image contains: {objects_str}

- Object name: {object_name}

- Target query: "{query}"

- Original instruction: "{original_instruction}"

In a retrieval task, given the image + an instruction, the
model should rank the target query highest among 5
candidates. The instruction is evaluated by whether
image+instruction embedding is most similar to the target
query embedding.

You are a real human user interacting with a smart
assistant. Your instructions must sound conversational,
casual, and natural. Avoid robotic phrasing, overly formal
language, or textbook-style descriptions.

Your task - generate instruction VARIANTS for robustness
testing:

1. Correct variants (4 items): Synonymous rewrites of the
   original instruction.
   - These should still point to the SAME object and same
     target query as the answer
   - Vary sentence structure: question -> imperative
     ("Please tell me..."), different question words (what ->
```

```
      which -> describe -> identify -> tell me about), use
      conversational fillers if appropriate (e.g., "could you
      tell me...", "I'm curious about..."). AVOID to begin
      with "what"
   - Keep the instruction natural, 5-15 words, in English
   - If the Object name of Target query exists in Original
     instruction, you can keep it in Correct variants
     generated by yourself

2. Implicit variants (4 items): Semantic equivalents that
   point to the SAME object, but WITHOUT using Object name.
   - MUST NOT include any descriptive words that appear in
     the Target query.
   - EXTREME VAGUENESS (CRITICAL): The instructions must be
     highly ambiguous and lack precise spatial or action
     details. Do NOT describe exact relationships. Instead,
     rely on:
     * Subjective/Visual Impressions
     * Rough Categorization
     * Vague Pointing/Deixis
     * General Function
   - STRICT NO-REPETITION RULE: Each variant must use a
     completely different level of vagueness and sentence
     structure. Do not repeat the same phrasing pattern.
   - Keep the instruction natural, 5-15 words, in English.

Output ONLY a JSON object (no other text):
{"correct_instructions": ["...", "...", "...", "..."],
 "implicit_instructions": ["...", "...", "...", "..."]}
```

D.3 Training Data Generation Prompt

The variable {pairs_text} is a pre-rendered string built from a list of (object_name, description) pairs, formatted as numbered lines:

```
<Image> This image contains several objects. For each
object below, generate a natural English question
(instruction) where the given description is the correct
answer.

{pairs_text}

- Each instruction MUST be a QUESTION starting with a
  Wh- question word: what, which, who, where, when, why,
  how (NOT can, could, would, please, describe, tell me)

- Avoid starting with auxiliary/modal verbs (can, could,
  would, do, does, is, are)

- 5-20 words, natural conversational English

- Each question must be answerable ONLY by its given
  Target query description, NOT by other descriptions in
  this list

- Questions should be DIVERSE in phrasing and question
  type -- don't repeat the same sentence pattern

- All output must be in English

Output ONLY a JSON object (no other text):
{"items": [
  {"object": "object_name", "query": "description",
   "instruction": "The question you generated"},
```

```
{"object": "object_name", "query": "description",
 "instruction": "The question you generated"},
...one entry per object above...]}
```

E HYPERPARAMETER SWEEP DATA

Tables 5–7 report the full experimental data for the hyperparameter sweeps. All models are trained from Qwen3-VL-Embedding-2B with LoRA fine-tuning. The baseline configuration (GF-Embedding) uses τ=0.01, LoRA α/rank=2 (rank=8, α=16), learning rate $=1\times10^{-5}$, and batch size $=128$. In each sweep, one hyperparameter is varied while the remaining hyperparameters are held at baseline.

Table 5: Temperature (τ) sweep result.

τ	MMEB_VQA	MMEB_CLS	Hit@1	ISA	KIA	IPR	IFA	IMS	IDA	ICE
0.01	74.60	70.50	11.24	55.33	46.28	95.05	88.33	15.86	53.85	23.59
0.02	74.40	70.50	12.04	57.73	48.25	95.02	89.01	16.90	54.75	24.71
0.03	74.30	70.50	12.88	59.43	49.76	94.96	89.52	17.65	55.25	25.70
0.04	74.30	70.30	13.30	61.12	51.32	94.91	90.01	18.16	55.79	26.55
0.05	74.20	70.20	13.54	61.42	51.75	94.89	90.06	18.36	55.70	26.71
0.06	74.10	70.20	13.84	61.47	51.87	94.84	90.17	18.54	55.58	26.67
0.07	74.00	70.20	13.96	60.91	51.53	94.84	89.98	18.57	55.25	26.56
0.08	73.90	70.10	13.48	60.07	50.87	94.84	89.67	18.40	54.71	26.21
0.09	73.80	70.10	13.36	59.31	50.49	94.90	89.59	18.40	54.48	26.19
0.10	73.70	70.10	12.90	58.32	49.95	94.97	89.48	18.08	54.06	25.89
0.11	73.70	70.10	13.02	58.39	49.98	94.96	89.41	18.02	53.92	25.81
0.12	73.70	70.00	12.68	57.77	49.52	94.98	89.31	17.87	53.65	25.55

Table 6: LoRA α/rank ratio sweep result.

α/rank	MMEB_VQA	MMEB_CLS	Hit@1	ISA	KIA	IPR	IFA	IMS	IDA	ICE
0.5	74.40	70.40	7.08	44.41	37.57	95.09	85.47	10.95	45.61	16.51
1	74.40	70.50	8.98	50.62	42.14	95.01	87.09	13.66	50.23	20.17
2	74.60	70.50	11.24	55.33	46.28	95.05	88.33	15.86	53.85	23.59
4	74.30	69.80	13.06	61.45	51.82	95.16	90.31	18.27	57.04	27.28
8	73.90	68.40	14.78	67.03	57.23	95.28	92.03	19.99	60.25	32.32
16	73.20	66.50	19.76	79.20	69.86	95.52	95.53	23.04	64.30	36.08
32	71.90	64.70	26.24	86.83	78.15	96.21	97.21	25.68	65.24	34.42

Table 7: Learning rate sweep result.

LR	MMEB_VQA	MMEB_CLS	Hit@1	ISA	KIA	IPR	IFA	IMS	IDA	ICE
1×10^{-5}	74.60	70.50	11.24	55.33	46.28	95.05	88.33	15.86	53.85	23.59
2×10^{-5}	74.20	69.40	14.00	64.48	54.60	95.16	91.25	19.12	58.01	28.44
3×10^{-5}	74.10	68.40	16.68	70.92	61.26	95.25	93.17	21.03	60.92	32.39
4×10^{-5}	73.30	67.30	20.16	78.31	68.40	95.52	95.30	22.75	62.94	33.77
5×10^{-5}	72.90	66.10	22.78	82.95	73.41	95.91	96.55	23.85	63.75	33.87
6×10^{-5}	72.40	64.80	24.62	85.05	75.89	96.20	97.00	24.99	64.72	34.99