Your Vision-Language Model Itself Is a Strong Filter: Towards High-Quality Instruction Tuning with Data Selection

Anonymous ACL submission

Abstract

Data selection in instruction tuning emerges as a pivotal process for acquiring high-quality data and training instruction-following large language models (LLMs), but it is still a new and unexplored research area for vision-language models (VLMs). Existing data selection approaches on LLMs either rely on single unreliable scores, or use downstream tasks for selection, which is time-consuming and can lead to potential over-fitting on the chosen evaluation datasets. To address this challenge, we introduce a novel dataset selection method, SELF-FILTER, that utilizes the VLM itself as a filter. This approach is inspired by the observation that VLMs benefit from training with the most challenging instructions. SELF-FILTER operates in two stages. In the first stage, we devise a scoring network to evaluate the difficulty of training instructions, which is co-trained with the VLM. In the second stage, we use the trained score net to measure the difficulty of each instruction, select the most challenging samples, and penalize similar samples to encourage diversity. Comprehensive experiments on LLaVA and MiniGPT-4 show that SELF-FILTER can reach better results compared to full data settings with merely about 15% samples, and can achieve superior performance against competitive baselines.

1 Introduction

Instruction-following foundation models, such as Gemini (Anil et al., 2023), GPT-4 and GPT-4 Vision (OpenAI, 2023), have shown exceptional performance in multi-modality tasks. Their capabilities are primarily derived from the Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022) paradigm and the Instruction Fine-Tuning (IFT) (Longpre et al., 2023) framework. LIMA (Zhou et al., 2023) indicates that while foundation models primarily acquire their knowledge during pre-training phases, only a small amount of instruction is needed for fine-tuning these models to produce the desired outputs during interactions. However, the low-quality instructions can significantly degrade the models' performance. To address this challenge, dataset selection tasks have been proposed, which aim to select high-quality instruction-tuning data to enhance the performance of instruction-following foundation models.

Existing data selection methods have many limitations in selecting high-quality instruction-tuning data. Traditional data pruning metrics (Toneva et al., 2019; Meding et al., 2022; Yang et al., 2023) usually use the influence on model accuracy to filter samples, *i.e.*, dependent on the classification tasks, and are unsuitable for the generative models. Zhou et al. (2023) manually crafted high-quality instruction dataset, which is costly and time-consuming. Chen et al. (2023) query ChatGPT to rank instructions, but the performance is unstable and there is no information from images in the multimodal setting. Other works (Cao et al., 2023; Li et al., 2023d) use pre-defined evaluation datasets to construct metrics, thus highly relying on the quality and the distribution of the selected evaluation datasets, which can potentially cause over-fitting problems and harm the models' generalization ability. These limitations prevent existing data selection approaches from being applied to VLMs. To our best knowledge, InstructionGPT-4 (Wei et al., 2023) is the only dataset selection method focusing on VLMs. They also introduce additional tasks to train a data selector, and their experiments are only limited to MiniGPT-4 on a small dataset with 3.4k samples.

To address the aforementioned issues, we propose SELF-FILTER by leveraging VLM itself as an effective data filter. Specifically, We first generate score embeddings for each training instruction through feature extractors, such as CLIP (Radford et al., 2021) and GPT-4 Vision, with the training instruction itself as inputs. After obtaining score

embeddings, we use them to train a score net together with the target VLM. The score net takes the score embeddings as input and produces the weight of corresponding training instructions in VLM training losses. We expect that these weights can indicate the "difficulty" of the samples to be learned by VLMs during instruction tuning. Then we choose the instruction data with the highest difficulty to construct the filtered dataset. Moreover, we introduce a penalty mechanism on similar training samples to enhance the diversity of instructions. Compared with InstructionGPT-4 (Wei et al., 2023), our method can easily transfer to any vision-language model with large-scale datasets.

Through comprehensive experiments on the widely studied vision-language models (*e.g.,* LLaVA and MiniGPT-4), we show that with only around 15% samples of the raw instruction tuning dataset, SELF-FILTER can surpass the model trained on the original instruction data across various evaluation datasets and benchmarks. SELF-FILTER also outperforms several competitive baselines and achieves state-of-the-art performance.

Our contributions are summarized as follows:

- We propose a novel method named SELF-FILTER, and demonstrate that large vision-language models themselves can serve as filters for instruction-finetuning. We show the efficacy of our method through extensive experiments.
- To the best of our knowledge, we are the first to show that on large-scale instruction tuning datasets, vision-language models do not necessarily require a large number of data. A small amount of high-quality data is sufficient for successful instruction tuning.
- Our method does not require additional predefined evaluation tasks or surrogate models. Importantly, it makes no assumptions about downstream tasks, thereby preserving the model's generalization capabilities—a critical factor for addressing complex real-world applications.

2 Related Work

2.1 Data Selection

Data selection is an emerging topic in large-language model instruction-tuning (Chen et al., 2023; Cao et al., 2023), aiming at selecting high-quality data and discarding harmful data, which could cause hallucinations. Alpagasus (Chen et al., 2023) is the pioneer of automatic filtering. They query ChatGPT for instruction quality, improving the efficiency of training. Li et al. (2023b) proposes using the IFD score as a proxy of the sample difficulty. Instruction Mining (Cao et al., 2023) uses a linear combination of several indicators to judge the sample quality. Concurrent work by Li et al. (2023d) proposes to use one-shot learning performance on pre-defined tasks to filter data.

InstructionGPT-4 (Wei et al., 2023) is the only work in VLM instruction tuning. They utilize the combination of multimodal scores as the indicators and rely on a regression model trained on predefined tasks to conduct data selection. Their application is also limited to MiniGPT-4 with only 3.4k instructions.

2.2 Instruction Finetuning

Instruction tuning is an essential step for pre-trained foundation models to obtain instruction-following capabilities, and deal with complex human queries. Alpaca (Taori et al., 2023), Vicuna (Chiang et al., 2023), WizardLM (Xu et al., 2023a) distill GPT-family models and release high-quality instruction-tuning datasets and build up the easy-to-use tools for instruction-finetuning. Following them, MiniGPT-4 (Zhu et al., 2023) and LLaVA (Liu et al., 2023b) first utilize instruction tuning in multimodal models. They adopt the same pipeline, designing templates to distill GPT-family models and obtain data. After that, more vision-language models like InstructBlip (Dai et al., 2023) and Qwen-VL-Chat (Bai et al., 2023) also exploit the instruction tuning procedure to gain chat ability.

A detailed related work discussion can be found in Appendix A.

3 Methodology

In this section, we briefly review the related background knowledge of vision-language models and elaborate on the task formulation. Then we thoroughly introduce our proposed SELF-FILTER.

3.1 Preliminary

Instruction-Following Vision-Language Models. The training process of current instruction-following vision-language models can roughly be divided into two parts, (1) *pre-training*, (2) *instruction fine-tuning (IFT)*, or *reinforcement learning from human feedback (RLHF)*. During pre-training,

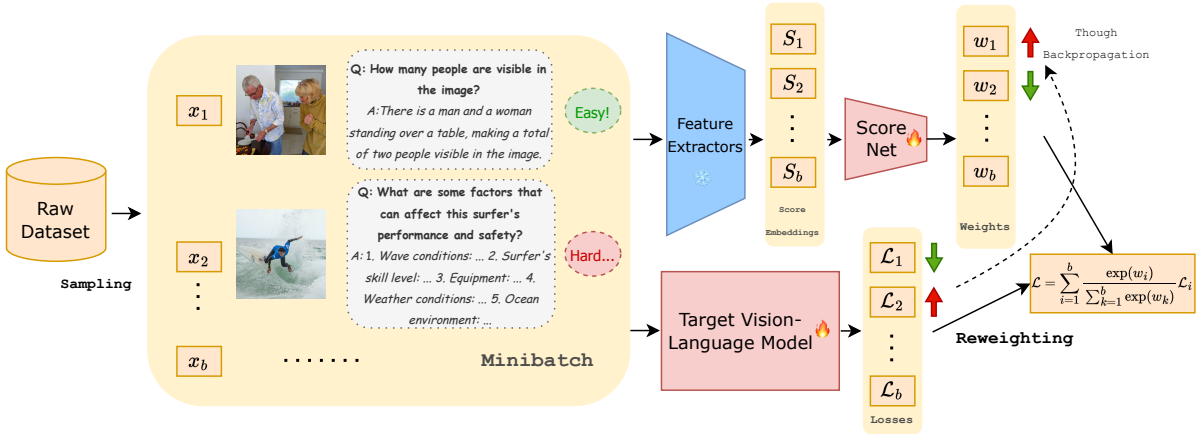


Figure 1: Illustration of the stage 1 of SELF-FILTER. In the target vision-language models, samples that are more challenging usually yield higher losses. During the training process, the loss of each instruction is combined with a learnable weight generated by the score net. Through minimizing this weighted loss function $\mathcal{L}$, the instructions with higher loss tend to have lower weights. It is important to note that in stage 1, both the target VLM and the score net are actively trained, with the pre-trained feature extractors remaining frozen. Following this, stage 2 involves employing the score net to estimate the weights for each instruction, in which a lower weight signifies a higher difficulty level for the model.

the models may fine-tune their image and text backbones and align their representations. For IFT and RLHF, the models will gain instruction-following ability to output high-quality responses in conversations.

Task Formulation. We present the definition of our data selection task in instruction-finetuning. Given the instruction tuning dataset $\mathcal{D} = \{\boldsymbol{x}_j\}_{j=1}^N$, where $\boldsymbol{x}_j = (\boldsymbol{x}_j^i, \boldsymbol{x}_j^t)$ denotes the input image and text pair, our task is to select a subset and to prune $\mathcal{D}$ to any desired size m, so that the vision-language model f can achieve the best performance on downstream tasks $\{T_i\}_{i=1}^t$ after training on the selected subset $\mathcal{D}_f^m \subset \mathcal{D}$, $|\mathcal{D}_f^m| = m$. Here the VLM f is already pre-trained. We present the total VLM loss as $\mathcal{L}$, and the VLM loss on a single instruction $\boldsymbol{x}_i$ as $\mathcal{L}_i$.

Notably, we could not have access to the distribution of downstream tasks in real-world applications, so we should preserve the models' generalization ability while conducting data selection.

3.2 SELF-FILTER

Motivation. The goal of data selection tasks is to find the instructions with the highest quality and diversity. In assessing the quality of instruction data, prior research (Xu et al., 2023a; Zhao et al., 2023; Cao et al., 2023) has indicated that the most challenging samples, or "hardest" samples, can provide significant insights and are therefore valuable for model training. These "hardest" samples are defined as those that present substantial learning challenges to VLMs during the training process. Motivated by this observation, we aim to develop an evaluation metric capable of quantifying the learning difficulty d_i (with VLMs) of each sample $\boldsymbol{x}_i$ to accurately reflect its quality.

While many metric-based data selection methods (Chen et al., 2023; Cao et al., 2023; Liu et al., 2023c) have been applied to large language models (LLMs), directly transferring them to more complex multimodal contexts proves challenging, often yielding suboptimal outcomes as evidenced by our experiments. Given the difficulty of evaluating multimodal instructions with only a singular metric, we propose to combine these features via a lightweight scoring network s, which can be either a Multi-Layer Perceptron (MLP) or a linear layer.

To increase the diversity within the filtered instruction dataset, we introduce a penalty mechanism targeting similar examples. Specifically, when a sample is chosen for the filtered dataset, we reduce the likelihood of selecting its k-nearest neighbors from the raw dataset. This approach is designed to ensure a broader variety of instructional content by actively reducing redundancy among the chosen samples.

In the rest part of this section, we introduce SELF-FILTER, which consists of two stages: a) score net training, and b) data selecting by quality and diversity.

Stage 1: Training the Score Net. Paul et al. (2021) identifies loss as an important indicator for measuring the difficulty level of training instructions, with more challenging instructions typically exhibiting higher loss. Motivated by data

Algorithm 1 Stage 1: Train the score net

Require: Raw multimodal instruction dataset $\mathcal{D}$, target vision-language model f, score net s and its parameters ϕ, batch size b, feature extractors M

for each step **do**
  Sample a minibatch $B = \{\boldsymbol{x}_i\}_{i=1}^{b} \sim \mathcal{D}$
  for $\boldsymbol{x}_i$ in B **do**
    Calculate the VLM loss $\mathcal{L}_i$ of $\boldsymbol{x}_i$
    $\boldsymbol{S}_i \leftarrow M(\boldsymbol{x}_i)$
    $w_i \leftarrow s(\boldsymbol{S}_i, \phi)$
  $w_i' \leftarrow \frac{\exp(w_i)}{\sum_{j=1}^{b} \exp(w_j)} b$
  $\mathcal{L} \leftarrow \frac{1}{b} \sum_{i=1}^{b} w_i' \mathcal{L}_i$
  Use $\mathcal{L}$ to backpropagate and optimize the vision-language model and the score net

re-weighting approaches (Gao et al., 2023; Zhang and Pfister, 2021), we propose utilizing learnable weights, produced by a score net, within the VLM loss framework to assess the difficulty of training instructions. Given that more challenging instructions typically incur higher loss, their corresponding learnable weights are inclined to decrease during the training process. Consequently, we can assess and rank the difficulty of instructions based on their learnable weights. The illustration of stage 1 of SELF-FILTER is shown in Figure 1.

The initial phase in training the score net involves preparing the input. This is accomplished by extracting features from the training instructions using various extractors, such as CLIP and GPT-4V. Let $M = \{M_i\}_{i=1}^{n}$ denote the series of feature extractors. We combine the output features from these extractors to form the input of the score net $\boldsymbol{S}_i \in \mathcal{S}$:

$$\boldsymbol{S}_i = M(\boldsymbol{x}_i) = [M_1(\boldsymbol{x}_i), \cdots, M_n(\boldsymbol{x}_i)] \quad (1)$$

After preparing the input, we use a score net $s(\cdot, \phi) : \mathcal{S} \rightarrow \mathbb{R}$, where ϕ represents the parameters of the score net, to project the generated input embeddings to the weight of the corresponding instructions in the learning objective of VLMs. Given a sample $\boldsymbol{x}_i$ and corresponding input embedding $\boldsymbol{S}_i$, the weight of $\boldsymbol{x}_i$ in the VLM loss is given by $w_i = s(\boldsymbol{S}_i, \phi)$. Denoted by $\mathcal{L}_i$ the original VLM loss on $\boldsymbol{x}_i$, the new loss after reweighting is given by $w_i \mathcal{L}_i$.

It is important to note that the weights generated by the score net for different instructions are independent and may vary significantly in scale, rendering them inadequate for cross-instruction difficulty ranking. To overcome this issue, we introduce an inter-batch weight normalization technique that applies the softmax function to the weights within the sampled batch. This results in a normalized weight, denoted as $w_i' = \frac{\exp(w_i)}{\sum_{k=1}^{b} \exp(w_k)} b$, where b is the batch size. Our learning objective is subsequently formulated as follows:

$$\mathcal{L} = \sum_{i=1}^{b} \frac{\exp(w_i)}{\sum_{k=1}^{b} \exp(w_k)} \mathcal{L}_i \quad (2)$$

Through combining the weights with the VLM loss, the score net can receive guidance along with the training process of the VLM itself, avoiding the introduction of possibly unreliable surrogate models or additional evaluation datasets, and preserving the generalization ability of our target VLM. The detailed algorithm of stage 1 is shown in Algorithm 1.

In our experiments, we observe that utilizing a limited set of pre-calculated scores as feature extractors M, *e.g.*, CLIP Score (Hessel et al., 2021) and GPT Score (Chen et al., 2023), yields satisfactory results. However, for larger data volumes, more comprehensive features like CLIP-encoded features tend to outperform, as they contain richer information. This enables the score net s to more effectively find the relationship between coarse features and instruction quality through larger scale training.

Algorithm 2 Stage2: Filter the instructions

Require: Raw multimodal instruction finetuning dataset $\mathcal{D}$, target pruning size m, k-nearest neighbor algorithm *KNN*, similarity metric *Sim*

$\mathcal{D}_f^m \leftarrow \{\}$
for $\boldsymbol{x}_i$ in $\mathcal{D}$ **do**
  $\boldsymbol{S}_i \leftarrow M(\boldsymbol{x}_i)$
  $w_i \leftarrow s(\boldsymbol{S}_i, \phi)$
  $d_i \leftarrow -w_i$
for m iterations **do**
  $i \leftarrow \text{argmax}_{i, \boldsymbol{x}_i \in \mathcal{D}} \; d_i$
  $\mathcal{D}_f^m \leftarrow \mathcal{D}_f^m \cup \{\boldsymbol{x}_i\}$
  $\mathcal{D} \leftarrow \mathcal{D} - \{\boldsymbol{x}_i\}$
  for $\boldsymbol{x}_j$ in $\text{KNN}(\boldsymbol{x}_i)$ **do**
    $d_j \leftarrow d_j - \gamma \text{Sim}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 d_i$
Output $\mathcal{D}_f^m$

Stage 2: Filtering the Instructions Considering Quality and Diversity. After obtaining the

trained score net s in stage 1, we proceed to compute the final weight w_i for each sample. Intuitively, more challenging multimodal instructions produce larger losses during training, which in turn, through backpropagation, results in a smaller weight w_i. Therefore, we define our difficulty metric as $d_i = -w_i$.

We do not directly use the training loss to rank instructions due to the observation that newly trained samples typically exhibit smaller losses. This could lead to biased ranking and, as demonstrated in our experiments (see Section 4.3), suboptimal performance.

Moreover, diversity has been recognized as a critical factor for effective data selection (Wei et al., 2023; Maharana et al., 2023). In our approach, we enhance diversity by penalizing the k-nearest neighbors of a newly selected sample, explicitly lowering their difficulty scores to encourage a broader variety of selections. After selecting an instruction $\boldsymbol{x}_i$ to the filtered dataset, we adjust the difficulty of its k-nearest neighbors examples by:

$$d_j = d_j - \gamma \text{Sim}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 d_i \,, \tag{3}$$

where $\boldsymbol{x}_j$ is one of the k-nearest neighbors of $\boldsymbol{x}_i$, and $\text{Sim}(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbb{R}$ represents their similarity. γ is a hyperparameter and is empirically set to 1. The whole process of stage 2 is described in Algorithm 2.

4 Experiments

In this section, we first detail our settings and the chosen hyperparameters. Then we introduce the evaluation benchmarks and datasets used in our experiments and the baseline methods. Finally, we show that our proposed SELF-FILTER can get state-of-the-art performance through extensive experiments.

4.1 Settings and Hyperparameters

We evaluate our approach on two extensively studied vision-language models: LLaVA (Liu et al., 2023b) and MiniGPT-4 (Zhu et al., 2023). These models employ vision encoders and large language models as backbones, and they design different pre-training tasks to train projection modules for aligning multimodal representations. Then they employ automatically generated multimodal instruction datasets through instruction-finetuning. Further information regarding the specific versions of the models is available in Appendix B.

The instruction-tuning dataset in LLaVA comprises 157,712 training samples, whereas MiniGPT-4 has 3,439 samples. Our approach adheres strictly to the original instruction-tuning settings and hyperparameters as outlined in their official implementations. We use 8 A6000 GPUs in our experiments.

For the selection of feature extractors M, in MiniGPT-4, we incorporate CLIP Score (Hessel et al., 2021), Imagereward (Xu et al., 2023b), ChatGPT, and GPT-4Vision. In LLaVA, the chosen feature extractors include CLIP Score, Imagereward, and ChatGPT (referred to as *Scores* in Table 1). Additionally, we experiment with using the features directly generated by the CLIP encoders (referred to as *CLIP* in Table 1). Comprehensive details about all the feature extractors and their utilization can be found in Appendix C.

For the score net, we implement it as a linear layer. We set the value of k to 10 for the KNN algorithm in the diversity module and employ cosine similarity between CLIP features to measure the sample similarity, as described in Eq. 3.

4.2 Evaluations

Benchmarks and Datasets We employ two evaluation tools, LVLM-eHub (Xu et al., 2023c) and VLMEvalKit (Contributors, 2023), to conduct a comprehensive assessment of our method. Our evaluation encompasses a diverse array of benchmarks and datasets, including MMBench (Liu et al., 2023d), MME (Fu et al., 2023), SEED-Bench (Li et al., 2023a), HallusionBench (Liu et al., 2023a), MathVista (Lu et al., 2023), ScienceQA (Lu et al., 2022), OK-VQA (Marino et al., 2019), TextVQA (Singh et al., 2019), VisDial (Das et al., 2017), VCR (Zellers et al., 2019), MSCOCO (Lin et al., 2014), and Pope (Li et al., 2023c). The interpretation of their evaluation metrics is detailed in Appendix D.

Baselines. We compare the performance of SELF-FILTER with random filtering and four other competitive baselines:

- **EL2N:** (Paul et al., 2021) Initially introduced for classification tasks, EL2N (Error L2-Norm) score is defined as $\mathbb{E}\|p(\boldsymbol{w}, \boldsymbol{x}) - \boldsymbol{y}\|_2$, where $\boldsymbol{w}$ represents the parameters, $\boldsymbol{x}$ denotes the input features, and $\boldsymbol{y} \in \{0,1\}^K$ is the ground truth label in the form of a one-hot vector. To align with the generative model, we modify the score as the form of the average cross-entropy loss for a single instruction.

Table 1: Main results for SELF-FILTER on LLaVA (Liu et al., 2023b). *Scores* represents using CLIP Score, Imagereward and ChatGPT as feature extractors, and *CLIP* presents using CLIP features. We also try removing the diversity module under the *Scores* setting. The meanings of the numbers are detailed in Appendix D.

		Baseline						SELF-FILTER		
		Full data	**Random**	**GraNd**	**EL2N**	**Prototy-picality**	**Alpaga-sus**	**Scores**	**w/o Diversity**	**CLIP**
Samples		158k	25k	25k	25k	25k	25k	25k	25k	25k
MMBench		23.97	30.67	28.26	30.41	29.64	34.71	30.58	24.66	**38.48**
MME	**Overall**	1132.89	1093.15	1138.49	969.48	1165.69	1096.05	1126.24	1079.9	**1218.15**
	Perception	884.68	858.15	884.49	723.77	904.98	827.84	850.89	804.19	**955.65**
	Reasoning	248.21	235.00	254.00	245.71	260.71	268.21	275.35	**275.71**	262.50
SEED-Bench		39.96	44.38	45.15	43.86	46.80	**47.67**	45.34	40.94	47.54
Hallusion Bench	**aACC**	45.32	43.85	42.27	41.22	45.32	42.69	**48.15**	45.53	45.85
	fACC	**14.74**	14.16	9.83	9.25	13.29	10.12	11.54	10.98	14.45
	qACC	10.33	9.01	8.57	7.25	10.77	6.81	**11.11**	10.99	10.33
MathVista		25.10	24.10	24.60	24.80	25.40	23.90	25.40	**26.70**	26.40
ScienceQA		55.03	56.12	52.79	58.01	56.87	58.75	56.22	55.83	**59.40**

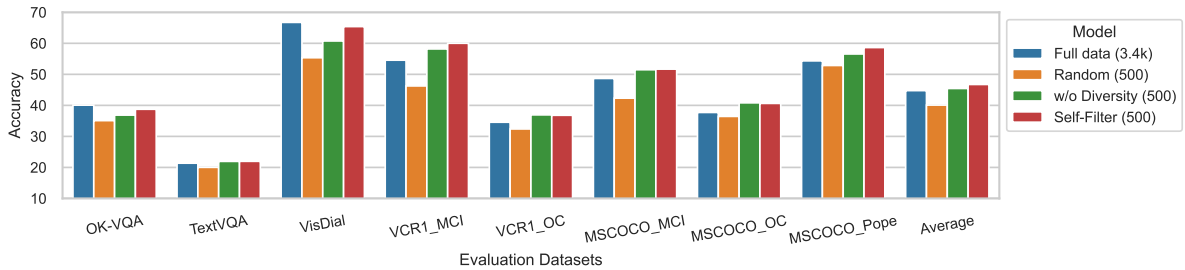


Figure 2: Main results on MiniGPT-4 (Zhu et al., 2023). *w/o Diversity* represents removing the diversity module in our method. SELF-FILTER achieves the best results on most tasks, and has a higher average accuracy.

- **GraNd:** (Paul et al., 2021) GraNd (Gradient Normed) score is defined as $\mathbb{E}_{\boldsymbol{w}}\|g(\boldsymbol{x}, \boldsymbol{y})\|_2$, which utilizes the norm of gradients caused by each sample to quantify their influence. A higher GraNd score suggests that the corresponding sample is more important.

- **Prototypicality:** (Sorscher et al., 2022) The prototypicality metric involves performing k-means clustering in the embedding space. The difficulty of each data point is determined by the Euclidean distance to its nearest cluster centroid, *i.e.*, the prototype. In our experiments, we utilize the last hidden layer feature as the embedding.

- **Alpagasus:** (Chen et al., 2023) Alpagasus queries ChatGPT to provide evaluation scores and detailed explanations for each data point. Then these scores are utilized for ranking and filtering instructions.

Our method, along with all the aforementioned baselines, does not require additional datasets or surrogate models, ensuring a fair comparison.

4.3 Experimental Results

MiniGPT-4 Results. We evaluate MiniGPT-4 with LVLM-eHub, and the results are presented in Figure 2. With only 500 samples (14.5% of the dataset), SELF-FILTER achieves the best results in 6 out of 8 tasks, significantly outperforming the random baseline. Additionally, our findings demonstrate that the diversity module contributes to enhancing the selection process. The specific numerical results for MiniGPT-4 can be found in Appendix F.

LLaVA Results. Our main experimental results on LLaVA using VLMEvalKit are displayed in Table 1. We train the model with 25k samples (15.9%) selected from the original dataset of 158k instances, resulting in substantial improvements compared to the full data settings. While several competitive baseline methods can also surpass the full dataset performance, our approach leads to higher results.

Furthermore, we observe that using CLIP features (referred to as the *CLIP* setting) leads to further performance improvement. This finding

Table 2: The easiest sample and hardest sample produced by SELF-FILTER.

Easiest Sample	Text	Image
[Human]: **[GPT]:** **[Human]:** **[GPT]:** **[Human]:** **[GPT]:**	How many people are visible in the image? There is a man and a woman standing over a table, making a total of two people visible in the image. What is the man doing in the image? In the image, the man is cutting into a chocolate frosted cake. Are there any other elements on the table besides the cake? Yes, there are orange flowers on the table in addition to the cake.	
Hardest Sample	**Text**	**Image**
[Human]: **[GPT]:**	What precautions should someone take when planning to surf in rough waters? When planning to surf in rough waters, like the man wearing a black wetsuit in the image, it is essential to take several precautions to ensure safety. These precautions include: 1. Wearing appropriate gear: A wetsuit provides buoyancy, warmth, and protection from any potential hazards in the water, such as rocks. 2. Checking weather and water conditions: Understanding the current ocean conditions and the forecast will help surfers anticipate how rough the waves might be and decide if it's safe to surf. 3. Being aware of rip currents: These strong currents can carry surfers away from the shore, so it's crucial to understand how to spot and escape them. 4. Surfing with a buddy: Having a surfing partner ensures there's someone to help or alert others in case of an emergency. 5. Knowing personal limits: Understanding one's skill level and not attempting to surf waves that are beyond that level is essential for safety. 6. Proper warm-up and stretching: This can help prevent injuries while surfing in challenging conditions. 7. Communicating with local surfers: They can offer insights into the specific beach break and advise on potential hazards and safe spots. By taking these precautions, surfers can significantly reduce the risks associated with surfing in rough waters and enjoy the exhilaration and challenges that such conditions offer.	

Table 3: Results for selecting the hardest samples and the easiest samples. 25k samples are used in training. Choosing the easiest samples found by SELF-FILTER can greatly hurt the performance.

		Hardest (Ours)	Easiest
MMBench		**38.48**	24.14
MME	**Overall**	**1218.15**	962.36
	Perception	**955.65**	713.43
	Reasoning	**262.5**	248.93
SEED-Bench		**47.54**	43.18
Hallusion Bench	**aACC**	**48.15**	43.01
	fACC	**14.45**	12.14
	aACC	**10.33**	8.13
MathVista		**26.4**	25.1
ScienceQA		**59.4**	55.88

suggests that the score net can effectively extract information from coarse features. In contrast, pre-calculated scores such as CLIP Score and ChatGPT outputs may be prone to inaccuracies and information losses.

5 Analysis

In this section, we first show the importance of the diversity module and choosing harder samples through ablation studies. Then we show that our method can indeed select the difficult samples in the case study. Finally, we analyze the influence of pruning size and batch size.

5.1 Ablation Study

As Table 1 and Figure 2 show, removing the diversity module and not imposing penalties on similar samples may maintain similar performance on certain datasets, but lead to degradation on others, such as VisDial for MiniGPT-4 and MMBench for LLaVA.

This underscores the importance of ensuring diversity during data selection. Given the potential variation in data quality across different topics, a sole emphasis on quality might result in an unbalanced distribution of topics.

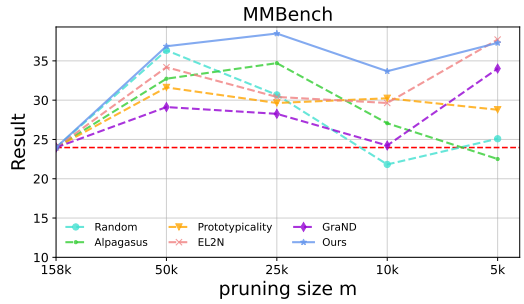
5.2 Why Choose the Hardest Samples?

We try to select the easiest samples as the filtered dataset in LLaVA under the *CLIP* setting, and the results are presented in Table 3. The evaluation scores experience a significant drop compared to the hardest policy, reaffirming the observations made by Xu et al. (2023a) that difficult samples are more informative and effective for instruction tuning.

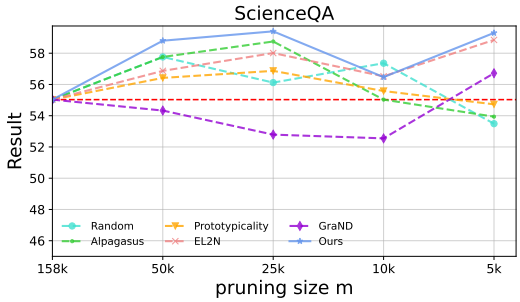
5.3 Case Study

In this section, we investigate the samples selected and abandoned by our method. Table 2 showcases the easiest and hardest samples identified by the score net from the training data in LLaVA. Additional examples can be found in Appendix E.

The discarded easy examples typically consist of uncomplicated and direct questions, such as counting the objects in an image and describing the scenario. In contrast, more challenging instances focus on intricate tasks involving reasoning and planning, often having greater length. The Pearson correla-



(a) MMBench



(b) ScienceQA

Figure 3: Experimental results under difference target pruning size m.

tion between the difficulty score and the sample text length is -0.6325, reflecting the significance of length as a factor but also emphasizing that it cannot be the sole determinant of difficulty. In summary, our approach adeptly and precisely identifies challenging samples for instruction tuning.

Additionally, it is evident that the distribution of topics among difficulty levels is uneven, with the top 5 most challenging samples all talking about surfing. Consequently, it is necessary to design the diversity module to ensure a more balanced representation across various subjects.

5.4 Performance under Different Pruning Size

We conducted experiments with different target pruning sizes, and the results are illustrated in Figure 3. We use the *CLIP* setting in LLaVA, and our method consistently outperforms others in the majority of cases. As the pruning size m decreases, the evaluation results initially show improvement by discarding low-quality data, reaching a peak at 25k samples. Subsequently, the results gradually decline, emphasizing the prominence of quantity as a determining factor. It's noteworthy that limited data significantly impacts data diversity.

In traditional classification tasks, Paul et al. (2021) observed a substantial performance decline with approximately 30%-50% data. In contrast, in the instruction tuning task, a mere 5k samples (3.17%) still yield reasonable results, implying that little knowledge is learned during this stage.

Table 4: Results for different batch size b.

Batch Size		16	8	4
MMBench		38.48	37.46	**43.13**
MME	Overall	**1218.15**	1178.02	1179.24
	Perception	**955.65**	930.52	924.95
	Reasoning	**262.5**	247.5	254.29
SEEDBench		47.54	**48.33**	48.20
Hallusion Bench	aACC	**45.85**	45.22	45.01
	fACC	14.45	**15.03**	13.01
	qACC	10.33	10.11	**10.77**
MathVista		**26.4**	25.5	25.6
ScienceQA		59.4	58.55	**60.09**

5.5 Influence of Different Batch Size

We performed experiments using varying batch sizes during the training of the score net in Algorithm 1. The local batch size b determines the number of observed samples when adjusting the weights. It's important to note that we concurrently modified the gradient accumulation steps to maintain the overall batch size for the vision-language model unchanged.

The results are presented in Table 4. In our primary experiments, we adhere to the LLaVA settings, employing a batch size of 16. Notably, the performance exhibits minimal variance across different local batch sizes, indicating the robustness of SELF-FILTER, capable of accurately extracting information even from a local window as small as 4 samples.

6 Conclusion

In this paper, we propose SELF-FILTER to select high-quality instructions from the noisy dataset. Our method leverages the vision-language model itself as a filter, and does not need any supplemental evaluation datasets, enhancing the model's generalization ability. We design a score net framework to effectively rank all the samples based on the predicted difficulty and discard low-quality instructions to help representation learning. Our comprehensive experiments on LLaVA and MiniGPT-4 demonstrate the effectiveness of our approach, and verify that large vision-language models learn most of their knowledge from the backbones and the pre-training stage, while a small amount of high-quality samples is sufficient for instruction tuning.

Limitations

Though we prove the effectiveness of SELF-FILTER on LLaVA and MiniGPT-4, we could further extend the experiments on other vision-language models. A more careful search for hyperparameters, e.g., the batch size and γ in the diversity module can generate better results.

Another limitation is that we could explore different architectures of the score net, for example, a small vision-language model. We could drop the feature extractors under this setting, and directly feed the samples into the score net to predict weights.

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *CoRR*, abs/2308.12966.

Yihan Cao, Yanbin Kang, and Lichao Sun. 2023. Instruction mining: High-quality instruction data selection for large language models. *CoRR*, abs/2307.06290.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2023. Alpagasus: Training A better alpaca with fewer data. *CoRR*, abs/2307.08701.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *CoRR*, abs/2305.06500.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1080–1089. IEEE Computer Society.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. 2023. MME: A comprehensive evaluation benchmark for multimodal large language models. *CoRR*, abs/2306.13394.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2023. Self-guided noise-free data generation for efficient zero-shot learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7514–7528. Association for Computational Linguistics.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023a. Seed-bench: Benchmarking multimodal llms with generative comprehension. *CoRR*, abs/2307.16125.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023b. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. *CoRR*, abs/2308.12032.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023c. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 292–305. Association for Computational Linguistics.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. 2023d. One shot learning as instruction data prospector for large language models. *CoRR*, abs/2312.10302.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.

Fuxiao Liu, Tianrui Guan, Zongxia Li, Lichang Chen, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2023a. Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v(ision), llava-1.5, and other multi-modality models. *CoRR*, abs/2310.14566.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *CoRR*, abs/2304.08485.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023c. What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning. *CoRR*, abs/2312.15685.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2023d. Mmbench: Is your multi-modal model an all-around player? *CoRR*, abs/2307.06281.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models. *CoRR*, abs/2310.02255.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *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*.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. 2023. D2 pruning: Message passing for balancing diversity and difficulty in data pruning. *CoRR*, abs/2310.07931.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. OK-VQA: A visual question answering benchmark requiring external knowledge. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3195–3204. Computer Vision Foundation / IEEE.

Kristof Meding, Luca M. Schulze Buschoff, Robert Geirhos, and Felix A. Wichmann. 2022. Trivial or impossible — dichotomous data difficulty masks model differences (on imagenet and beyond). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

OpenAI. 2023. Gpt-4 technical report.

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. 2022. Training language models to follow instructions with human feedback. In *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*.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20596–20607.

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. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24*

July 2021, Virtual Event, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. 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*, pages 8317–8326. Computer Vision Foundation / IEEE.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. In *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*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Lai Wei, Zihao Jiang, Weiran Huang, and Lichao Sun. 2023. Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4. *CoRR*, abs/2308.12067.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions. *CoRR*, abs/2304.12244.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. 2023b. Imagereward: Learning and evaluating human preferences for text-to-image generation. *CoRR*, abs/2304.05977.

Peng Xu, Wenqi Shao, Kaipeng Zhang, Peng Gao, Shuo Liu, Meng Lei, Fanqing Meng, Siyuan Huang, Yu Qiao, and Ping Luo. 2023c. Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models. *CoRR*, abs/2306.09265.

Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. 2023. Dataset pruning: Reducing training data by examining generalization influence. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6720–6731. Computer Vision Foundation / IEEE.

Zizhao Zhang and Tomas Pfister. 2021. Learning fast sample re-weighting without reward data. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 705–714. IEEE.

Yingxiu Zhao, Bowen Yu, Binyuan Hui, Haiyang Yu, Fei Huang, Yongbin Li, and Nevin L. Zhang. 2023. A preliminary study of the intrinsic relationship between complexity and alignment. *CoRR*, abs/2308.05696.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: less is more for alignment. *CoRR*, abs/2305.11206.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *CoRR*, abs/2304.10592.

A Additional Related Work

A.1 Vision-Language Model

Vision-language models have various training strategies and various architectures. CLIP (Radford et al., 2021) uses contrastive learning to acquire high-quality unimodal representations. BLIP (Li et al., 2022) designs a Q-Former to bridge two modalities. Current instruction-following vision-language models, *e.g.*, LLaVA (Liu et al., 2023b), MiniGPT-4 (Zhu et al., 2023) and InstructBLIP (Dai et al., 2023) usually exploit pre-trained language models (e.g., Llama (Touvron et al., 2023), Vicuna (Chiang et al., 2023)) and vision models (e.g., ViT (Dosovitskiy et al., 2021), CLIP encoders) as backbones, and use projection modules to transform visual information into the embedding space of language models.

A.2 Data Selection

Data selection, or data pruning is always an important topic in the machine learning area. It tries to train a better model with less data, or greatly reduce the data size while do not lose too much model performance. Toneva et al. (2019) design a forgetting score to reflect whether the sample is easy to learn. Paul et al. (2021) point out that loss and gradient norm can be informative metrics for pruning. Sorscher et al. (2022) choose to use the distance from the cluster centroid to measure difficulty. Meding et al. (2022) propose to use multiple models to evaluate each image and classify it into "trivial", "impossible" or normal. Yang et al. (2023) examine the influence of the model's generalization ability when removing a sample.

Data selection is an emerging topic in large-language model instruction-tuning (Chen et al., 2023; Cao et al., 2023), aiming at selecting high-quality data and discarding harmful data, which could cause hallucinations. Alpagasus (Chen et al., 2023) is the pioneer of automatic filtering. They query ChatGPT for instruction quality, improving the efficiency of training. Li et al. (2023b) proposes using the IFD score as a proxy of the sample difficulty. Instruction Mining (Cao et al., 2023) uses a linear combination of several indicators to judge the sample quality. Concurrent work by Li et al. (2023d) proposes to use one-shot learning performance on pre-defined tasks to filter data.

A significant difference is that most previous methods (Cao et al., 2023; Wei et al., 2023; Li et al., 2023d) need additional evaluation tasks to learn the weights for different indicators, but we design a score net to automatically combine them, and our score net can utilize very coarse features like the CLIP feature which is a 1536-dimensional vector, avoiding too much information loss when calculating the indicators.

InstructionGPT-4 (Wei et al., 2023) is the only work in VLM instruction tuning. They utilize the combination of multimodal scores as the indicators and rely on a regression model trained on pre-defined tasks to conduct data selection. Their application is also limited to MiniGPT-4 with only 3.4k instructions.

A.3 Instruction Finetuning

Instruction tuning is an essential step for pre-trained foundation models to obtain instruction-following capabilities, and deal with complex human queries. Alpaca (Taori et al., 2023), Vicuna (Chiang et al., 2023), WizardLM (Xu et al., 2023a) distill GPT-family models and release high-quality instruction-tuning datasets and build up the easy-to-use tools for instruction-finetuning. Following them, MiniGPT-4 (Zhu et al., 2023) and LLaVA (Liu et al., 2023b) first utilize instruction tuning in multimodal models. They adopt the same pipeline, designing templates to distill GPT-4 and obtain data. After that, more vision-language models like InstructBlip (Dai et al., 2023) and Qwen-VL-Chat (Bai et al., 2023) also exploit the instruction tuning procedure to gain chat ability.

B Vision-Language Model Version

For LLaVA, we use the LLaVA v1.0 7B model. The language backbone is Vicuna-7B-v1.3, and the vision encoder is clip-vit-large-patch14. For MiniGPT-4, we use Vicuna V0 7B as the language model.

C Feature Extractors

C.1 GPT Input Templates

The data structures of the instruction tuning datasets for LLaVA and MiniGPT have some differences, so we design different GPT templates for them in Table 5. The templates are adapted from Chen et al. (2023).

We then use the output numbers as the scores for each instruction.

Table 5: GPT input templates.

Model		Template
MiniGPT-4	**ChatGPT**	We would like to request your feedback on the performance of AI assistant in response to the instruction and the given input displayed following. Instruction: Describe this image in detail. Response: [Caption] Please rate according to the quality of the response to the instruction. Each assistant receives a score on a scale of 0 to 10, where a higher score indicates higher level of the quality. Please first output a single line containing the value indicating the scores. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias.
	GPT-4Vision	We would like to request your feedback on the performance of AI assistant in response to the instruction and the given image above. Instruction: Describe this image in detail. Response: [Caption] Please rate according to the quality of the response and the input image. Each assistant receives a score on a scale of 0 to 10, where a higher score indicates higher level of the quality. Please first output a single line containing the value indicating the scores. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias.
LLaVA	**ChatGPT**	We would like to request your feedback on the performance of AI assistant in response to the instruction and the given input displayed following. Conversation: [Conversation] Please rate according to the quality of the response to the instruction. Each assistant receives a score on a scale of 0 to 10, where a higher score indicates higher level of the quality. Please first output a single line containing the value indicating the scores. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias.

C.2 Imagereward

Imagereward (Xu et al., 2023b) is a general-purpose text-to-image human preference reward model. We directly input the image-text pair in the instruction tuning datasets to get the output score.

C.3 CLIP Score

CLIP Score (Hessel et al., 2021) is defined as the cosine similarity between the text and image features. The model we used is *clip-vit-large-patch14*.

C.4 CLIP Features

We use the *clip-vit-large-patch14* image and text encoders to encode the input samples. The embedding size for both of them is 768, making the final concatenated CLIP feature with a dimension of 1536.

D Evaluation Benchmarks and Datasets

LVLM-eHub We use OK-VQA, TextVQA, VisDial, VCR, MSCOCO and Pope in LVLM-eHub for MiniGPT-4. They are all transformed into the visual question answering (VQA) form, and the numbers represent accuracy. Specifically, MCI represents the multi-class identification task, and OC represents the object counting task. Pope (Li et al., 2023c) is an evaluation metric for object hallucination.

VLMEvalKit. We use MMBench, MME, SEED-Bench, Hallusion Bench, Math Vista and Sci-

enceQA on VLMEvalKit for LLaVA experiments. Among them, MMBench, MME and SEED-Bench measure the general ability of VLMs, and a higher number indicates better performance. Hallusion Bench tests the VLMs' robustness against possible language hallucination and visual illusion. *aACC* is the simple average accuracy, *fACC* is the accuracy per figure, and *qACC* is the accuracy per question. More detailed explanations for those metrics can be found in Liu et al. (2023a). Math Vista measures models' math ability, while SiceneQA is the science-related question-answering dataset. Their numbers are the accuracy.

E Additional Case study

In this section, we display the top 5 easiest and hardest samples found by SELF-FILTER in Table 6 and Table 7

F MiniGPT4 Experimental Results

We provide the exact numbers for the main experimental results on MiniGPT-4 in Table 8.

Table 6: Top 5 Easiest samples.

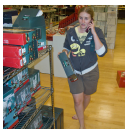
1st Easiest Sample	Text	Image
[Human]:	How many people are visible in the image?	
[GPT]:	There is a man and a woman standing over a table, making a total of two people visible in the image.	
[Human]:	What is the man doing in the image?	
[GPT]:	In the image, the man is cutting into a chocolate frosted cake.	
[Human]:	Are there any other elements on the table besides the cake?	
[GPT]:	Yes, there are orange flowers on the table in addition to the cake.	
2nd Easiest Sample	**Text**	**Image**
[Human]:	Who is sitting at the table in the image?	
[GPT]:	A young lady is sitting at the table in the image. She appears to be celebrating her birthday, as there is a birthday cake with lit candles present.	
[Human]:	What is happening with the birthday cake?	
[GPT]:	An older woman is holding the birthday cake with lit candles and is bringing it to the young woman who's sitting at the dining room table.	
[Human]:	Is there any indication that the photo is old?	
[GPT]:	Based on my visual observation, I cannot determine whether the photo is old or not, as my analysis is purely based on the content of the image and not its quality, color, or other aspects that could suggest its age.	
[Human]:	What is the relationship between the young lady and the older woman?	
[GPT]:	Although it is difficult to be certain about the relationship between the young lady and the older woman based on the image alone, the scenario suggests that they may be family members or close friends, as they appear to be sharing a birthday celebration together. The older woman could possibly be the mother, grandmother, or an aunt of the young lady.	
3rd Easiest Sample	**Text**	**Image**
[Human]:	What activity is the woman in the image engaged in?	
[GPT]:	The woman in the image is engaged in shopping while talking on her cell phone inside of a store.	
[Human]:	What is the woman holding in her hand besides her cell phone?	
[GPT]:	Besides her cell phone, the woman is holding a small box, which possibly contains a holiday decoration.	
[Human]:	Is the woman walking or standing still while talking on her phone?	
[GPT]:	The woman is walking while talking on her cell phone in the store.	
[Human]:	Are there any noticeable features about the woman's footwear?	
[GPT]:	The woman is barefooted while talking on her phone, which is an unusual choice of footwear for shopping.	
[Human]:	What can we infer about the woman's situation during this instance?	
[GPT]:	Based on the available information, we can infer that the woman is multitasking, as she is shopping and talking on her cell phone simultaneously. The fact that she is barefooted suggests she might be preoccupied, in a hurry, or simply relaxed at the moment, not paying particular attention to her choice of footwear. Since she is holding a small box that contains a holiday decoration, it is possible that she is shopping for seasonal items or looking for a gift. The woman is likely in the middle of a discussion or seeking advice from someone on the other end of the phone call, which could be related to choosing the right product or making an important decision regarding her shopping.	
4th Easiest Sample	**Text**	**Image**
[Human]:	How many women are in the image?	
[GPT]:	There are two women in the image.	
[Human]:	What are the two women doing?	
[GPT]:	The two women are sitting at a table, having a meal together in a restaurant or a cafe.	
[Human]:	What type of food is visible on the table?	
[GPT]:	There are plates with donuts on the table in front of the women.	
[Human]:	What kind of seating arrangement do the women appear to be in?	
[GPT]:	The women appear to be sitting in a booth at a restaurant.	
[Human]:	Can you describe the social aspect of the scene?	
[GPT]:	The image portrays a casual and social setting, with two women sharing a meal and enjoying each other's company. They are likely having a conversation, exchanging stories, or discussing various topics while sitting in the booth. The presence of the donuts on the table suggests that this could be a snack or dessert shared between friends or family members. It reflects a common everyday scenario where people gather around food to strengthen their bonds, relax, and have a good time. The restaurant or cafe setting provides a comfortable and welcoming atmosphere that encourages social interactions and leisurely dining experiences.	
5th Easiest Sample	**Text**	**Image**
[Human]:	What might be the special occasion the man is celebrating?	
[GPT]:	Given the image shows a man bending over in front of a cake with lit candles and a cat nearby, it suggests that the man might be celebrating a special occasion, such as his birthday. The lit candles on the cake are a common tradition during birthday celebrations, and the presence of a cat could indicate that the man is at home with his pet, enjoying a cozy, intimate celebration.	

Table 7: Top 5 Hardest samples.

1st Hardest Sample	**Text**	**Image**
[Human]:	What precautions should someone take when planning to surf in rough waters?	
[GPT]:	When planning to surf in rough waters, like the man wearing a black wetsuit in the image, it is essential to take several precautions to ensure safety. These precautions include: 1. Wearing appropriate gear: A wetsuit provides buoyancy, warmth, and protection from any potential hazards in the water, such as rocks. 2. Checking weather and water conditions: Understanding the current ocean conditions and the forecast will help surfers anticipate how rough the waves might be and decide if it's safe to surf. 3. Being aware of rip currents: These strong currents can carry surfers away from the shore, so it's crucial to understand how to spot and escape them. 4. Surfing with a buddy: Having a surfing partner ensures there's someone to help or alert others in case of an emergency. 5. Knowing personal limits: Understanding one's skill level and not attempting to surf waves that are beyond that level is essential for safety. 6. Proper warm-up and stretching: This can help prevent injuries while surfing in challenging conditions. 7. Communicating with local surfers: They can offer insights into the specific beach break and advise on potential hazards and safe spots. By taking these precautions, surfers can significantly reduce the risks associated with surfing in rough waters and enjoy the exhilaration and challenges that such conditions offer.	
2nd Hardest Sample	**Text**	**Image**
[Human]:	What challenges do surfers commonly face?	
[GPT]:	Surfers commonly face several challenges related to the dynamic nature of the ocean and the physical demands of the sport. In the image, a person in a wet suit is riding a wave on a surfboard, which depicts some of these challenges: 1. Variable weather conditions: Surf conditions depend on factors like wind, tides, and swells; these change constantly and might affect the surfer's ability to ride waves safely and effectively. 2. Physical demands: Surfing requires balancing on a surfboard, which demands strong core, upper body, and leg muscles. Paddling to catch waves can be tiring, and surfers need stamina and endurance to stay in the water for extended periods. 3. Timing and technique: Effective surfing requires excellent coordination, timing, and technique since surfers need to catch waves at the right moment, stay on the surfboard, and perform various maneuvers, all while maintaining balance. 4. Sharks and other marine creatures: In some regions, surfers may encounter sharks or other potentially dangerous sea creatures. Awareness, safety measures, and understanding of the marine environment can help minimize the risks associated with such encounters. 5. Rip currents and undertows: These powerful water currents can pose a risk to surfers as they might make it difficult to return to shore or maintain control in the water. Overall, surfers need to develop their skills, strength, and ocean knowledge to navigate these challenges and enjoy the thrill of riding waves safely.	
3rd Hardest Sample	**Text**	**Image**
[Human]:	What are some factors that can affect this surfer's performance and safety?	
[GPT]:	Several factors can affect this surfer's performance and safety while riding the wave. Some of these factors include: 1. Wave conditions: The size, power, and shape of the wave can influence the surfer's ability to perform maneuvers and maintain balance. Unexpected changes in wave conditions or encountering a challenging wave could increase the risk of falling, getting caught inside or even being pushed towards hazardous areas. 2. Surfer's skill level: The surfer's experience and skill level play a crucial role in their performance and safety. A less experienced surfer might struggle to maintain control and execute complex maneuvers, which could result in accidents or injuries. 3. Equipment: The surfboard, leash, and wetsuit can impact the surfer's performance and safety. A poorly chosen surfboard might make it challenging to navigate the waves effectively, and a damaged leash could cause the surfer to lose their board in a wipeout, increasing the risk of injury. 4. Weather conditions: Wind and tide conditions can affect the surf and the surfer's ability to ride the wave smoothly. Strong winds might make it difficult to maintain balance and control, while extreme tide conditions may expose hidden obstacles or create unfavorable surfing conditions. 5. Ocean environment: Hidden rocks, reefs, or other hazards in the water can pose risks to the surfer's safety. Additionally, ocean currents and rip tides can move the surfer away from their intended position, which could lead to dangerous situations. By being aware of these factors, surfers can take necessary precautions, improve their skills, and choose suitable equipment to ensure a safe and enjoyable surfing experience.	
4th Hardest Sample	**Text**	**Image**
[Human]:	How many people are surfing in the image?	
[GPT]:	There are two people surfing in the image.	
[Human]:	What is the condition of the waves in the ocean?	
[GPT]:	The waves in the ocean are big and choppy, providing an exciting and challenging surfing experience for the surfers.	
[Human]:	Is one of the surfers standing up on the surfboard?	
[GPT]:	Yes, one of the surfers is standing up on the surfboard, while the other surfer is not.	
[Human]:	What are some challenges surfers might face when riding big, choppy waves?	
[GPT]:	Riding big, choppy waves presents several challenges for surfers, some of which include: 1. Balance and stability: Maintaining balance on a surfboard while maneuvering choppy waves can be difficult as the unpredictable movement of the water makes it harder for surfers to find a steady footing. 2. Timing: Catching and riding a big wave requires precise timing. Surfers need to paddle at the right speed and position themselves properly to catch the wave. Choppy conditions can make it harder to determine the best moment to catch a wave and ride it successfully. 3. Strength and endurance: To successfully ride choppy waves, surfers must have strong upper body and leg muscles to paddle, maintain balance, and push through the turbulence. These waves can also be physically exhausting, which requires good overall fitness and endurance. 4. Risk of accidents: Big, choppy waves increase the risk of accidents like wipeouts, collisions with other surfers, and encounters with dangerous underwater obstacles. The rough conditions might also cause surfers to lose control of their boards, potentially resulting in injuries. 5. Unpredictability: The unpredictable nature of choppy waves makes it more challenging for surfers to anticipate the wave's behavior and adjust their techniques accordingly. Surfers must be highly adaptable and capable of making split-second decisions to maneuver through these conditions effectively. In conclusion, surfing big, choppy waves requires advanced skills, good physical fitness, attentiveness, and adaptability. The unpredictable nature of the water and the potential risks involved make it a thrilling yet challenging activity for surfers.	
5th Hardest Sample	**Text**	**Image**
[Human]:	How many surfers are visible in the image?	
[GPT]:	There are two surfers visible in the image - one riding a wave on top of a surfboard, and the other swimming out to surf.	
[Human]:	What is happening to the surfer on the surfboard?	
[GPT]:	The surfer on the surfboard is in the midst of a fall as the wave is breaking, resulting in a wipeout.	
[Human]:	What is the second person doing in the water?	
[GPT]:	The second person is swimming out towards the waves, possibly intending to surf or catch some waves as well.	
[Human]:	What is the overall atmosphere of the scene in the image?	
[GPT]:	The overall atmosphere of the scene in the image is dynamic and action-packed, with surfers engaging in their sport amidst the powerful waves. It captures the excitement and challenges of surfing, as well as the skill and determination required to ride the waves successfully.	
[Human]:	What are the potential risks of wipeouts in surfing, and how can surfers minimize these risks?	
[GPT]:	Wipeouts are an inherent part of surfing and can pose risks to surfers due to the potential for injury or more serious accidents. Some of the potential risks associated with wipeouts include: 1. Collisions with the surfboard or other objects: Falling off the surfboard can result in surfers colliding with their own boards, causing injuries. Additionally, they might collide with rocks, reefs, or other surfers in crowded surf breaks. 2. Drowning or near-drowning: Wipeouts can cause surfers to be held underwater by the wave, resulting in disorientation and potentially drowning if they are unable to resurface in time. 3. Being caught in a rip current: A wipeout may force the surfer into a rip current, a strong water stream that can carry them away from the shore quickly and create difficulty in returning to the beach. 4. Impact injuries: A hard fall from a surfboard can result in the surfer hitting the water surface with force, leading to injuries such as sprains, fractures, or even head trauma. To minimize these risks, surfers can take several precautions: 1. Wearing appropriate protective gear: Using helmets, wetsuits or rash guards, and other protective equipment can help shield the surfer's body from injuries during wipeouts. 2. Developing swimming and water safety skills: Surfers should ensure they are strong swimmers and understand ocean safety principles, such as how to identify and escape from rip currents and using the best techniques for resurfacing after being held underwater. 3. Practicing proper surfing techniques: Mastering the sport's fundamentals, including paddling, duck diving, and foot positioning on the board, can help surfers maintain control and minimize the risk of wipeouts. 4. Surfing within their skill level: Surfers should be aware of their abilities and choose waves, surf breaks, and conditions that match their skill level, avoiding situations that might expose them to unnecessary risk. 5. Staying aware of their surroundings: Surfers should always maintain situational awareness in the water, paying attention to other surfers, potential hazards, and changing ocean conditions to prevent accidents. By following these recommendations, surfers can reduce the risks associated with wipeouts and enjoy the thrilling experience of surfing more safely.	

Table 8: Main results on MiniGPT-4.

	Full Data (3.4k)	**Random (500)**	**w/o Diversity (500)**	**Ours (500)**
OK-VQA	**40.07**	35.08	36.86	38.74
TextVQA	21.36	20.02	21.94	**21.96**
VisDial	**66.74**	55.35	60.75	65.39
VCR1_MCI	54.56	46.26	58.18	**60.01**
VCR1_OC	34.56	32.39	**36.93**	36.80
MSCOCO_MCI	48.67	42.32	51.48	**51.68**
MSCOCO_OC	37.71	36.44	**40.81**	40.63
MSCOCO_Pope	54.34	52.85	56.56	**58.63**
Average	44.75	40.09	45.44	**46.73**