HQS-TUNING: A HIDDEN-STATE QUALITY-GUIDED DATA FILTERING FRAMEWORK FOR EFFICIENT MULTIMODAL INSTRUCTION FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction fine-tuning is a key technique for enhancing the performance of Multimodal Large Language Models (MLLMs), but massive and noisy datasets often incur prohibitive training costs with diminishing returns. Recent studies suggest that filtering a small amount of high-quality data for instruction fine-tuning can achieve highly efficient and superior training outcomes. However, existing multimodal data filtering approaches predominantly depend on external evaluation models or predefined proxy metrics, without leveraging the internal perception of the target MLLM itself. This limitation often results in a mismatch between the filtering criteria and the actual learning requirements of the model being fine-tuned. To address these issues, we propose a novel perspective: the image-conditioned hidden states of MLLMs implicitly reflect their evaluation of multimodal training data quality. Based on this insight, we propose an evaluator-guided two-stage data filtering framework. It extracts the target MLLM's hidden states as representative features and builds a lightweight classification model using strict quality-tier supervision to score and select the optimal multimodal training subset. Our extensive experiments are conducted across two mainstream MLLMs and eight diverse multimodal benchmarks. ① In terms of data efficiency, our experiments demonstrate that fine-tuning on less than 1% of the large-scale LLaVA-665K dataset (merely 5K samples) enables our method to outperform models trained on the complete 665K dataset. ② In comprehensive benchmark evaluations, our approach consistently surpasses multiple state-of-the-art multimodal data selection algorithms (such as COMPACT, ICONS, and Self-Filter) under the same data budgets, maintaining robust performance advantages across varied data scales.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) (Yao, 2025) typically build on Large Language Models (LLMs) as their linguistic core, incorporating non-text modalities such as vision and achieving cross-modal understanding and generation through modality alignment. Recently, MLLMs have made impressive progress across a wide range of vision-language tasks, including visual question answering (Antol et al., 2015), image captioning (Xu et al., 2015), and visual reasoning (Johnson et al., 2017). Among these tasks, instruction tuning serves as a crucial bridge between pre-trained representations and downstream applications. By training on multimodal instruction-response data, models learn to follow user instructions, understand image content, and generate text outputs that meet task requirements, thereby enhancing their generalizability and adaptability to downstream tasks (Liu et al., 2023; Dai et al., 2023). Consequently, instruction tuning is a key technology driving performance improvements in MLLMs.

During the multimodal instruction tuning phase, early research primarily focused on scaling up data size, with fine-tuning dataset volumes growing steadily over time. For instance, instruction data grew rapidly from over a hundred thousand in early stages (e.g., 158K in LLaVA-Instruct (Liu et al., 2023)) to several hundreds of thousands (e.g., 665K in LLaVA-1.5 (Liu et al., 2024)), and subsequently to the million level (e.g., million-level datasets like ShareGPT4V (Chen et al., 2024a)) and the multi-million level (e.g., 3.2M in SVIT (Zhao et al., 2023)). However, as data scales surge, training costs skyrocket, and performance gains from simply stacking data gradually hit a bottle-

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.

neck. Against this backdrop, as models iterated, the research community's focus began shifting from blindly expanding quantity to deeper exploration of instruction quality. The LIMA model proposed by Zhou et al. (2023) offers a compelling perspective on this paradigm shift: their study shows that fine-tuning with just 1,000 carefully curated, high-quality instruction samples can substantially improve model alignment performance. This finding strongly suggests that in the era of massive datasets, developing efficient, practical data-filtering and selection algorithms to extract a small, high-quality subset from vast, noisy training sets can not only drastically reduce computational overhead but also achieve superior instruction-tuning results.

Building on this, data selection has emerged as a hot research direction in multimodal instruction tuning. On one hand, some studies focus on assessing sample utility using difficulty or quality signals, such as calculating the perplexity of text responses via reference language models (Marion et al., 2023), or deriving difficulty scores directly from the vision-language model to select challenging and diverse samples (Chen et al., 2024c). On the other hand, many studies focus on data distribution and representational diversity. These approaches conduct selection by eliminating semantically redundant samples (Abbas et al., 2023), balancing sample difficulty and local diversity through graph message passing (Maharana et al., 2024), or extracting representative prototypes within a self-supervised feature space (Sorscher et al., 2022). Other research has also explored selection based on cross-task consensus influence (Wu et al., 2025) and optimized training subsets by increasing compositional complexity (Wu et al., 2026). Overall, most existing methods rely on external evaluation models or predefined proxy metrics and *fail to adequately incorporate the multimodal internal perception of the target model being fine-tuned*. This discrepancy often creates a mismatch between the selection criteria and the model's actual learning needs.

To reduce reliance on external proxy signals and improve target-model alignment, we assess multimodal data quality using internal representations from the target MLLM. The method combines two complementary ideas. At the representation level, we extend the hidden-state principle of CPQS-Tuning (Ren et al., 2026) from text-only instructions to multimodal instruction data. Specifically, we extract **image-conditioned hidden states** as the target MLLM jointly processes each sample's image and text. These states encode the model's response to multimodal content and provide target-aligned features for quality prediction. At the supervision level, we construct **strict quality tiers** from evaluator-assigned scores (Ouyang et al., 2022), excluding ambiguous samples and retaining only clearly high- and low-quality examples. These labels supervise a lightweight classifier over the image-conditioned hidden states. The trained classifier ranks the full data pool by predicted high-quality probability and selects the subset best aligned with target-model fine-tuning.

Based on the aforementioned concepts, we design an evaluator-guided two-stage CNN selection framework executing four specific steps (see details in Section 3): **(1) Multimodal Quality Evaluator and Initial Supervision**: We use a multimodal evaluator to score a small calibration set, partitioning initial "high-quality" and "low-quality" samples. **(2) Stage-I Hidden-State Quality Scorer and Coarse Candidate Mining**: We extract the target MLLM's hidden states on these initial samples to train a Stage-I CNN, using it to rapidly mine coarse candidate sets from both ends of the quality spectrum within a large-scale data pool. **(3) Evaluator-Guided Re-evaluation and Strict Label Refinement**: The evaluator re-evaluates only the coarse candidate set, discarding samples with ambiguous scores to construct high- and low-quality label sets with a strict quality boundary. **(4) Stage-II Hidden-State Quality Scorer and Final Data Ranking**: We train a Stage-II CNN using the strict label set. During final prediction, this model analyzes the target MLLM's hidden states for each instruction and outputs its predicted probability of being high-quality, which serves as the final score to rank the dataset and select the Top-K high-quality training subset.

Through extensive experiments, we validated that our algorithm performs excellently in multimodal instruction data selection tasks. We tested our proposed method on the large-scale LLaVA-665K dataset (Liu et al., 2024) across two distinct vision-language models (Qwen3.5-9B (Team, 2026) and Qwen3-VL-4B-Instruct (Team, 2025)). The experimental results demonstrate that the data subset selected by our method accounts for less than 1% of the original dataset (i.e., merely 5K samples), yet its downstream performance exceeds that of models trained on the complete 665K dataset. Furthermore, at identical data scales, our algorithm consistently surpasses multiple state-of-the-art data selection baselines (such as COMPACT (Wu et al., 2026), ICONS (Wu et al., 2025), and Self-Filter (Chen et al., 2024c)) across eight multimodal benchmark tests. Comprehensive data scaling and ablation studies also confirm that our two-stage strict-label refinement strategy effectively

extracts high-quality samples, maintaining a stable performance advantage over baseline methods across various data budgets ranging from 1K to 65K.

The main contributions of this paper can be summarized as follows:

Concept We show that the image-conditioned hidden states of MLLMs implicitly evaluate multimodal training data quality, introducing a novel paradigm for data selection based on the model's internal perception.

Method We propose an evaluator-guided two-stage CNN filtering framework. By constructing contrastive labels with strict quality boundaries, this framework efficiently and accurately extracts the optimal high-quality multimodal data subset that best aligns with the target model's fine-tuning needs.

Study We present extensive empirical studies utilizing two mainstream vision-language models (Qwen3.5-9B (Team, 2026) and Qwen3-VL-4B-Instruct (Team, 2025)) across eight multimodal benchmarks. The results indicate that the proposed method surpasses full-dataset fine-tuning with a minimal data budget (e.g., 5K) and consistently outperforms state-of-the-art data selection algorithms across various data scales.

2 MOTIVATION

Existing multimodal data-selection methods estimate sample utility using MLLM-based difficulty, semantic redundancy or diversity, cross-task influence, or compositional complexity (Maharana et al., 2024; Wu et al., 2026). Although these methods can reduce training data volume, they generally depend on external models or auxiliary evaluation mechanisms, making data filtering expensive and not guaranteeing optimal downstream performance. In particular, samples judged high quality by a proxy evaluator may not be the most suitable for the target model being fine-tuned.

To address this limitation, we propose using the target vision-language model's image-conditioned hidden states as an internal quality signal. The model processes the image, instruction, and answer jointly, so its hidden states may encode information about the usefulness and quality of each multimodal training example. We further hypothesize that this signal is easier to learn when the supervision examples exhibit clear quality contrast: clearly good and clearly poor examples should define a more stable quality direction than examples with only marginal quality differences. A lightweight CNN trained on these hidden states can therefore rank new multimodal samples without repeatedly querying an external evaluator.

To verify these hypotheses, we construct an independent 200-example hold-out set from GPT-5.5-verified VQAv2 data (Jia et al., 2025), containing 100 high-quality and 100 low-quality examples with distinct image groups. The hold-out is disjoint from both CNN supervision sets at the sample and image-group levels. High-quality examples have an overall GPT-5.5 (Anjum et al., 2026) score of at least 4.5 and scores of at least 4.0 on the image, text, alignment, and answer dimensions, whereas low-quality examples score at most 3.0. These opposite-end thresholds provide a broadly interpretable quality contrast rather than a marginal score difference. We mean-pool the final-layer hidden states of a frozen Qwen3.5-9B model and train a linear probe using stratified five-fold cross-validation. We then compare two CNN scorers under identical architectures, class sizes, backbones, and optimization settings. The *Strict-gap* scorer is trained with a larger quality gap, using 800 score-5.0 examples and 800 examples scoring at most 3.75, whereas the *Weak-gap* scorer is trained with a smaller gap, using the same 800 score-5.0 examples and 800 examples scoring exactly 4.5. Both scorers rank the same held-out set. We report hidden-state probe AUROC and, when selecting 100 samples, the overlap between the CNN-selected high/low tails and the GPT-5.5 teacher-defined tails. A random-subset reference is included; because the hold-out is balanced, its expected overlap is 50%, while the plotted values are observed Monte Carlo means.

As shown in Fig. 1(a), the hidden-state representations yield a five-fold out-of-fold AUROC of 0.960 (95% confidence interval: $[0.933, 0.983]$), demonstrating that multimodal quality is readable from the frozen model representation. In Fig. 1(b), the Strict-gap estimator recovers 94% of both the GPT-5.5 high-quality and low-quality tails when selecting 100 samples, whereas the Weak-gap estimator recovers 81% for both tails. Random selection recovers only 49.88% and 50.12%, respectively, with a 95% interval of approximately $[43\%, 57\%]$. Therefore, a larger, more interpretable supervision gap yields a more discriminative hidden-state quality direction than a weak gap or random selection.

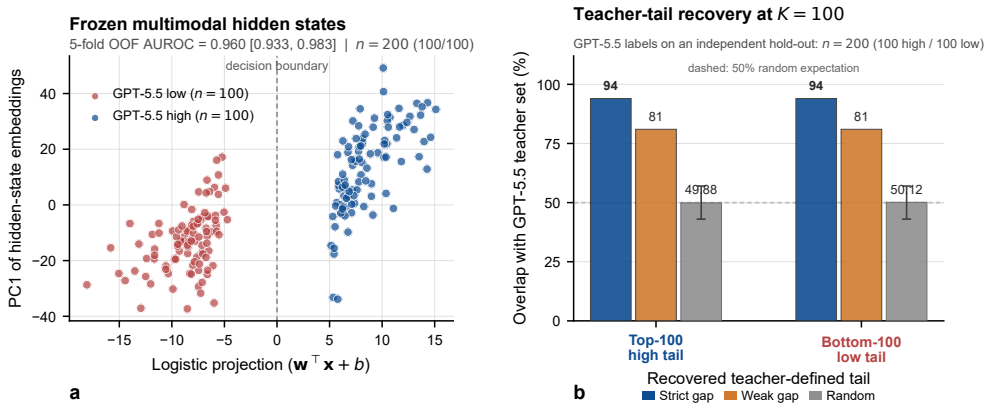

Figure 1: Hidden-state quality readability and teacher-tail recovery.

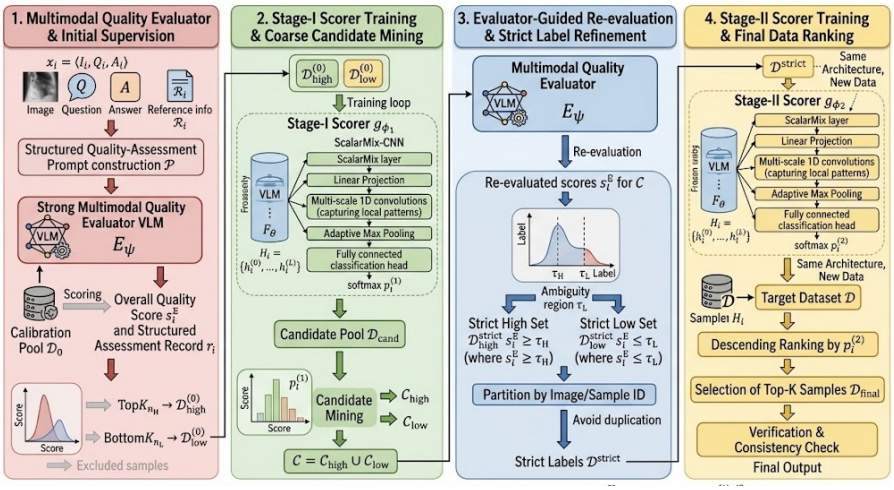

Figure 2: Overall algorithm architecture diagram (Gemini was used only for figure post-processing to improve layout, font clarity, and color consistency; all scientific content was manually verified by the authors).

These results motivate our hidden-state CNN evaluator, while the downstream fine-tuning stages and their ablations are studied separately.

3 Proposed Method

We propose a quality-evaluator-guided two-stage CNN framework for selecting high-quality samples from large-scale multimodal instruction data. As shown in Figure 2, it comprises four phases centered on two rounds of CNN scorer training. First, the evaluator labels a small calibration set. Second, these labels train a Stage-I CNN on image-conditioned hidden states from a frozen vision-language model to rank a large candidate pool. Third, the evaluator re-evaluates samples from both ranking extremes to construct strict labels while excluding ambiguous cases. Finally, the refined labels train a Stage-II CNN, whose high-quality probability ranks the target dataset for Top-K selection.

3.1 Multimodal Quality Evaluator and Initial Supervision

To obtain the initial supervision needed to train the Stage-I scorer, we first use a strong MLLM E_ψ, with parameters or service configuration ψ, as an evaluator. It annotates an additional calibration

pool with coarse quality signals. For a sample $x_i = \langle I_i, Q_i, A_i \rangle$, i is its index, I_i is the image or image reference, Q_i is the user question or dialogue, and A_i is the target answer. Given the structured prompt $\mathcal{P}$ and optional reference information $\mathcal{R}_i$ (for example, a reference answer or candidate choices), the evaluator returns

$$(s_i^{\mathrm{E}}, r_i) = E_\psi(\mathcal{P}(I_i, Q_i, A_i, \mathcal{R}_i)),$$

where s_i^{E} is the scalar overall quality score and r_i records dimension scores, validity flags, and evidence. The evaluator aggregates image quality, text quality, image–text alignment, answer quality, training utility, and safety into s_i^{E}. Only s_i^{E} is used to form binary high/low supervision; r_i is retained for interpretation and error analysis.

Let $\mathcal{D}_{\mathrm{target}}$ be the target dataset that will eventually be ranked for downstream fine-tuning. To train our scorers, we utilize a separate large source dataset $\mathcal{D}_{\mathrm{src}}$. We first sample a small initial calibration pool $\mathcal{D}^{(0)} \subset \mathcal{D}_{\mathrm{src}}$ used only for these initial annotations. The source and target datasets are disjoint; $\mathcal{D}^{(0)}$ is not part of the target dataset $\mathcal{D}_{\mathrm{target}}$ to be ranked. After invalid or unevaluable samples are removed, the valid pool is $\mathcal{D}^{(0)}_{\mathrm{valid}}$. We select n_{H} high-score and n_{L} low-score records from this pool:

$$\mathcal{D}^{(0)}_{\mathrm{high}} = \mathrm{TopK}_{n_{\mathrm{H}}}\{(x_i, s_i^{\mathrm{E}}) \mid x_i \in \mathcal{D}^{(0)}_{\mathrm{valid}}\}, \qquad \mathcal{D}^{(0)}_{\mathrm{low}} = \mathrm{BottomK}_{n_{\mathrm{L}}}\{(x_i, s_i^{\mathrm{E}}) \mid x_i \in \mathcal{D}^{(0)}_{\mathrm{valid}}\}.$$

Here, $\mathrm{TopK}_{n_{\mathrm{H}}}$ and $\mathrm{BottomK}_{n_{\mathrm{L}}}$ return the n_{H} highest- and n_{L} lowest-scoring records, respectively. These are calibration-label counts, distinct from the re-evaluation counts $m_{\mathrm{H}}, m_{\mathrm{L}}$ and the final selection budget K. Intermediate-score records are excluded to reduce label ambiguity, and invalid evaluations are excluded rather than treated as low-quality examples. The two tails therefore provide the coarse binary labels for Stage-I training.

3.2 Stage-I Hidden-State Quality Scorer and Coarse Candidate Mining

To rapidly screen a much larger pool and identify samples at opposite ends of the predicted quality distribution, we train a lightweight Stage-I hidden-state scorer g_{ϕ_1} on $\mathcal{D}^{(0)}_{\mathrm{high}} \cup \mathcal{D}^{(0)}_{\mathrm{low}}$. Here, g_{ϕ_1} is the scorer function and ϕ_1 denotes its trainable parameters. The scorer provides inexpensive approximate scores for candidate mining, after which the selected candidates are re-evaluated by the quality evaluator. Let F_θ be the target MLLM with frozen parameters θ. For each sample, we feed (I_i, Q_i, A_i) to F_θ and extract hidden states from its $L+1$ selected language-model layers:

$$H_i = F_\theta(I_i, Q_i, A_i) = \left\{h_i^{(0)}, h_i^{(1)}, \ldots, h_i^{(L)}\right\}, \qquad h_i^{(\ell)} \in \mathbb{R}^{T_i \times d_F}.$$

where $\ell = 0, \ldots, L$, $L+1$ is the number of selected layers, $h_i^{(\ell)}$ is the token-by-feature matrix at layer ℓ, T_i is the retained token count, and d_F is the hidden width. Thus, H_i is a set of layer representations containing image, question, and answer information.

The scorer first uses ScalarMix to learn normalized weights over the $L+1$ layers and combine them into one token sequence. A linear projection maps the feature width from d_F to the scorer width d_P, where d_P is the projected feature dimension. One-dimensional convolutions with kernel widths $k \in \{3, 4, 5\}$, adaptive max pooling, and a two-logit classification head then produce class logits. For class $c \in \{0, 1\}$, where $c = 0$ denotes low quality and $c = 1$ denotes high quality, the ranking score is

$$p_i^{(1)} = \mathrm{softmax}\left(g_{\phi_1}(H_i)\right)_1,$$

where $p_i^{(1)} \in [0, 1]$ is the Stage-I high-quality probability; the superscript identifies the scorer stage, not the class.

After training, we apply the Stage-I scorer to a large candidate pool $\mathcal{D}_{\mathrm{cand}} \subseteq \mathcal{D}_{\mathrm{src}} \setminus \mathcal{D}^{(0)}$ to obtain approximate scores for many samples at low cost. This mining pool is drawn from the same source dataset as the calibration pool, but they are mutually exclusive. Furthermore, $\mathcal{D}_{\mathrm{cand}}$ is entirely separate from the downstream target dataset $\mathcal{D}_{\mathrm{target}}$. Let m_{H} and m_{L} denote the numbers of high- and low-tail candidates sent for evaluator re-evaluation, and define the scored candidate records as $\mathcal{S}_{\mathrm{cand}} = \{(x_i, p_i^{(1)}) \mid x_i \in \mathcal{D}_{\mathrm{cand}}\}$:

$$\mathcal{C}_{\mathrm{high}} = \mathrm{TopK}_{m_{\mathrm{H}}}(\mathcal{S}_{\mathrm{cand}}), \qquad \mathcal{C}_{\mathrm{low}} = \mathrm{BottomK}_{m_{\mathrm{L}}}(\mathcal{S}_{\mathrm{cand}}).$$

The resulting records retain each sample and its Stage-I score. Re-evaluating only these two tails concentrates the expensive evaluator calls on samples most likely to provide contrasting quality evidence.

3.3 Evaluator-Guided Re-evaluation and Strict Label Refinement

Because Stage-I labels are coarse, we submit the candidate set $\mathcal{C} = \mathcal{C}_{\text{high}} \cup \mathcal{C}_{\text{low}}$ to the same evaluator. Here, $\mathcal{C}$ is a set of $(x_i, p_i^{(1)})$ records, deduplicated by image or source-sample identifier before labeling.

Using the re-evaluated scores, we define strict labels with $\tau_{\text{L}} < \tau_{\text{H}}$, where τ_{H} is the minimum high-quality score and τ_{L} is the maximum low-quality score:

$$\mathcal{D}_{\text{high}}^{\text{strict}} = \left\{ x_i \mid (x_i, p_i^{(1)}) \in \mathcal{C}, s_i^{\text{E}} \geq \tau_{\text{H}} \right\}, \quad \mathcal{D}_{\text{low}}^{\text{strict}} = \left\{ x_i \mid (x_i, p_i^{(1)}) \in \mathcal{C}, s_i^{\text{E}} \leq \tau_{\text{L}} \right\}, \quad \tau_{\text{L}} < \tau_{\text{H}}.$$

Samples with $\tau_{\text{L}} < s_i^{\text{E}} < \tau_{\text{H}}$ or failed evaluations are excluded from Stage-II training; equality at either threshold belongs to the corresponding class. The strict-label set is

$$\mathcal{D}^{\text{strict}} = \mathcal{D}_{\text{high}}^{\text{strict}} \cup \mathcal{D}_{\text{low}}^{\text{strict}}.$$

Image or source-sample groups are kept within one split to prevent leakage. These labels come from evaluator scores, rather than Stage-I hard predictions.

3.4 Stage-II Hidden-State Quality Scorer and Final Data Ranking

We train the Stage-II scorer g_{ϕ_2}, with separate trainable parameters ϕ_2, on $\mathcal{D}^{\text{strict}}$. It uses the same frozen F_θ, hidden states, and ScalarMix-CNN architecture as Stage I; only the scorer parameters and supervision differ. Thus, Stage II refines the quality boundary learned from the calibration tails.

For each sample $x_i \in \mathcal{D}_{\text{target}}$, we extract H_i with the frozen MLLM and compute its Stage-II high-quality probability:

$$p_i^{(2)} = \text{softmax}\left(g_{\phi_2}(H_i)\right)_1.$$

The superscript (2) identifies the scorer stage. We sort the $|\mathcal{D}_{\text{target}}|$ target samples by $p_i^{(2)}$ and select the final budget K (or all samples when $K \geq |\mathcal{D}_{\text{target}}|$):

$$\mathcal{D}_{\text{final}} = \text{TopK}_K\left(\left\{ (x_i, p_i^{(2)}) \right\}_{i=1}^{|\mathcal{D}_{\text{target}}|} \right).$$

The probability gives a continuous ordering and supports different budgets without retraining. Before downstream fine-tuning, we verify image paths, dialogue fields, and image–placeholder consistency.

4 Experiments

4.1 Experimental Setup

Data and models. VQAv2 (Jia et al., 2025) is used only to train the two-stage quality scorer. GPT-5.5 evaluates a 2K calibration set; its 800 high- and 800 low-score tails form 1,600 Stage-I labels, of which 1,280 are used for training. Stage I then mines candidates from 100K VQAv2 examples; evaluator re-scoring retains 2,200 strict examples per class for Stage II, yielding 4,400 labels, of which 3,520 are used for training. All splits are image-disjoint. We evaluate all selection methods on the same validated LLaVA-1.5 Mix-665K pool (Liu et al., 2024) of 665,298 samples. Each selection method chooses 65,536 examples from this pool, while Full-665K uses all valid samples. Downstream experiments use Qwen3.5-9B (Team, 2026) and Qwen3-VL-4B-Instruct (Team, 2025), with identical pretrained checkpoints and training configurations across methods.

Baselines. We compare our method with eight representative data-selection methods and one full-data reference under identical model architectures and training configurations. The eight methods, each selecting 65K examples from LLaVA-665K, are (1) Random, (2) the Perplexity approach (Marion et al., 2023) (arXiv 2023), (3) the ICML 2023 method SemDeDup (Abbas et al., 2023), (4) the ICLR 2024 method D2-Pruning (Maharana et al., 2024), (5) the NeurIPS 2022 method Self-Sup (Sorscher et al., 2022), (6) the ACL 2024 method Self-Filter (Chen et al., 2024c), (7) the ICONS

Table 1: Baseline comparison on Qwen3.5-9B across eight multimodal benchmarks. All values except MME are percentages. The "-65K" suffix identifies the training-data budget used by the corresponding method.

Method	CV-Bench	InfoVQA	MME	MMStar	SeedB2+	TextVQA	LLaVA-W	MM-Vet	Rel.
Full-665K	86.35	86.41	2353.06	68.29	73.17	84.80	77.80	65.23	100.00
D2-Pruning-65K	86.66	85.81	2412.11	68.14	73.03	84.79	77.60	64.63	100.07
ICONS-65K	87.68	86.76	**2461.13**	69.71	73.12	84.51	72.40	65.00	100.11
SemDeDup-65K	87.07	86.81	2411.61	68.64	70.93	83.96	77.40	67.43	100.39
Perplexity-65K	87.79	86.07	2446.58	69.07	73.12	84.89	79.69	64.46	100.96
Self-Sup-65K	87.53	86.77	2399.36	69.48	72.77	84.57	77.40	67.57	100.97
Self-Filter-65K	86.43	85.83	2444.78	70.26	72.99	84.76	77.76	67.41	101.15
Random-65K	86.66	86.73	2442.93	69.73	73.39	**84.94**	80.90	64.27	101.20
COMPACT-65K	87.07	**86.84**	2437.47	69.96	**73.43**	83.93	75.10	69.68	101.26
Ours-5K	**87.83**	86.14	2454.07	**70.90**	72.86	83.74	**84.30**	69.91	**102.92**
Ours-65K	87.30	86.80	2430.91	69.85	72.60	84.16	80.70	**71.10**	102.29

approach (Wu et al., 2025) (arXiv 2025), and (8) the ICLR 2026 method COMPACT (Wu et al., 2026). (9) Full-665K denotes the complete LLaVA-665K visual instruction-tuning dataset used in the CVPR 2024 model LLaVA-v1.5 (Liu et al., 2024) and serves as the full-data reference.

Training and deployment. All downstream vision-language models are fine-tuned with LLaMA-Factory (Zheng et al., 2024) and LoRA for one epoch, using a learning rate of 1×10^{-4} and a fixed random seed of 42. All other training settings are held constant across methods, so only the selected training subset varies. After training, each backbone and its LoRA adapter are loaded with vLLM (Kwon et al., 2023) for inference. We use lmms-eval (Zhang et al., 2024) to evaluate each model on the corresponding multimodal benchmarks under the same generation and evaluation settings. For completeness, Appendix A.1 gives the full training, inference, API, and runtime configuration, while Appendix A.2 provides the quality-evaluator prompt and strict labeling criteria

Evaluation metrics. Following COMPACT (Wu et al., 2026), we evaluate all methods on eight multimodal benchmarks. CV-Bench (Tong et al., 2024), MMStar (Chen et al., 2024b), SeedBench2+ (Li et al., 2024), and TextVQA (Singh et al., 2019) use accuracy-based metrics; InfoVQA (Mathew et al., 2022) uses ANLS; and MME (Fu et al., 2025) reports the raw sum of its perception and cognition scores. LLaVA-Wild (Liu et al., 2023) and MM-Vet (Yu et al., 2024) use their respective model-based protocols, with the same GPT-5.6-sol evaluator, prompts, and scoring settings applied to all methods. We aggregate the eight native benchmark scores using mean relative performance:

$$\mathrm{Rel}(m) = \frac{100}{8} \sum_{b=1}^{8} \frac{S_{m,b}}{S_{\mathrm{ref},b}},$$

where $S_{m,b}$ is the score of method m on benchmark b, and $S_{\mathrm{ref},b}$ is the score of the same backbone trained on the reference dataset. Thus, we first express each benchmark score relative to its corresponding full-data score, then average the eight resulting ratios. Full-665K serves as the reference for the main comparison, whereas Full-10K serves as the reference for the two-stage ablation. The reference model therefore has a $\mathrm{Rel.}$ score of 100%, and higher values indicate better overall performance.

4.2 Main Results

We evaluate our method against baselines on Qwen3.5-9B and Qwen3-VL-4B-Instruct across eight multimodal benchmarks, as shown in Tables 1 and 2. Method suffixes denote training budgets (Full-665K, 65K for baselines/Ours-65K, and 5K for Ours-5K). Except for raw MME scores, all metrics are percentages. Rel. (%) indicates average performance relative to Full-665K (100.00), with best results bolded.

Table 1 presents the results on Qwen3.5-9B. **Overall, our method with the matched 65K budget outperforms all 65K baselines, while scaling down to just 5K samples surprisingly achieves the best overall performance.** Specifically, (i) with 65K samples (Ours-65K), our method reaches a relative score of 102.29%, exceeding the top baseline (COMPACT-65K) by 1.03 percentage points

Table 2: Baseline comparison on Qwen3-VL-4B-Instruct across eight multimodal benchmarks. All values except MME are percentages.

Method	CV-Bench	InfoVQA	MME	MMStar	SeedB2+	TextVQA	LLaVA-W	MM-Vet	Rel.
Full-665K	84.00	80.21	2328.23	65.10	70.00	81.41	72.60	60.37	100.00
ICONS-65K	84.53	80.75	2312.31	64.84	70.44	80.49	72.40	59.82	99.82
COMPACT-65K	85.41	80.52	**2363.85**	65.38	70.09	80.33	71.30	60.00	100.05
Self-Sup-65K	84.61	80.81	2278.81	64.78	70.79	81.42	75.10	59.13	100.17
SemDeDup-65K	84.46	81.08	2260.26	63.45	68.07	81.52	76.60	62.61	100.35
Perplexity-65K	80.63	80.83	2332.44	64.07	**70.84**	81.08	76.30	63.12	100.73
Random-65K	84.76	80.50	2286.27	65.92	70.18	80.81	77.50	60.96	101.00
D2-Pruning-65K	83.40	**81.32**	2346.71	63.69	70.09	81.52	75.50	63.95	101.19
Self-Filter-65K	85.03	80.08	2350.46	**66.79**	70.18	**81.61**	**79.10**	59.91	101.66
Ours-5K‡	85.33	80.21	2338.47	62.23	70.09	80.77	72.90	**64.82**	100.59
Ours-65K	**86.01**	80.71	2350.62	65.47	70.31	81.22	76.80	64.77	**102.23**

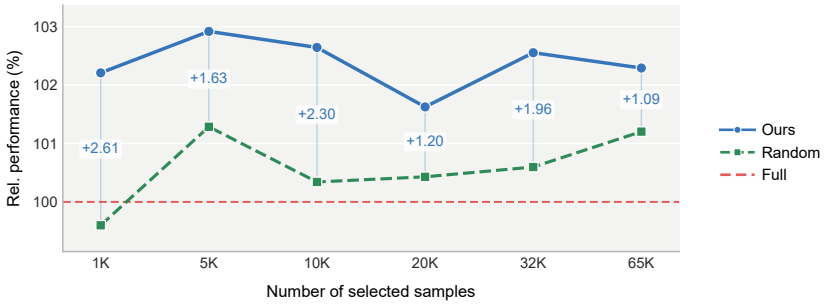

Figure 3: Aggregate relative performance across selection budgets on Qwen3.5-9B. Solid blue and dashed green lines denote Ours and Random, respectively; the red dashed line marks Full-665K at 100%. Labels indicate the Ours–Random differences in percentage points, and the horizontal axis is logarithmic.

and outperforming Full-665K on six benchmarks; (ii) at an extremely compact budget of 5K samples (Ours-5K), our method further improves the relative score to 102.92%, outperforming Full-665K and our 65K subset by 2.92 and 0.63 points, respectively; and (iii) remarkably, our 5K setting achieves the highest scores on CV-Bench, MMStar, and LLaVA-Wild while using fewer than 1% of the original training data.

Table 2 presents the corresponding results on Qwen3-VL-4B-Instruct. **Overall, our method with 65K samples ranks first across all evaluated approaches, while our 5K setting remains competitive, consistently outperforming the Full-665K baseline.** Specifically, (i) with 65K samples, our method attains a peak relative score of 102.23%, exceeding the top-performing baseline (Self-Filter-65K) by 0.57 percentage points; (ii) it outperforms the full dataset on seven benchmarks, with TextVQA as the sole exception; and (iii) even when trained on only 5K samples, our method reaches 100.59% and leads all methods on MM-Vet, demonstrating that the optimal data-selection budget varies across downstream backbones. Beyond the multimodal benchmarks, we evaluate transferability to general language tasks and compare hidden-state sources in Appendices A.4 and A.5.

4.3 Effect of Selection Budget

To evaluate our algorithm across different data-selection budgets, we compare it against random selection on Qwen3.5-9B at scales ranging from 1K to 65K examples. As illustrated in Figure 3, **our method consistently outperforms random selection across all evaluated budgets.** Specifically, (i) our advantage is most pronounced at the 1K budget (+2.61 percentage points), demonstrating high data efficiency under strict constraints; (ii) from 5K to 65K, our method maintains steady gains of 1.09 to 2.30 points over random selection; and (iii) it achieves its peak relative score at only 5K samples (102.92%), surpassing Full-665K with less than 1% of the training data. To assess

Table 3: Ablation of the multimodal quality evaluator and two-stage refinement on a fixed 10K candidate pool. Each selection method retains the top 1K samples, whereas Full-10K uses the entire pool. All values except MME are percentages.

Method	CV-Bench	InfoVQA	MME	MMStar	SeedB2+	TextVQA	LLaVA-W	MM-Vet	Rel.
Evaluator-1K	**87.53**	85.95	2438.85	69.65	71.94	83.84	80.10	68.90	100.60
Stage-I-1K	87.07	86.33	2412.55	68.39	72.82	83.37	**82.00**	67.52	100.36
Stage-II (Ours)-1K	87.45	85.85	**2443.56**	**71.26**	72.16	83.43	79.30	**69.13**	**100.78**
Full-10K	87.41	**86.77**	2430.56	69.02	**73.30**	**85.01**	76.40	67.06	100.00

whether this gain depends on the scoring head, we further compare CNN, MLP, and Transformer architectures in Appendix A.3.

4.4 Effect of Two-Stage Refinement

To validate the effectiveness of our proposed two-stage refinement pipeline, we conduct an ablation study using a fixed 10K candidate pool sampled from LLaVA-665K. As shown in Table 3, Evaluator-1K, Stage-I-1K, and Stage-II-1K each select the top 1K samples using their respective scoring mechanisms, and all configurations fine-tune Qwen3.5-9B under identical settings, with Full-10K serving as the 100.00% reference. **Overall, our full two-stage method (Stage-II-1K) achieves the highest aggregate relative performance while retaining only one-tenth of the candidate pool.** Specifically, (i) the standalone Evaluator-1K and Stage-I-1K reach 100.60% and 100.36%, respectively; (ii) Stage-II-1K further improves the relative score to 100.78%, exceeding the intermediate variants by 0.18 and 0.42 percentage points; and (iii) while Full-10K remains best on InfoVQA, SeedBench2+, and TextVQA, all three 1K filtered subsets consistently outperform the full 10K baseline in aggregate performance.

5 Related Work

Visual Instruction Tuning. Visual instruction tuning (VIT) connects a pretrained vision encoder with a language model and trains the resulting system on multimodal instructions. LLaVA (Liu et al., 2023) introduced this framework, and LLaVA-1.5 (Liu et al., 2024) improved its connector and training data. As VIT datasets have grown, attention has shifted from data volume to sample informativeness and complexity. COMPACT (Wu et al., 2026), for example, constructs questions that combine multiple visual capabilities to improve reasoning under a limited data budget. These approaches primarily construct or augment training data; selecting informative samples from a fixed multimodal pool remains a complementary problem.

Visual Instruction Data Selection. Existing methods estimate sample utility from quality or difficulty, diversity, influence, or compositionality. Perplexity-based selection (Marion et al., 2023) and Self-Filter (Chen et al., 2024c) use difficulty-related signals, whereas SemDeDup (Abbas et al., 2023), D2-Pruning (Maharana et al., 2024), and Self-Sup (Sorscher et al., 2022) emphasize diversity or representation coverage. ICONS (Wu et al., 2025) selects examples by their cross-task influence, while CPQS-Tuning (Ren et al., 2026) uses hidden states to estimate instruction quality for text-only tuning. These approaches largely rely on external models, proxy scores, or text-only internal signals. **In contrast, our method combines multimodal evaluator supervision with image-conditioned hidden states from the target vision-language model, jointly assessing the visual and textual quality of each instruction sample.**

6 Conclusion

We introduced a model-aware selector that combines multimodal evaluator supervision with image-conditioned hidden states from the target MLLM. A two-stage CNN scorer refines quality boundaries and ranks samples continuously. On LLaVA-665K, selected subsets consistently outperform matched-budget baselines; on Qwen3.5-9B, only 5K examples exceed the Full-665K reference. These results support model-internal perception as a practical signal for efficient multimodal data curation, while the preferred budget remains backbone- and task-dependent.

AI USE STATEMENT

In this work, we used generative AI tools for text editing and language polishing, and used Gemini only for post-processing the layout, font clarity, and colour consistency of one method figure (see Figure 2, and we have also noted the use of AI in the caption of Figure 2). The GPT-5.5 evaluator and Qwen-family models used as experimental components are described explicitly in the method and experimental sections; they are not authoring assistants. The scientific content, experimental design, data, and claims were created and verified by the authors, and we introduced no AI-generated data outside the stated experimental pipeline. We reviewed all AI-assisted text and figure edits and take full responsibility for the final content of this work, including all claims and artifacts.

REPRODUCIBILITY STATEMENT

To ensure the full reproducibility of our experimental results and the proposed framework, we have provided the complete, anonymized source code in the supplementary materials. This includes all data evolution scripts, training and validation pipelines, and evaluation kernels required to replicate the results presented in this paper.

REFERENCES

Amro Abbas, Kushal Tirumala, Daniel Simig, Surya Ganguli, and Ari S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *CoRR*, abs/2303.09540, 2023. doi: 10.48550/ARXIV.2303.09540. URL https://doi.org/10.48550/arXiv.2303.09540.

Usama Bin Anjum, Kana Hirasawa, and Urooj Fatima. A comparative study of chatgpt in scientific writing assistance: accuracy, style, and hallucination patterns across gpt-4o, gpt-5.0, and GPT-5.1. *AI Ethics*, 6(2):204, 2026. doi: 10.1007/S43681-026-01075-6. URL https://doi.org/10.1007/s43681-026-01075-6.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XVII*, volume 15075 of *Lecture Notes in Computer Science*, pp. 370–387. Springer, 2024a. doi: 10.1007/978-3-031-72643-9_22. URL https://doi.org/10.1007/978-3-031-72643-9_22.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b. URL http://papers.nips.cc/paper_files/paper/2024/hash/2f8ee6a3d766b426d2618e555b5aeb39-Abstract-Conference.html.

Ruibo Chen, Yihan Wu, Lichang Chen, Guodong Liu, Qi He, Tianyi Xiong, Chenxi Liu, Junfeng Guo, and Heng Huang. Your vision-language model itself is a strong filter: Towards high-quality instruction tuning with data selection. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pp. 4156–4172. Association for Computational Linguistics, 2024c. doi: 10.18653/V1/2024.FINDINGS-ACL.246. URL https://doi.org/10.18653/v1/2024.findings-acl.246.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models

with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/d79a27cf2772fe00be7f341efc0eb517-Abstract-Datasets_and_Benchmarks_Track.html.

Ziheng Jia, Zicheng Zhang, Jiaying Qian, Haoning Wu, Wei Sun, Chunyi Li, Xiaohong Liu, Weisi Lin, Guangtao Zhai, and Xiongkuo Min. Vqa^2: Visual question answering for video quality assessment. In Cathal Gurrin, Klaus Schoeffmann, Min Zhang, Luca Rossetto, Stevan Rudinac, Duc-Tien Dang-Nguyen, Wen-Huang Cheng, Phoebe Chen, and Jenny Benois-Pineau (eds.), *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pp. 6751–6760. ACM, 2025. doi: 10.1145/3746027.3754696. URL https://doi.org/10.1145/3746027.3754696.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In Jason Flinn, Margo I. Seltzer, Peter Druschel, Antoine Kaufmann, and Jonathan Mace (eds.), *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pp. 611–626. ACM, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *CoRR*, abs/2404.16790, 2024. doi: 10.48550/ARXIV.2404.16790. URL https://doi.org/10.48550/arXiv.2404.16790.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 26286–26296. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02484. URL https://doi.org/10.1109/CVPR52733.2024.02484.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. D2 pruning: Message passing for balancing diversity & difficulty in data pruning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=thbtoAkCe9.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *CoRR*, abs/2309.04564, 2023. doi: 10.48550/ARXIV.2309.04564. URL https://doi.org/10.48550/arXiv.2309.04564.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2582–2591. IEEE, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Yi Ren, Yanhui Li, Tianyi Zhang, and Diandong Liu. Cpqs-tuning: A model self-perception-based data filtering algorithm for efficient instruction fine-tuning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 8317–8326. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00851. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Singh_Towards_VQA_Models_That_Can_Read_CVPR_2019_paper.html.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/7b75da9b61eda40fa35453ee5d077df6-Abstract-Conference.html.

Qwen Team. Qwen3-vl technical report. *CoRR*, abs/2511.21631, 2025. doi: 10.48550/ARXIV.2511.21631. URL https://doi.org/10.48550/arXiv.2511.21631.

Qwen Team. Qwen3.5-omni technical report. *CoRR*, abs/2604.15804, 2026. doi: 10.48550/ARXIV.2604.15804. URL https://doi.org/10.48550/arXiv.2604.15804.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

Xindi Wu, Mengzhou Xia, Rulin Shao, Zhiwei Deng, Pang Wei Koh, and Olga Russakovsky. ICONS: influence consensus for vision-language data selection. *CoRR*, abs/2501.00654, 2025. doi: 10.48550/ARXIV.2501.00654. URL https://doi.org/10.48550/arXiv.2501.00654.

Xindi Wu, Hee Seung Hwang, Polina Kirichenko, Esin Tureci, and Olga Russakovsky. Visual compositional tuning. In *International Conference on Learning Representations*, volume 2026, pp. 126293–126316, 2026.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057. PMLR, 2015.

Yanyu Yao. A survey on multimodal large language models. In *Computer Science Undergradaute Conference 2025@ XJTU*, 2025.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. In

Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57730–57754. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/yu24o.html.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024. URL https://arxiv.org/abs/2407.12772.

Bo Zhao, Boya Wu, and Tiejun Huang. SVIT: scaling up visual instruction tuning. *CoRR*, abs/2307.04087, 2023. doi: 10.48550/ARXIV.2307.04087. URL https://doi.org/10.48550/arXiv.2307.04087.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL http://arxiv.org/abs/2403.13372.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: less is more for alignment. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/ac662d74829e4407ce1d126477f4a03a-Abstract-Conference.html.

A Additional Analyses

This appendix provides additional implementation details and controlled analyses that complement the main experiments. We present the quality-evaluator prompt and examine the scorer architecture, general-benchmark transfer, hidden-state source, and effect of the selection budget.

A.1 Implementation and Evaluation Details

The scorer supervision uses 2,000 VQAv2 samples for initial evaluation. We retain 800 examples from each score tail and split them by image identifier into 1,280/160/160 training, validation, and test examples. Stage I scores a 100,000-example candidate pool; we re-evaluate 5,000 high-scoring, 5,000 low-scoring, 3,000 uncertain, and 1,000 random examples. Strict re-evaluation retains 2,200 high- and 2,200 low-quality examples, split into 3,520/440/440 image-disjoint training, validation, and test sets.

All downstream models are fine-tuned with LLaMA-Factory (Zheng et al., 2024) and LoRA ($r = 8$, $\alpha = 16$, dropout 0), using learning rate 1×10^{-4}, per-device batch size 2, gradient accumulation 8, AdamW, cosine decay, warm-up ratio 0.1, one epoch, `bf16`, and seed 42. Inference uses lmms-eval (Zhang et al., 2024) with vLLM (Kwon et al., 2023), batch size 64, maximum context length 32,768, `bf16`, disabled Qwen thinking mode, and identical generation settings across methods.

Following Visual Compositional Tuning (Wu et al., 2026), CV-Bench uses accuracy, InfoVQA ANLS, MME the raw perception-plus-cognition sum, MMStar and SeedBench2+ overall accuracy, TextVQA exact match, and the native model-based protocols for LLaVA-Wild and MM-Vet. GPT-5.6-sol is used with the same prompt and scoring configuration for the latter two benchmarks. The relative score is computed from unrounded values, while tables display two decimals.

A.2 Quality-Evaluator Prompt Template

The following box gives the complete prompt template used by the multimodal quality evaluator. Braced expressions are replaced with sample-specific values.

Multimodal quality evaluator prompt

System prompt. You are a strict general multimodal dataset quality evaluator. Judge whether one image–question–answer sample is suitable for vision-language instruction tuning. Return JSON only and do not use markdown. Be strict about visual grounding, answer correctness, and image–text consistency. Do not reward fluent but unsupported answers. If image pixels are unavailable, use metadata-only assessment and do not claim visual evidence. Penalize image-independent questions, ambiguous references, missing answers, hallucinated visual content, wrong choice labels, malformed ShareGPT/LLaMA-Factory records, privacy leaks, and unsafe or biased content.

Assessment prompt. Evaluate one general multimodal training sample. Assign scores from 0.0 to 5.0: 5.0 excellent, 4.0 keepable, 3.0 review, 2.0 weak, 1.0 poor, and 0.0 unusable. Reserve scores above 4.5 for samples without material grounding or answer problems. Return compact JSON and keep each `issues` or `evidence` list to at most two short phrases.

Sample context. The prompt receives the image(s), image-path status, question, assistant answer, reference answer or candidate choices, inferred task label, candidate capability tags, question type, and conversation-turn count.

Image condition. When image pixels are unavailable, set the assessment mode to `metadata_only`; inspect only image-path metadata and do not infer visual content.

Task-specific rubric. The evaluator selects one rubric from the following set: `visual_qa` requires image-grounded answers and correct choice matching; `grounded_reasoning` requires reasoning over visible objects, relations, counts, text, or scene context; `image_captioning` requires accurate descriptions without hallucination; `ocr` requires readable image text and faithful transcription; `document_qa` requires reasoning over document fields, tables, forms, or layouts; `chart_qa` requires grounding in axes, legends, cells, or diagram structure; `classification` requires a visually decidable and specific label;

`multi_image` requires comparison or integration across images; `conversation` requires coherent multi-turn visual grounding; `text_only` is retained only when text-only training is intentional; and `unknown` uses the general clarity, grounding, usefulness, and safety criteria.

Capability tags. Possible tags include `object_recognition`, `action_recognition`, `text_recognition`, `spatial_recognition`, `spatial_relationship`, `object_interaction`, `scene_understanding`, `counting`, `color`, `shape`, `chart_or_table_understanding`, `document_layout`, `world_knowledge`, and `reasoning`.

JSON output. The returned object contains `task_label`, `capability_tags`, and six scored sections: `image_quality`, `text_quality`, `image_text_alignment`, `answer_quality`, `compositional_value`, and `safety`. Each section records its score, relevant validity flags, issues, and evidence. The object also contains `overall.score`, `passes_threshold`, `recommended_action` (`keep`, `revise`, `drop`, or `manual_review`), `risk_level`, `failure_reasons`, and `revision_suggestions`.

Runtime fields. The final prompt appends `Threshold: {threshold}/5.0`, `Rubric: {rubric}`, `Sample: {snapshot}`, `Choices: {choices}`, and `Image path issues: {image_errors}`.

Let $q_{i,d}$ denote the score of sample i on dimension d. A valid `overall.score` is used directly. Otherwise, the fallback score is

$$\widetilde{s}_i = 0.15q_{i,\text{img}} + 0.18q_{i,\text{text}} + 0.25q_{i,\text{align}} + 0.27q_{i,\text{ans}} + 0.10q_{i,\text{comp}} + 0.05q_{i,\text{safe}}. \quad (1)$$

For strict pseudo-label construction, complete and parse-valid samples are labeled *high* when $s_i \geq 4.5$ and all four core dimensions are at least 4.0. Samples with $s_i \leq 3.0$ are labeled *low*; all others remain unlabeled. Compositional-value and safety scores are retained for auditing but are not hard gates.

A.3 Architecture Ablation of the Hidden-State Quality Scorer

We examine whether the architecture of the sequence-processing head affects the utility of the selected data. We randomly sample a fixed candidate pool of 10K examples from LLaVA-665K and keep the frozen Qwen3.5-9B backbone, image-conditioned hidden states, evaluator-derived strict labels, candidate pool, 1K selection budget, and downstream fine-tuning configuration unchanged. Only the scoring head is varied. The default Stage-II CNN uses multi-scale one-dimensional convolutions with kernel sizes of 3, 4, and 5. The MLP applies mean and maximum pooling followed by a 512-dimensional hidden layer. The Transformer uses two encoder layers, eight attention heads, a 512-dimensional feed-forward layer, and a maximum sequence length of 128 tokens. Each scorer ranks the pool by its predicted high-quality probability and selects the top 1K examples. Full-10K uses the complete candidate pool without quality-based selection and serves as the relative-performance reference.

Table 4: Architecture ablation of the hidden-state quality scorer on Qwen3.5-9B. All values except MME are percentages; MME reports the raw total score.

Method	CV-Bench	InfoVQA	MME	MMStar	SeedB2+	TextVQA	LLaVA-W	MM-Vet	Rel.
Stage-II CNN-1K	**87.45**	85.85	**2443.56**	**71.26**	72.16	83.43	**79.30**	69.13	**100.78**
MLP-1K	87.38	86.45	2429.64	70.66	72.42	83.64	75.50	**69.54**	100.20
Transformer-1K	87.23	85.44	2421.13	70.71	72.33	83.78	77.80	68.07	100.11
Full-10K	87.41	**86.77**	2430.56	69.02	**73.30**	**85.01**	76.40	67.06	100.00

Table 4 presents the downstream performance of the different scorer architectures under the same candidate pool and 1K selection budget. **Overall, the Stage-II CNN scorer achieves the highest aggregate relative performance, indicating that the multi-scale convolutional head provides the most favorable cross-task trade-off in this controlled comparison.** Specifically, (a) Stage-II CNN-1K reaches 100.78%, exceeding the MLP and Transformer variants by 0.58 and 0.67 percentage points, respectively, and outperforming Full-10K by 0.78 percentage points; it also obtains

the best CV-Bench, MME, MMStar, and LLaVA-W results. (b) The alternative heads exhibit complementary task-specific behavior: among the three learned scorers, the MLP performs best on InfoVQA, SeedBench2+, and MM-Vet, whereas the Transformer performs best on TextVQA; Full-10K remains strongest on InfoVQA, SeedBench2+, and TextVQA when the unfiltered reference is included. (c) All three learned scorers outperform Full-10K in Rel. while using only one-tenth as many training examples, showing that the selection benefit is not exclusive to one head, although the CNN yields the best aggregate result. Because each architecture is represented by a single run, the differences constitute descriptive rather than statistical evidence.

A.4 Additional General-Benchmark Evaluation

All models are trained from the same Qwen3.5-9B backbone following the training protocol used in the main experiments. Each 65K data-selection method uses exactly 65,536 training samples, whereas Full uses the complete 665K training set. We use `lmms-eval` with zero-shot prompting and the same inference configuration for every model. Performance is measured using MMLU accuracy and HellaSwag normalized accuracy, and Avg. is their arithmetic mean.

Table 5: Comparison of data-selection methods on Qwen3.5-9B using MMLU and HellaSwag. All values are percentages.

Size	Algorithm	MMLU	HellaSwag	Avg.
65K	Random	78.30	78.54	78.42
65K	Perplexity	78.39	78.19	78.29
65K	SemDeDup	77.49	78.47	77.98
65K	D2-Pruning	78.13	78.36	78.25
65K	Self-Sup (Prototype)	78.17	78.28	78.23
65K	Self-Filter	77.65	78.27	77.96
65K	ICONS	78.05	78.48	78.27
65K	COMPACT	77.80	78.18	77.99
65K	**Ours**	**78.65**	**79.19**	**78.92**
665K	Full	77.65	78.24	77.95

Table 5 presents the general-benchmark performance of the data-selection methods on MMLU and HellaSwag. **Overall, Ours achieves the best performance on both benchmarks and the highest average score, demonstrating the strongest general-benchmark transfer among the evaluated methods under the same 65K data budget.** Specifically, (a) Ours obtains 78.65 on MMLU, 79.19 on HellaSwag, and an average of 78.92, ranking first in all three columns. (b) Relative to the strongest competing 65K result in each column, Ours improves MMLU by 0.26 percentage points over Perplexity, HellaSwag by 0.65 points over Random, and Avg. by 0.50 points over Random. (c) Ours also exceeds Full by 1.00, 0.95, and 0.97 percentage points on MMLU, HellaSwag, and Avg., respectively, while using approximately one-tenth of the training samples, indicating a favorable performance–data trade-off in this evaluation.

A.5 Effect of Hidden-State Source

We compare Qwen3.5-9B and Qwen3-VL-4B-Instruct as the frozen models used to extract hidden states for data selection. Both scorers select 65K examples and use the same downstream training configuration. The 9B and 4B columns report the relative performance of each selected subset on the corresponding downstream backbone. Avg. Rel. is the macro-average across the two backbones, computed from unrounded values.

Table 6: Effect of the hidden-state source on cross-backbone selection performance. Each column reports relative performance for a fixed downstream backbone. Avg. Rel. is the macro-average across the two downstream backbones.

Hidden-state source	Qwen3.5-9B Rel.	Qwen3-VL-4B Rel.	Avg. Rel.
Qwen3.5-9B states	**102.29**	102.12	102.21
Qwen3-VL-4B states	102.23	**102.23**	**102.23**

Table 6 compares the transferability of hidden-state quality features across representation sources and downstream backbones. Both sources achieve similar performance, with all pairwise differences within 0.11 percentage points. Matching the hidden-state source to the downstream backbone yields slightly higher point estimates: 102.29 versus 102.23 for Qwen3.5-9B, and 102.23 versus 102.12 for Qwen3-VL-4B-Instruct. However. Overall, the results indicate that the learned quality signal transfers effectively across the two MLLMS backbones, while source matching may provide a marginal benefit under the evaluated settings.