Honeybee: Locality-enhanced Projector for Multimodal LLM

Junbum Cha* Wooyoung Kang* Jonghwan Mun* Byungseok Roh
Kakao Brain
{junbum.cha, edwin.kang, jason.mun, peter.roh}@kakaobrain.com

Abstract

In Multimodal Large Language Models (MLLMs), a visual projector plays a crucial role in bridging pre-trained vision encoders with LLMs, enabling profound visual understanding while harnessing the LLMs' robust capabilities. Despite the importance of the visual projector, it has been relatively less explored. In this study, we first identify two essential projector properties: (i) flexibility in managing the number of visual tokens, crucial for MLLMs' overall efficiency, and (ii) preservation of local context from visual features, vital for spatial understanding. Based on these findings, we propose a novel projector design that is both flexible and locality-enhanced, effectively satisfying the two desirable properties. Additionally, we present comprehensive strategies to effectively utilize multiple and multifaceted instruction datasets. Through extensive experiments, we examine the impact of individual design choices. Finally, our proposed MLLM, Honeybee, remarkably outperforms previous state-of-the-art methods across various benchmarks, including MME, MMBench, SEED-Bench, and LLaVA-Bench, achieving significantly higher efficiency. Code and models are available at https://github.com/kakaobrain/honeybee.

1. Introduction

Large Language Models (LLMs) have made great progress in recent years, mainly thanks to instruction tuning. Visual instruction tuning [33] has been proposed to extend LLMs into Multimodal LLMs (MLLMs) to perceive and understand visual signals (*e.g.*, images). The main idea for MLLMs is to introduce a projector connecting the vision encoder and LLM, and to learn the projector using visual instruction data while keeping the parameters of the vision encoder and LLM. Such a simple technique allows to preserve and leverage the pre-trained knowledge and abilities in vision encoder and LLM, making resulting MLLMs unlock new capabilities, such as generating stories, poems, advertisements, code, and more from given images; those tasks have traditionally been considered challenging for conventional vision-language foundation models [55, 58]. Such success leads to increasing attention for research into MLLMs taking multimodal inputs (*e.g.*, videos [28], audio [13], 3d world [17], point cloud [51]) beyond text.

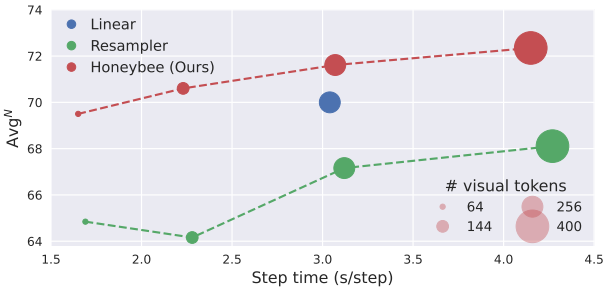


Figure 1. **Performance *vs.* efficiency for projectors** where Avg^{N} means an average of normalized benchmark scores (MME, MM-Bench, and SEED-Bench). Honeybee with the locality-enhanced projector (*i.e.*, C-Abstractor) offers a more favorable balance between efficiency and performance over existing projectors.

	MMB	SEED^{I}	MME^{P}	MME	LLaVA^{W}
Previous SoTA	67.7 [32]	68.1 [32]	1531 [32]	1848 [2]	70.7 [32]
Honeybee (Ours)	**73.6** (+5.9)	**68.6** (+0.5)	**1661** (+130)	**1977** (+129)	**77.5** (+6.8)

Table 1. **Comparison with SoTA.** The proposed Honeybee outperforms the previous state-of-the-art MLLMs on various benchmarks with significant gaps.

For MLLMs, the projector plays a critical role in the following two aspects: 1) *performance*: as it bridges the vision and language models by translating visual features into visual tokens so that the language model can understand, the quality of conveyed visual tokens directly impacts the overall performance of the MLLM; and 2) *efficiency*: as most of the computational burden lies with the language model, the efficiency of MLLMs is heavily influenced by the number of resulting visual tokens. However, despite its critical importance, the projector has been relatively underexplored in the literature and most MLLMs simply adopt either linear projectors [7, 33] or abstractors [2, 11, 27, 53, 64].

Notably, recent MLLMs prefer abstractors to linear pro-

*Equal contribution

jectors; this is primarily due to their flexibility in handling the number of resulting visual tokens, thus offering versatile design options for achieving a preferable balance between efficiency and effectiveness. However, according to our observation in Fig. 3, it turns out that the abstractors face challenges when learning tasks oriented towards spatial understanding compared to the linear projectors. This difficulty arises from the absence of a locality-aware design during the abstraction process, leading it to primarily abstract visual information from a few regions rather than retaining information about all regions, thus losing finer details crucial for spatial comprehension. In contrast, linear projectors excel at preserving the local context of visual features via one-to-one transformation. This strong preservation of locality allows effective spatial understanding.

Motivated by this, we propose novel locality-enhanced abstractors as the projector, which exhibit a more favorable balance between performance (by locality preservation) and efficiency (by abstraction capability) as presented in Fig. 1. To be specific, we introduce two locality-enhanced abstractors, C-Abstractor and D-Abstractor, by employing two powerful operations in locality modeling—convolution and deformable attention. Such injection of locality-aware design into abstractors not only promotes the overall performance improvement of MLLMs in handling intricate visual information but also capitalizes on computational efficiency during the subsequent response generation phase of LLMs.

On top of the MLLM with a locality-enhanced projector, named *Honeybee*, we offer a hidden recipe for cutting-edge MLLMs. Notably, a prevalent strategy in recent MLLM training involves multiple instruction data: 1) GPT-assisted instruction-following dataset like LLaVA [33] and 2) vision-language task datasets with *instructization*[1] process [11]. To take maximized advantage from these datasets, we present important but less explored design choices for 1) how to utilize multifaceted instruction data and 2) the effective way for an instructization process. We perform extensive experiments to verify the impact of individual design choices on diverse benchmarks and hope to offer valuable insights into training strong MLLMs.

Our main contributions are summarized as follows:

- We identify two important projector properties: 1) locality preservation of visual features and 2) flexibility to manage the number of visual tokens, and propose locality-enhanced abstractors that achieve the best of both worlds.
- We propose a (hidden) effective way to tackle multifaceted datasets as well as the instructization process, maximizing the benefit from instruction data.
- With the locality-enhanced projector and explored hidden recipes, our Honeybee achieves state-of-the-art performances across the various MLLM benchmarks—MME, MMBench, SEED-Bench, and LLaVA-Bench (Table 1).

[1]Instructization denotes conversion of raw data into instruction-following format using pre-defined templates.

2. Related Work

2.1. Multimodal Large Language Models

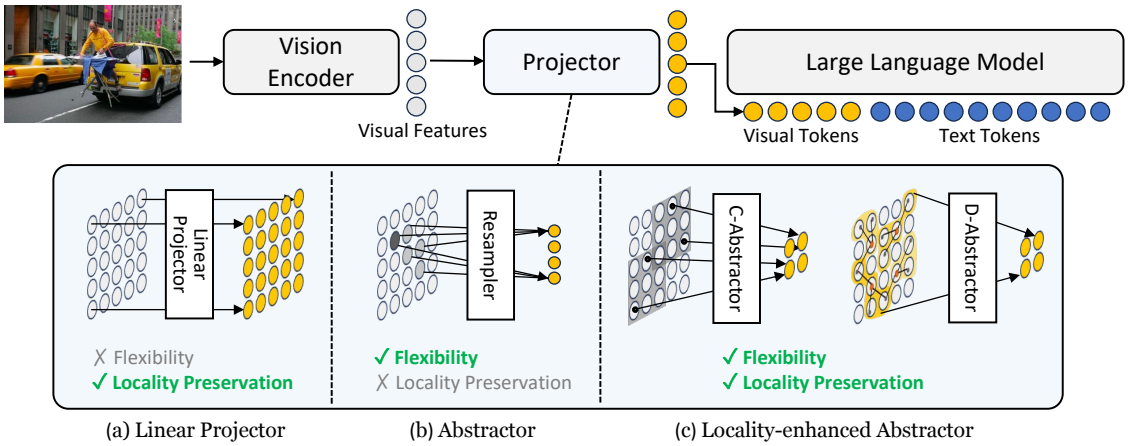
The remarkable instruction-following and generalization abilities of recent LLMs have ushered in extending LLMs to Multimodal LLMs (MLLMs). Early works such as Flamingo [1] and BLIP-2 [27] successfully adapted LLMs to visual tasks, showing notable zero-shot generalization and in-context learning capabilities. More recently, MLLMs are further advanced mainly through visual instruction tuning, which includes utilizing vision-language (VL) datasets [2, 11, 59] and enhancing visual instruction-following data [31, 33, 39, 61, 63, 64]. Also, several studies focus on grounding capabilities of MLLMs by utilizing additional datasets specifically designed for these tasks [7, 44, 52, 54]. However, recent MLLMs have not yet deeply explored visual projectors, despite the proper design of projectors is critical in both the effectiveness and efficiency of MLLMs.

2.2. Multimodal Instruction-following Data

The breakthrough from GPT-3 [4] to ChatGPT [42] highlights the importance of instruction-following data in empowering LLM to understand and follow natural language instructions. Similarly, integrating visual instruction data is essential for training MLLMs to handle various instructions, thus increasing their versatility. Several studies employ a powerful LLM, *e.g.*, GPT-4 [43], to generate visual instruction data for complex VL tasks, such as generating stories, poems, detailed captions from given images [31, 33, 61, 63, 64]. Another line of studies has explored transforming existing VL task datasets into an instruction-following format using pre-defined templates, called *instructization* [2, 11, 32, 59]. While there is active development and expansion of instruction-following datasets, the research focusing on how to combine and utilize these datasets remains underexplored.

2.3. Benchmarks for MLLM

MME [14], MMBench [34], and SEED-Bench [25] have been introduced as comprehensive benchmarks for the *objective evaluation* of MLLMs with yes/no or multiple-choice questions. These benchmarks encompass a broad spectrum of evaluation tasks, ranging from coarse- and fine-grained perceptual analysis to visual reasoning tasks. On the other hand, as the capabilities of MLLMs evolve to handle more complex VL tasks such as visual storytelling and instruction-following in an open-set manner with free-form text, other types of benchmarks have been proposed, *i.e.*, *subjective evaluation*. Following NLP studies [9, 35], several studies leverage powerful LLMs, *e.g.*, GPT-4 [43], to



(a) Linear Projector (b) Abstractor (c) Locality-enhanced Abstractor

Figure 2. **Conceptual comparison between projectors** in terms of how to transform visual features into visual tokens. (a) Linear projector performs a one-to-one transformation, thus effective in preserving local contexts of visual features, but limited in flexibility. (b) Abstractor such as resampler offers flexibility by abstracting the visual features into a smaller number of visual tokens but is limited in local context preservation by focusing on salient regions. (c) Our locality-enhanced abstractors can achieve both flexibility and locality preservation.

assess the response quality of MLLMs [3, 33, 57]. This approach aims for a more detailed evaluation of the proficiency of MLLMs. In this paper, we aim to provide valuable insights into training a robust and high-performing MLLM through extensive analysis.

3. Honeybee: Locality-enhanced MLLM

3.1. Overview

Generally, the goal of Multimodal Large Language Models (MLLMs) is to learn a model that can produce instruction-following responses for the given multimodal inputs. In this paper, we consider images as additional modality inputs to MLLMs. Thus, the language model becomes a receiver of both visual and text (instruction) tokens while generating text responses in an autoregressive manner. Formally, a multimodal input consists of two types of tokens: image tokens $\mathbf{X}_{\texttt{img}}$ and text tokens $\mathbf{X}_{\texttt{text}}$. Then, the language model predicts the response $\mathbf{Y} = \{w_i\}_{i=1}^{L}$ conditioned on the multimodal input where L means the number of tokens in the response. Therefore, the response is predicted by

$$p(\mathbf{Y}|\mathbf{X}_{\texttt{img}}, \mathbf{X}_{\texttt{text}}) = \prod_{i=1}^{L} p(w_i|\mathbf{X}_{\texttt{img}}, \mathbf{X}_{\texttt{text}}, w_{<i}). \quad (1)$$

Architecture. MLLMs are generally composed of three networks: 1) *vision encoder*, 2) *projector*, and 3) *large language model (LLM)*. The vision encoder provides a sequence of region-level visual features for detailed image understanding. The projector is in charge of transferring the visual features to visual tokens for the subsequent language model. Then, the LLM processes the fused visual and instruction tokens and produces a response autoregressively.

Efficiency of MLLMs. In the MLLM architecture, the LLM predominantly accounts for the entire computation and memory consumption of the MLLM. Thus, with the same LLM, the efficiency of the MLLM—in terms of computation, memory consumption, and throughput—is mainly affected not by the efficiency of the visual encoder and projector, but by the number of resulting visual tokens fed into the LLM. This is also shown in Fig. 1 and Appendix A.

Revisiting existing projectors. The projector takes the N visual features and converts them into M visual tokens. For the projector, MLLMs adopt an operation between a linear projection and an abstraction of visual features. The linear projection is simple yet effective, particularly in preserving knowledge and understanding of vision encoder (*e.g.*, the locality of visual features), but faces challenges in scalability and efficiency, primarily due to its inherent constraint of one-to-one transformation between visual features and tokens (*i.e.*, $M = N$). On the other hand, the abstraction offers a more adaptable approach to determining the quantity of visual tokens (M). For example, resampler and Q-former utilize M (generally $< N$ for efficiency) learnable queries and cross-attention to extract visual cues from visual features [1, 2, 11, 53, 64]. While such flexibility by abstraction allows better efficiency, but it can inherently suffer from a risk of information loss from the vision encoder.

3.2. Locality-enhanced Projector

In this section, we first describe our motivation for locality-enhanced projectors. Then, we present two types of locality-enhanced projectors (C-Abstractor and D-Abstractor) and describe the training pipeline.

3.2.1 Motivation

The projector is crucial as it bridges visual and language models, translating image features into a format that is comprehensible and utilizable by the language model. Consid-

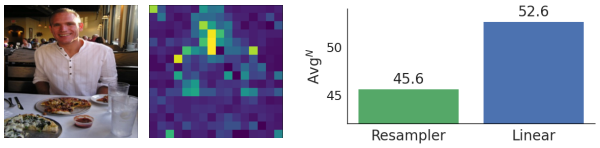


Figure 3. (Left) an example of an attention map from the resampler and (Right) a comparison of spatial understanding capability for the resampler and linear projector where Avg^{N} is computed using six spatial understanding tasks from MME, MMB, and SEED^{I}.

ering its role, when designing a projector, the most important factor is flexibility in deciding the number of resulting visual tokens. As described above, the number of visual tokens produced by the projector determines the overall efficiency and computational amount of MLLM. Considering the scenario of handling multiple or large images, improving efficiency through flexibility in reducing the number of visual tokens is highly required for scalability. This requirement has led to the preference for abstractors like resamplers and Q-formers over linear projectors in recent MLLMs [2, 11, 27, 53].

However, we observe the resampler suffers from tackling spatial understanding tasks compared to the linear projector. Note that a linear projector retains all the local context of visual features through a one-to-one projection without loss. In contrast, in Fig. 3, the resampler tends to summarize information primarily from a few regions (*e.g.*, man) while potentially overlooking details in some local regions (*e.g.*, meals, cups, background people). We believe that this difference between two models in the preservation of all local contexts (during abstraction) significantly impacted spatial understanding performance.

Stemming from these observations, we propose two novel visual projectors, C-Abstractor and D-Abstractor, under two key design principles: (*i*) enabling flexibility over the number of visual tokens and (*ii*) effectively preserving the local context. These new projectors are designed to maintain the strengths of the abstractor, such as computational efficiency via flexibility in managing visual token numbers, while also improving the preservation of local features. This enhancement not only boosts the overall performance of MLLMs in handling complex visual information but also benefits from the computational efficiency during the subsequent response generation phase of LLMs. The conceptual comparison between the existing and proposed projectors is illustrated in Fig. 2.

3.2.2 Architecture

C-Abstractor. In deep learning, convolution has been the most successful architecture for modeling local context [24, 48, 50]. Thus, we design Convolutional Abstractor, C-Abstractor, for effective local context modeling. Fig. 4a depicts the entire architecture, comprising L ResNet

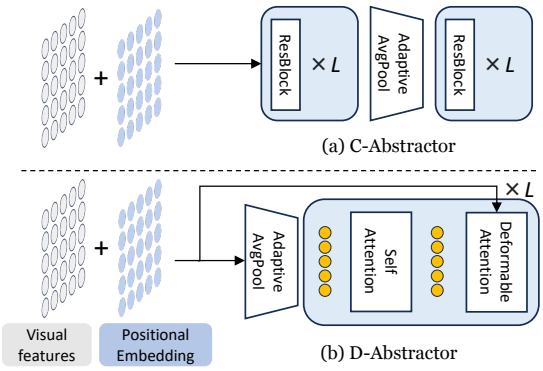


Figure 4. Conceptual architecture of our proposed abstractors.

blocks [50] followed by adaptive average pooling and another L ResNet blocks. This design allows to abstract visual features to any squared number of visual tokens, and even project to more visual tokens than the original number of visual features. We also tested several variants [36, 48], but ResNet [50] shows the best performance. Further details are provided in Appendix B.

D-Abstractor. While convolution is a successful concept in local context modeling, one can argue that it introduces overly strict inductive biases for locality. Hence, we propose Deformable attention-based Abstractor, D-Abstractor, enhancing the locality-awareness of the resampler during abstraction while keeping its flexibility. Specifically, the deformable attention [65] benefits in preserving local context; each learnable query gathers visual features via a 2-D coordinate-based sampling process using reference points and sampling offsets focusing on near the reference points. Here, we propose an advanced initialization method of reference points where the reference points are manually initialized, distributing uniformly over the whole feature map. This additional technique allows D-Abstractor to capture fine-grained and comprehensive information for a given image. More detailed explanations are given in Appendix B.

3.3. Training

We train Honeybee in the two-stage pipeline. In the first stage, we freeze the vision encoder and LLM, focusing on training the proposed locality-enhanced abstractor. In the second stage, we train both the projector and LLM to enhance deeper visual understanding and generation abilities.

Pre-training for vision-language alignment. The goal of pre-training is to learn a newly introduced visual projector to build connections between the vision encoder and LLM. Using the image-text data (*e.g.*, BlipCapFilt [26], COYO [5]), the pre-training enables MLLM to develop a nuanced understanding of how visual cues align with textual descriptions. During pre-training, the vision encoder and LLM are frozen to keep the fundamental understanding already established in vision and language models.

Task	Datasets	#samples
Captioning	BlipCapFilt [26], COYO100M [5]	200M
VQA (Open)	VQAv2 [16], GQA [20], OCRVQA [41], VSR [30]	2.2M
VQA (MC)	ScienceQA [38], A-OKVQA [47]	0.03M
REC	RefCOCO [21], RefCOCO+ [56], RefCOCOg [40], VG [23]	5.7M
Instruction	LLaVA150K [33], ShareGPT [10]	0.2M

Table 2. List of all training datasets.

Visual instruction tuning. After the pre-training of the projector for vision-language alignment, in the second stage, we jointly train the projector and LLM to enhance instruction-following capabilities and achieve a more profound visual understanding. For instruction-following, we utilize two GPT-assisted instruction-following datasets, LLaVA [33] and ShareGPT [10]. In addition, to enhance visual understanding, we instructize a wide range of existing datasets, as listed in Table 2. Specifically, our approach includes: 1) employing a range of tasks such as VQA [16, 20, 30, 41], multiple-choice VQA [38, 47], captioning [5, 26], and referring expression comprehension (visual grounding and grounded captioning) [21, 23, 40, 56]; 2) using multiple datasets for each task; 3) applying a fine-grained but single template for each dataset. Detailed examples and descriptions are in Appendix E. We thoroughly explore template-based instruction formatting strategies and the utilization of multifaceted datasets in Section 4.

4. Hidden Recipe for Visual Instruction Tuning

In Section 3, we examine the limitations of current projectors and propose methods for enhancing locality. However, a clear recipe for training cutting-edge Multimodal LLMs (MLLMs) remains unclear. While it is widely known that training MLLMs using existing datasets as instruction tuning by template-based instruction formatting is beneficial [2, 11, 32], the details of the instructization process are still underexplored—questions persist regarding dataset selection, utilization, and combination strategies. In this section, we aim to clarify these aspects via the five research questions: (i) To what extent does each dataset contribute to the performance of specific tasks? (ii) What is an effective balancing strategy between diverse datasets? (iii) What is the appropriate granularity for the templates? (iv) How significant is the diversity of the templates? (v) Do conversation-like multi-turn templates provide additional benefits?

Dataset combination. In recent MLLM studies, a diverse range of datasets has been employed for training powerful MLLMs [2, 6, 11, 32, 59]. This prevalent practice, however, is not accompanied by comprehensive analysis to identify which datasets are critical for specific tasks. To offer an in-depth analysis of this, we design a systematic ablation experiment. As outlined in Table 2, we categorize the datasets into several task groups. Then, we examine the variations in benchmark performances by sequentially excluding each task group during instruction tuning. Through these ablation experiments, we hope to offer valuable insights into the key factors for design choice regarding the dataset combination.

Dataset balancing. While a wide range of datasets are available for training MLLMs, their sizes differ substantially, as shown in Table 2. Also, when training MLLMs, it is common practice to restrict the number of training iterations to preserve the knowledge of a pre-trained LLM. Consequently, properly balancing the training datasets is crucial to maximize learning diverse skills within the short training schedule. To examine this, we compare five different balancing strategies: 1) *per-dataset*: uniform sampling for each dataset, 2) *per-task*: uniform sampling for each task, 3) *per-sample-100k*: uniform sampling for each sample with clipping the maximum size of each dataset to 100k [49], 4) *per-dataset-tuned*: empirically tuned balancing based on per-dataset strategy.

Template granularity. While the use of pre-defined templates for transforming existing datasets into an instruction format is widely recognized [11, 32, 49, 59], the appropriate granularity for applying these templates is not clearly established. We design the experiments to compare two approaches with different template granularity: 1) *fine-grained*: applying unique templates for each dataset [49], and 2) *coarse-grained*: applying the shared templates across datasets within the same task category [11, 32].

Template diversity. Prior to the emergence of GPT-assisted conversation datasets, securing template diversity was critical, often achieved by employing a range of diverse pre-defined templates alongside input inversion strategies[2] [22, 37, 59]. However, the introduction of GPT-assisted datasets has seemingly diminished the emphasis on the diversity of templates [32]. The exact role and significance of employing multiple templates and input inversion techniques in the context of GPT-assisted datasets remain less understood. To investigate this, we compare three distinct approaches utilizing: 1) a single template, 2) multiple templates, and 3) multiple templates with input inversion.

Multi-turn template. When utilizing existing datasets, it's common to find multiple input-target pairs for a single image, as seen in VQA datasets with several QA pairs per image. The multi-turn strategy merges these pairs into a single, conversation-like multi-turn example. However, this approach can merge semantically overlapped input-target pairs into one example, potentially encouraging simplistic shortcuts in finding answers, particularly in the autoregressive training of MLLMs. To mitigate this, we introduce an additional de-duplication strategy, which removes semantically duplicate input-target pairs from the multi-turn examples, thereby preventing shortcut training. We detail this strategy with examples in Appendix E.

[2]Input inversion is a task augmentation strategy by reversing input and target, *e.g.*, inversion of VQA generating questions from image and answer.

5. Experiments

5.1. Evaluation Setting

Benchmarks. We adopt four benchmarks specifically designed for Multimodal LLM (MLLM) evaluation, including MME [14], MMBench [34], SEED-Bench [25] and LLaVA-Bench (In-the-Wild) [33]. The first three assess various capabilities of MLLMs, such as perceptual understanding and visual reasoning, using binary yes/no questions (MME) or multiple-choice questions (MMBench, SEED-Bench). Note that we use splits of MME with perception tasks (MME^{P}), MMBench-dev (MMB), and SEED-Bench Image-only (SEED^{I}), respectively. Our focus on perception tasks in MME are explained in Appendix F. On the other hand, LLaVA-Bench (In-the-Wild), LLaVA^{W}, exploits GPT-4 to assess MLLM's descriptive responses, providing a comprehensive view of the model's performance in natural language generation and human preference.

Metrics. We report the official metrics computed using official implementation for individual benchmarks by default; we also report the normalized average Avg^{N} [8, 29] across benchmarks, defined as the average of scores normalized by their respective upper bound scores, facilitating straightforward comparisons.

5.2. Implementation Details

We employ 7B and 13B Vicuna-v1.5 [10] as the language model. We leverage the pre-trained CLIP ViT-L/14 [45] with resolutions of 224 and 336 for 7B- and 13B-LLM, respectively. We use the features from the second-last layer of CLIP instead of the last layer. Any image indicator tokens, *e.g.*, special tokens enclosing visual tokens, are not used. We train the entire LLM instead of parameter-efficient fine-tuning. The long (200k pre-training, 10k instruction tuning) and short (50k pre-training, 4k instruction tuning) training schedules are used for final model comparisons and detailed analyses, respectively. The short schedule is applied with Vicuna-7B, CLIP ViT-L/14, and C-Abstractor with 224 resolution and M=144 visual tokens unless stated otherwise. See Appendix C for more details.

5.3. Analysis on Locality-Enhanced Projector

Spatial understanding capability. To investigate the impact of local context preservation, we compare the spatial understanding capability on six tasks from MME, MMBench, and SEED-Bench. Table 3 summarizes the results; notably, the resampler, without consideration of local context preservation, shows poor performance. Locality-aware modeling in our projectors dramatically improves the spatial understanding capability compared to the resampler. Also, our projectors show comparable or improved performance over the linear projector with even better efficiency.

Projector	# Visual Tokens	MME	MMBench			SEED-Bench		Avg^{N}
		POS	SR	OL	PR	SR	IL	
Resampler	144	75.0	22.2	43.2	62.5	47.5	50.6	43.9
Linear	256	140.0	24.4	40.7	70.8	48.9	60.9	52.6
C-Abstractor	144	135.0	24.4	54.3	66.7	49.0	58.8	53.5
D-Abstractor	144	138.3	24.4	45.7	70.8	49.3	57.8	52.9

Table 3. **Comparison of spatial understanding capability between projectors.** The abbreviations for task names mean Position (POS) for MME, Spatial Relationship (SR), Object Localization (OL), and Physical Relation (PR) for MMBench, Spatial Relation (SR) and Instance Location (IL) for SEED-Bench. Avg^{N} indicates the normalized average over six tasks.

Performance-efficiency balance. Fig. 1 presents a comparison in terms of performance *vs.* efficiency while varying the number of visual tokens. The linear projector cannot offer flexibility due to its one-to-one conversion. Resampler and C-Abstractor provide flexible design capabilities, allowing us to customize the model to meet different requirements with a preferable balance between efficiency and effectiveness. While the resampler suffers from limited performances, our method using 144 or 256 visual tokens performs better than the linear counterpart.

5.4. Hidden Recipe for Visual Instruction Tuning

Dataset combination. Table 4 shows our comprehensive ablation study to identify the individual impact of datasets on various multimodal benchmarks. First, we investigate the significance of dataset diversity within each task type, by training Honeybee on the single dataset from each task type. This reveals an overall performance drop (D1 *vs.* D2), underscoring the importance of the *diversity of datasets* within each task type. Subsequent analysis investigates the impact of each task type by sequentially excluding specific tasks (D1 *vs.* D3-6), with the exclusion of open-ended VQA tasks notably decreasing benchmark scores. This suggests that diverse multimodal knowledge of these datasets enriches MLLM knowledge across various dimensions. Meanwhile, excluding multiple-choice VQA tasks significantly affects scores in benchmarks such as MMB and SEED^{I}, highlighting their role in aligning response patterns. The absence of captioning data particularly reduces LLaVA^{W} scores to 59.8, implying LLaVA^{W} benchmark's preference for narrative and descriptive responses, and the importance of captioning data in training. Lastly, the exclusion of visual or text instruction-following datasets (D1 *vs.* D7-10) significantly impacts LLaVA^{W}, reiterating the necessity of these datasets for instruction-following ability. In summary, these experiments show the importance of task diversity in training MLLM, encompassing a variety of task types and datasets within each task type.

Dataset balancing. The necessity of hand-crafted dataset balancing is addressed in previous studies [11, 37]. Based

	Task type						MLLM benchmark				
	Template-based				GPT-assisted		Multiple choice		Binary yes/no		GPT eval
	VQA (Open)	VQA (MC)	REC	Cap	V-Inst	T-Inst	MMB	SEED^{I}	MME^{P}	MME	LLaVA^{W}
D1	✓	✓	✓	✓	✓	✓	69.2	64.2	1568	1861	64.5
D2	✓*	✓*	✓*	✓*	✓*	✓*	67.4 (↓1.8)	63.1	1454 (↓114)	1754 (↓107)	62.2 (↓2.3)
D3		✓	✓	✓	✓	✓	68.8	62.4 (↓1.8)	1310 (↓258)	1605 (↓256)	67.0
D4	✓		✓	✓	✓	✓	30.4 (↓38.8)	20.8 (↓43.4)	1536	1829	65.4
D5	✓	✓		✓	✓	✓	68.5	63.5	1524	1787	67.0
D6	✓	✓	✓		✓	✓	69.7	63.9	1540	1846	59.8 (↓4.7)
D7	✓	✓	✓	✓		✓	70.0	64.0	1507	1805	51.9 (↓12.6)
D8	✓	✓	✓	✓	✓		68.7	64.5	1559	1851	62.7 (↓1.8)
D9	✓	✓	✓	✓			70.0	64.5	1527	1800	26.1 (↓38.4)
D10					✓	✓	43.7 (↓25.5)	0.0 (↓64.2)	1123 (↓445)	1441 (↓420)	67.0

Table 4. **The impact of data mixtures during instruction tuning.** Abbreviations for instruction data types stand for VQA (Open): open-ended visual question answering, VQA (MC): visual question answering with multiple choice, REC: referring expression comprehension, Cap: captioning, V-Inst: visual instruction, T-Inst: text-only instruction-following. The ✓* indicates that only one dataset from each task type is used to train a model, including GQA, ScienceQA, RefCOCO, COYO100M, LLaVA150k, and ShareGPT for each task.

Mixture type	MMB	SEED^{I}	MME^{P}	Avg^{N}
per-dataset	68.7	64.1	1543.2	70.0
per-task	65.7	62.1	1488.9	67.4
per-sample-100k	63.6	62.8	1494.8	67.1
per-dataset-tuned	**69.2**	**64.2**	**1568.2**	**70.6**

(a) **Dataset balancing.** Hand-crafted balancing is the best, with per-dataset strategy serving as an effective starting point for tuning.

Type	Identifier	MMB	SEED^{I}	MME^{P}	Avg^{N}	LLaVA^{W}
Inst.	instruction	**69.2**	**64.2**	**1568.2**	**70.6**	**64.5**
Multi.	dataset name	66.8	**64.2**	1483.1	68.4	64.3
Multi.	task name	68.4	64.1	1507.5	69.3	64.2

(b) **Instruction tuning *vs.* Multi-task learning.** Instruction tuning (inst.) is more effective compared to multi-task learning (multi.).

Granularity	Diversity	MMB	SEED^{I}	MME^{P}	Avg^{N}	LLaVA^{W}
fine	single	**69.2**	**64.2**	1568.2	**70.6**	**64.5**
coarse	single	68.9	64.0	1553.8	70.2	64.3
fine	multi	68.1	**64.2**	**1581.2**	70.5	61.0
fine	multi+flip	67.4	63.3	1575.9	69.8	62.7

(c) **Template granularity and diversity.** The fine-grained and single template works the best for instructization.

MT	Dedup	MMB	SEED^{I}	MME^{P}	Avg^{N}
		69.1	63.5	1518.2	69.5
✓		67.8	63.7	1546.1	69.6
✓	✓	**69.2**	**64.2**	**1568.2**	**70.6**

(d) **Multi-turn and de-duplication strategies.** Employing both strategies results in the best score.

Table 5. **Ablations on dataset balancing and instructization.** Avg^{N} indicates normalized average of MMB, SEED^{I}, and MME^{P}. Default settings are marked in gray.

on our observations in Table 4, we tune the balance of each dataset with the two principles: limiting epochs for smaller datasets and allowing up to about a few epochs for key datasets. Table 5a demonstrates the effectiveness of our manually tuned *per-dataset-tuned* approach. Without hand-crafting, the *per-dataset* can be a reliable alternative. More details are provided in Appendix C.

Instruction tuning *vs.* multi-task learning. Table 5b shows the advantages of instruction tuning with template-based formatting over multi-task learning using simple identifiers. This result aligns with prior studies [11, 49], showing the efficacy of instruction tuning in our setting.

Template granularity. Table 5c demonstrates that the fine-grained template (first row) consistently outperforms the coarse-grained template (second row) across all benchmarks. We observe that in datasets such as RefCOCO and RefCOCO+, while the input distribution $p(\mathbf{X}_{\texttt{img}}, \mathbf{X}_{\texttt{text}})$ is similar, the answer distribution $p(\mathbf{Y}|\mathbf{X}_{\texttt{img}}, \mathbf{X}_{\texttt{text}})$ differs. In this scenario, the coarse-grained template makes the model suffer from differentiating answers for similar inputs.

Template diversity. To compare the effect of template diversity on model performance, we evaluate three scenarios with different diversities: using a single template (single), employing 10 templates for each dataset (multi), and inverting 3 out of 10 templates (multi+flip). Interestingly, our experiments reveal that increasing template diversity does not guarantee a performance boost, as shown in Table 5c. This is consistent results with recent studies [32], showing that effective zero-shot generalization is achievable even without using multiple templates.

Multi-turn template. Table 5d shows the effectiveness of both multi-turn template and de-duplication strategies. The results imply removing the semantically overlapping pairs in each example is effective for mitigating shortcut training.

Additional recipes. Apart from dataset and instructization strategies, training recipes also incorporate several subtle yet crucial design choices, including the selection of features in vision encoder, LLMs, LLM training techniques, image indicators, pre-training and instruction tuning iterations. These recipes are detailed in Appendix D.

Method	LLM	Projector	Vision Enc.	Res.	MMB	MME^{P}	MME	SEED^{I}	LLaVA^{W}
Approaches using 7B LLM									
LLaVA (v1) [33]	LLaMA-7B	Linear	CLIP ViT-L/14	224	38.7	502.8	717.5	33.5	-
MiniGPT-4 [64]	Vicuna-7B	Resampler	EVA-CLIP ViT-G	224	24.3	581.7	726.0	47.4	-
LLaMA-AdapterV2 [15]	LLaMA-7B	LLaMA-Adapter	CLIP ViT-L/14	224	41.0	972.7	1221.6	32.7	-
mPLUG-Owl [53]	LLaMA-7B	Resampler	CLIP ViT-L/14	224	49.4	967.3	1243.4	34.0	-
InstructBLIP [11]	Vicuna-7B	Q-former	EVA-CLIP ViT-G	224	36.0	-	-	58.8	60.9
IDEFICS	LLaMA-7B	Flamingo	OpenCLIP ViT-H/14	224	48.2	-	-	44.5	-
Shikra [7]	Vicuna-7B	Linear	CLIP ViT-L/14	224	58.8	-	-		-
Qwen-VL [2]	Qwen-7B	Resampler	OpenCLIP ViT-bigG	448	38.2	-	-	62.3	-
Qwen-VL-Chat [2]	Qwen-7B	Resampler	OpenCLIP ViT-bigG	448	60.6	1487.5	1848.3	**65.4**	-
LLaVA-1.5 [32]	Vicuna-7B	Linear	CLIP ViT-L/14	336	64.3	1510.7	-	-	63.4
Honeybee (M=144)	Vicuna-7B	C-Abstractor	CLIP ViT-L/14	224	70.1	**1584.2**	**1891.3**	64.5	**67.1**
		D-Abstractor			**70.8**	1544.1	1835.5	63.8	66.3
Approaches using 13B LLM									
MiniGPT-4 [64]	Vicuna-13B	Resampler	EVA-CLIP ViT-G	224	-	866.6	1158.7	-	-
BLIP-2 [27]	Vicuna-13B	Q-former	EVA-CLIP ViT-G	224	-	1293.8	-	-	38.1
InstructBLIP [11]	Vicuna-13B	Q-former	EVA-CLIP ViT-G	224	44.0	1212.8	1504.6	-	58.2
LLaVA-1.5 [32]	Vicuna-13B	Linear	CLIP ViT-L/14	336	67.7	1531.3	1826.7	68.1	70.7
Honeybee (M=256)	Vicuna-13B	C-Abstractor	CLIP ViT-L/14	336	73.2	1629.3	1944.0	**68.2**	**75.7**
		D-Abstractor			**73.5**	**1632.0**	**1950.0**	66.6	72.9

Table 6. **Comparison with other state-of-the-art MLLMs.** Res. and M indicate the image resolution and the number of visual tokens, respectively. We highlight the **best results** and second-best results in bold and underline.

5.5. Putting It Altogether

In Table 6, we compare our Honeybee, optimized as previously discussed, with other state-of-the-art MLLMs. Honeybee outperforms comparable 7B-scale MLLMs in all benchmarks, except for SEED^{I}. It is worth noting that competing methods like Qwen-VL [2] and LLaVA-1.5 [32] use larger vision encoders (*e.g.*, ViT-bigG for Qwen-VL) or larger images (448 and 336) with increased visual tokens (256 and 576), while Honeybee employs ViT-L/14 with 224 resolution and 144 visual tokens. Given the focus on the detailed visual understanding of SEED^{I} (See Appendix F), larger images or more visual tokens can be beneficial. With increased visual tokens (144 to 256), Honeybee achieves the best score in SEED^{I} (65.5) with 7B-scale LLM, as shown in Table 7. With 13B-scale LLMs, Honeybee surpasses all previous methods in every benchmark. The detailed scores are available in Appendix G.1.

5.6. Additional Results

Pushing the limits. In our final 7B and 13B models, we use 144 and 256 visual tokens (M) respectively, balancing efficiency and performance. As indicated in Fig. 1 and Appendix A, increasing M consistently improves performance. Our experiments, aligning M in Honeybee with the linear projector (Table 7), show performance enhancement at the cost of efficiency. Additional comparisons with previous methods are in Appendix G.2.

ScienceQA [38] evaluation results are presented in Appendix G.3. Remarkably, without any specialized fine-tuning, our generalist Honeybee achieves state-of-the-art performance (94.39), outperforming specialist models such as MM-CoT (91.68) [62] and LLaVA+GPT-4 (92.53) [33].

LLM	Res.	M	s/step	MMB	MME^{P}	MME	SEED^{I}	LLaVA^{W}
7B	224	144	2.23	70.1	1584.2	1891.3	64.5	67.1
		256	3.07	**71.0**	**1592.7**	**1951.3**	**65.5**	**70.6**
13B	336	256	5.52	73.2	1629.3	1944.0	68.2	75.7
		576	9.80	**73.6**	**1661.1**	**1976.5**	**68.6**	**77.5**

Table 7. **Pushing the limits** with C-Abstractor by increasing the number of visual tokens (M). *s/step* denotes pre-training step time.

Qualitative examples are provided in Appendix H.2.

6. Conclusion

The advent of visual instruction tuning has brought remarkable advances in MLLMs. Despite these strides, areas such as projector design and the approach in handling multifaceted data with instructization processes remain underexplored or unclear. Addressing these gaps, we identify the desirable but overlooked projector property, *i.e.*, locality preservation, and propose the locality-enhanced projector offering a preferable balance between performance and efficiency. In addition, we provide extensive experiments to identify the impact of individual design choices in handling multifaceted instruction data, unveiling hidden recipes for high-performing MLLM development. Finally, our proposed MLLM, Honeybee, remarkably outperforms previous state-of-the-art methods across various benchmarks.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning. In *NeurIPS*, 2022. 2, 3

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966*, 2023. 1, 2, 3, 4, 5, 8, 14, 17

[3] Shuai Bai, Shusheng Yang, Jinze Bai, Peng Wang, Xingxuan Zhang, Junyang Lin, Xinggang Wang, Chang Zhou, and Jingren Zhou. TouchStone: Evaluating Vision-Language Models by Language Models. *arXiv preprint arXiv:2308.16890*, 2023. 3

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-shot Learners. In *NeurIPS*, 2020. 2

[5] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. COYO-700M: Image-Text Pair Dataset. https://github.com/kakaobrain/coyo-dataset, 2022. 4, 5

[6] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechu Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. MiniGPT-v2: Large Language Model as A Unified Interface for Vision-Language Multi-task Learning. *arXiv preprint arXiv:2310.09478*, 2023. 5

[7] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic. *arXiv preprint arXiv:2306.15195*, 2023. 1, 2, 8

[8] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal Image-Text Representation Learning. In *ECCV*, 2020. 6

[9] Cheng-Han Chiang and Hung-yi Lee. Can Large Language Models Be an Alternative to Human Evaluations? *arXiv preprint arXiv:2305.01937*, 2023. 2

[10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality, 2023. 5, 6, 18, 19

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. *arXiv preprint arXiv:2305.06500*, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[12] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 13

[13] Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. Pengi: An Audio Language Model for Audio Tasks. *arXiv preprint arXiv:2305.11834*, 2023. 1

[14] 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. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. *arXiv preprint arXiv:2306.13394*, 2023. 2, 6

[15] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model. *arXiv preprint arXiv:2304.15010*, 2023. 8

[16] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *CVPR*, 2017. 5

[17] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3D-LLM: Injecting the 3D World into Large Language Models. *arXiv preprint arXiv:2307.12981*, 2023. 1

[18] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 14

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 12

[20] Drew A Hudson and Christopher D Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *CVPR*, 2019. 5, 14

[21] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to Objects in Photographs of Natural Scenes. In *EMNLP*, 2014. 5

[22] Dohwan Ko, Ji Soo Lee, Wooyoung Kang, Byungseok Roh, and Hyunwoo J. Kim. Large Language Models are Temporal and Causal Reasoners for Video Question Answering. In *EMNLP*, 2023. 5

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *IJCV*, 2017. 5

[24] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 4

[25] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-Bench: Benchmarking Multimodal LLMs with Generative Comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 2, 6

[26] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *ICML*, 2022. 4, 5

[27] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *ICML*, 2023. 1, 2, 4, 8, 14

[28] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 1

[29] Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, Yu Cheng, Luowei Zhou, Xin Eric Wang, William Yang Wang, et al. VALUE: A Multi-Task Benchmark for Video-and-Language Understanding Evaluation. *arXiv preprint arXiv:2106.04632*, 2021. 6

[30] Fangyu Liu, Guy Edward Toh Emerson, and Nigel Collier. Visual Spatial Reasoning. *Transactions of the Association for Computational Linguistics*, 2023. 5

[31] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning Large Multi-Modal Model with Robust Instruction Tuning. *arXiv preprint arXiv:2306.14565*, 2023. 2

[32] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 1, 2, 5, 7, 8, 12, 14, 17

[33] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *NeurIPS*, 2023. 1, 2, 3, 5, 6, 8, 17, 18, 19

[34] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is Your Multi-modal Model an All-around Player? *arXiv preprint arXiv:2307.06281*, 2023. 2, 6

[35] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. *arXiv preprint arXiv:2303.16634*, 2023. 2

[36] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 4, 12

[37] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. In *ICML*, 2023. 5, 6

[38] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *NeurIPS*, 2022. 5, 8, 17

[39] Yadong Lu, Chunyuan Li, Haotian Liu, Jianwei Yang, Jianfeng Gao, and Yelong Shen. An Empirical Study of Scaling Instruct-tuned Large Multimodal Models. *arXiv preprint arXiv:2309.09958*, 2023. 2

[40] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and Comprehension of Unambiguous Object Descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 5

[41] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. OCR-VQA: Visual Question Answering by Reading Text in Images. In *ICDAR*, 2019. 5

[42] OpenAI. ChatGPT, 2023. 2

[43] OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023. 2, 17

[44] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding Multimodal Large Language Models to the World. *arXiv preprint arXiv:2306.14824*, 2023. 2

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 6, 14

[46] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020. 13

[47] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-OKVQA: A Benchmark for Visual Question Answering using World Knowledge. In *ECCV*, 2022. 5

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[49] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *ICLR*, 2022. 5, 7

[50] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 4, 12

[51] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. PointLLM: Empowering Large Language Models to Understand Point Clouds. *arXiv preprint arXiv:2308.16911*, 2023. 1

[52] Shiyu Xuan, Qingpei Guo, Ming Yang, and Shiliang Zhang. Pink: Unveiling the Power of Referential Comprehension for Multi-modal LLMs. *arXiv preprint arXiv:2310.00582*, 2023. 2

[53] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mPLUG-Owl: Modularization Empowers Large Language Models with Multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 1, 3, 4, 8, 14, 18, 19

[54] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and Ground Anything Anywhere at Any Granularity. *arXiv preprint arXiv:2310.07704*, 2023. 2

[55] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. 1

[56] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling Context in Referring Expressions. In *ECCV*, 2016. 5

[57] 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. *arXiv preprint arXiv:2308.02490*, 2023. 3

[58] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A New Foundation Model for Computer Vision. *arXiv preprint arXiv:2111.11432*, 2021. 1

[59] Yan Zeng, Hanbo Zhang, Jiani Zheng, Jiangnan Xia, Guoqiang Wei, Yang Wei, Yuchen Zhang, and Tao Kong. What Matters in Training a GPT4-Style Language Model with Multimodal Inputs? *arXiv preprint arXiv:2307.02469*, 2023. 2, 5

[60] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention. *arXiv preprint arXiv:2303.16199*, 2023. 17

[61] Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. LLaVAR: Enhanced Visual Instruction Tuning for Text-rich Image Understanding. *arXiv preprint arXiv:2306.17107*, 2023. 2

[62] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal Chain-of-Thought Reasoning in Language Models. *arXiv preprint arXiv:2302.00923*, 2023. 8, 17

[63] Bo Zhao, Boya Wu, and Tiejun Huang. SVIT: Scaling up Visual Instruction Tuning. *arXiv preprint arXiv:2307.04087*, 2023. 2

[64] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 2, 3, 8

[65] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR*, 2021. 4, 12

Projector	M	s/step	MMB	SEED^{I}	MME^{P}	Avg^{N}
Linear	256	3.04	67.1	65.1	1556.5	70.0
Resampler	64	1.69	65.9	58.9	1394.7	64.8
	144	2.28	66.0	57.0	1389.6	64.2
	256	3.12	67.1	59.9	1489.6	67.2
	400	4.27	67.7	61.5	1502.5	68.1
C-Abstractor	64	1.65	69.2	62.9	1528.1	69.5
	144	2.23	69.2	64.2	1568.2	70.6
	256	3.07	70.2	65.3	1586.8	71.6
	400	4.15	70.8	65.5	1615.0	72.3

Table 8. Detailed scores of projectors by the number of visual tokens (M). *s/step* indicates the time spent to perform one step in pre-training.

Architectures	MMB	SEED^{I}	MME^{P}	Avg^{N}
Linear	67.1	65.1	1556.5	70.0
2-layer MLP	68.3	64.5	1557.2	70.2
6-layer MLP	68.5	63.5	1509.2	69.1
Resampler	66.0	57.0	1389.6	64.2
Resampler$_{\text{w/ pos-emb}}$	65.9	58.0	1384.7	64.4
ResNet (C-Abstractor)	69.2	64.2	1568.2	70.6
ConvNext	66.2	61.9	1525.4	68.1
StandardConv	67.4	57.1	1409.7	65.0
Deformable (D-Abstractor)	68.6	63.2	1548.3	69.7
Deformable$_{\text{w/o v-pooled Q}}$	68.4	63.1	1521.7	69.2
Deformable$_{\text{w/o M-RP}}$	68.5	62.9	1497.0	68.7

Table 9. Ablations for various architectural design choices in each projector. We use 144 visual tokens (M=144) for all architectures except for Linear and MLPs (M=256) due to their inflexibility.

A. Efficiency of MLLMs

As described in Section 3 of the main text, the efficiency of MLLMs is predominantly affected not by the efficiency of the vision model or projector, but by the number of visual tokens (*i.e.*, the number of output tokens of the projector). Table 8 demonstrates this description, complementing Fig. 1. Notably, while the resampler has substantially larger parameters than linear (105M *vs*. 4M parameters), MLLM with resampler with $M = 144$ is more efficient than MLLM with linear ($M = 256$), as shown by lower step times (2.28 *vs*. 3.04). Our C-Abstractor, adhering to our design principles of flexibility and locality preservation, stands out as a Pareto-front model compared to both resampler and linear.

B. Details on Projectors

In this section, we provide further ablations and descriptions for design choices of individual projectors.

B.1. Linear Projector

In the recent study, LLaVA (v1.5) [32] utilizes a 2-layer MLP instead of a single linear projection for enhancing the vision-language connector's representation power. This approach led to an investigation of how varying the number of MLP layers impacts overall performance. As shown in Table 9, the 2-layer MLP-based projector marginally improves the overall performance compared to the linear projector. However, we observe a slight performance drop when further increasing the number of MLP layers (*i.e.*, 6-layer MLP). We note that our C- and D-Abstractors achieve better or comparable benchmark scores while using fewer visual tokens, indicating our abstractors' superiority regarding the balance of efficiency and performance.

B.2. Resampler

As described in the main text, our design focuses on two principles: 1) flexibility in visual token counts, which is the key factor to the efficiency of MLLM, and 2) preservation of local context, which is critical for spatial understanding. Our first try is augmenting visual features with positional embeddings in the resampler framework, but it does not yield notable improvements (See Resampler$_{\text{w/ pos-emb}}$ in Table 9). This leads us to design two novel abstractors, C-Abstractor and D-Abstractor.

B.3. C-Abstractor

Under our design principles on flexibility and locality, we introduce convolution layers and adaptive average pooling into the projector. The overall architecture is illustrated in Fig. 4. We compare three convolution blocks: 1) ResNet bottleneck block [50] with squeeze-excitation [19], 2) ConvNext block [36], and 3) a standard convolution block (3×3 convolution layer). Table 9 shows ResNet block outperforms ConvNext and standard convolution (StandardConv) blocks. Hence, we employ ResNet block for C-Abstractor. While further architectural variations are explorable under the proposed design principles, we leave them for future investigation.

B.4. D-Abstractor

We first describe how deformable attention [65] works in D-Abstractor. The core components of deformable attention include (i) 2-D reference points p, (ii) 2-D sampling offsets Δo, and (iii) attention weights A. For individual learnable queries $\mathbf{z}$, the feature aggregation from the visual feature map X_{feat} is formulated by[3]:

$$\mathbf{z}^{l+1} = \sum_{k=1}^{K} A_k^l \cdot X_{feat}(p + \Delta o_k^l), \tag{2}$$

where K is the number of sampling offsets per reference point, and l is the index of the attention layer. All the ref-

[3]We recommend reading [65] for more details.

	Ablated setting	Default value	Changed value	MMB	SEED^{I}	MME^{P}	MME	Avg^{N}	LLaVA^{W}
	(Default) Honeybee with short training schedule			69.2	64.2	1568.2	1860.7	70.6	64.5
(i)	Image indicator	✗	✓	67.4	62.5	1543.4	1809.5	69.0	60.5
(ii)	Visual feature layer	Second-last	Last	69.2	63.7	1566.1	1839.3	70.4	62.1
(iii)	LLM	Vicuna-v1.5	LLaMA-2-chat	70.0	63.6	1551.7	1822.0	70.4	62.8
(iv)	LLM tuning	Full	LoRA ($r = 64$)	35.0	48.9	1016.1	1156.1	44.9	59.2
			LoRA ($r = 256$)	47.3	49.9	959.1	1217.3	48.4	64.0
(v)	Pre-training steps	50k	200k	69.1	63.8	1586.6	1855.2	70.7	66.4
(vi)	Instruction tuning steps	4k	10k	69.3	64.3	1586.8	1868.6	71.0	66.6
			16k	70.9	63.8	1550.6	1856.7	70.7	66.0

Table 10. **Additional recipes.** The default value indicates the choice used in our default ablation setting with the short training schedule.

Configuration	Pre-training	Instruction Tuning
Trainable modules	Abstractor	Abstractor, LLM
Batch size	256	128
Learning rate	3e-4	2e-5
Minimum LR	1e-5	1e-6
LR schedule	Cosine decay	
Warmup steps	2000	150
Training steps	200k	10k
Weight decay	0.01	1e-4
Optimizer	AdamW	
Optimizer HPs	$\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1e-6$	
Gradient clipping	1.0	

Table 11. **Training hyperparameters.** HP and LR indicate hyperparameter and learning rate, respectively. Note that we use LR of 1e-4 for D-Abstractor.

erence points, sampling offsets, and attention weights are obtained via linear projection over the learnable queries $\mathbf{z}$; that is, they are all learnable values. The introduction of reference points and sampling offsets for learnable queries allows locality modeling by enabling the collection of features near reference points via the sampling offsets.

On top of the deformable attention, we additionally present two techniques to improve local context modeling: 1) learnable query initialization through adaptive average pooling to the visual feature map instead of random initialization (*v-pooled Q*), and 2) a manual initialization of reference points uniformly distributing on visual feature maps instead of centralized initialization (*M-RP*). With these techniques, we can make reference points cover the whole region of an image, which results in offering more benefits in preserving local context with fine-grained information for a given image. The results in Table 9 demonstrate that two techniques provide overall performance improvements of MLLMs.

Task	Dataset	Ratio	Task	Dataset	Ratio
VQA (Open)	VQAv2	10.3%	REC	RefCOCO	10.3%
	GQA	10.3%		RefCOCO+	10.3%
	OCRVQA	5.1%		RefCOCOg	10.3%
	VSR	2.6%		VG	5.1%
VQA (MC)	ScienceQA	5.1%	Instruction	LLaVA150K	10.3%
	A-OKVQA	10.3%		ShareGPT	2.6%
Captioning	COYO100M	7.7%			

Table 12. Sampling ratio during instruction tuning.

C. Implementation Details

The detailed hyperparameters (HPs) are summarized in Table 11. Additionally, we utilize total six blocks in both C-Abstractor and D-Abstractor (*i.e.*, $L = 3$ for C-Abstractor and $L = 6$ for D-Abstractor in Fig. 4). We use a single node with A100 80GB $\times$ 8, employing deepspeed zero-2 [46] and flash-attention v2 [12] for all experiments, except for the long schedule pre-training where we use multinode setups.

Sampling ratio for datasets. As described in Section 4, balancing the wide range of datasets is important to train precise MLLMs. To maximize the learning of diverse knowledge from multifaceted datasets, we manually determine the sampling ratios of these datasets during training. In pre-training, COYO100M and BlipCapFilt are used in a 1:1 ratio. For instruction tuning, the specific sampling ratios of each dataset, determined through short schedule ablations, are detailed in Table 12. Notably, datasets such as VSR, ShareGPT, ScienceQA, OCRVQA, and Visual Genome (VG) have lower sampling ratios. The restricted scale of ShareGPT, VSR, and ScienceQA is due to their small dataset sizes, limited to a maximum of 3 epochs in short schedule criteria. On the other hand, the sampling ratio for OCRVQA and VG is set to 5.1%, derived empirically from ablation experiments. The exclusion of BlipCapFilt in instruction tuning stems from computational resource con-

Task	Dataset	Template
Captioning	BlipCapFilt	AI: {**caption**}
	COYO100M	AI: {**caption**}
VQA (Open)	VQAv2	Human: Answer the question using a single word or phrase. {**question**} AI: {**answer**}
	GQA	Human: Answer the question using a single word or phrase. {**question**} AI: {**answer**}
	OCRVQA	Human: Answer the question using a single word or phrase. {**question**} AI: {**answer**}
	VSR	Human: Answer the question using a single word or phrase. {**question**} Please answer yes or no. AI: {**answer**}
VQA (MC)	ScienceQA	Human: Answer with the option's letter from the given choices directly. {**question**} Context: {**context**} There are several options: {**option**} AI: {**answer**}
	A-OKVQA	Answer with the option's letter from the given choices directly. {**question**} There are several options: {**option**} AI: {**answer**}
REC	RefCOCO	Human: Provide the bounding box coordinate of the region this sentence describes: {**phrase**} AI: {**bbox**}
		Human: Provide a description for the region {**bbox**}, utilizing positional words to refer to objects. Example: 'The large blue teddy bear next to the red balloon' AI: {**phrase**}
	RefCOCO+	Human: Provide the bounding box coordinate of the region this sentence describes: {**phrase**} AI: {**bbox**}
		Human: Provide a description for the region {**bbox**}, focusing on the appearance of objects without using positional words. Example: 'The large blue teddy bear holding a red balloon.' AI: {**phrase**}
	RefCOCOg	Human: Provide the bounding box coordinate of the region this sentence describes: {**phrase**} AI: {**bbox**}
		Human: Provide a description for the region {**bbox**}, using detailed and descriptive expressions to refer to objects. Example: 'The large blue teddy bear holding a red balloon with a joyful expression.' AI: {**phrase**}
	Visual Genome	Human: Provide the bounding box coordinate of the region this sentence describes: {**phrase**} AI: {**bbox**}
		Human: Provide a short description for this region: {**bbox**} AI: {**phrase**}
Instruction	LLaVA150k	Human: {**instruction**} AI: {**response**}
	ShareGPT	Human: {**instruction**} AI: {**response**}

Table 13. **Templates for individual dataset.** We develop the templates based on LLaVA (v1.5) [32]. {*} is replaced depending on dataset examples where red-colored one means a target output. Note that *bbox* is expressed as normalized coordinates $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$.

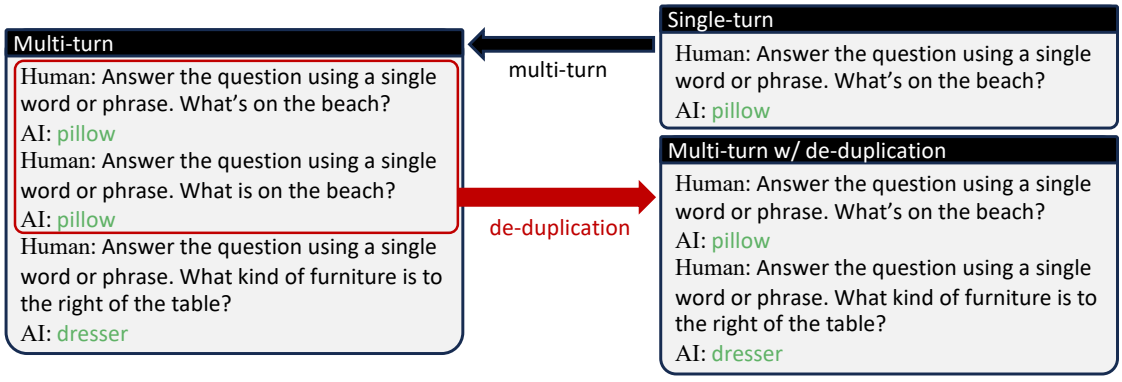


Figure 5. **The construction process of a multi-turn example with de-duplication.** This example is sampled from the GQA [20] dataset.

straints, not from ablation results; we observe that including it does not notably affect the average performance.

D. Additional Recipes

Table 10 presents additional ablation studies for our design choices. (i) There are several studies employing image indicator tokens [2, 53], yet they do not demonstrate the effectiveness of the indicator tokens. Our experiments show that omitting indicator tokens improves performance. (ii) We experiment with visual feature sources from the CLIP vision model [45]. The results show that utilizing features from the second-last layer rather than the last layer yields better performance [27]. (iii) LLaMA-2-chat and Vicuna-v1.5 show similar results, with Vicuna marginally outperforming, thus we use Vicuna. (iv) We applied LoRA to every query and value layer of attention following the original paper [18], yet found full tuning of LLM to be superior. While there may be ways to better utilize LoRA, such as increas-

Q: What item is hanging on the wall behind the person in the image?
A. Picture B. Clock
C. Shelf D. Cabinet

Q: What color are the socks of the player nearest to the ball in the image?
A. Yellow and blue **B. Red**
C. Black and white D. Blue and yellow

Q: In the image, where is the person surfing?
A. On a surfboard riding a large wave
B. In a group of surfers riding wave
C. Close to the shore
D. In the middle of the ocean

Figure 6. **Examples of SEED-Bench.** The examples require in-depth visual understanding; we highlight the regions (yellow boxes) that we need to focus on to get the correct answer (red-colored option).

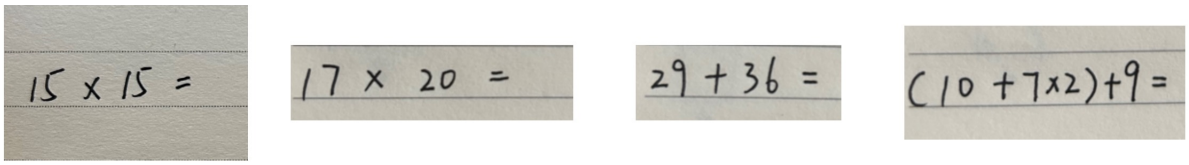
Q: The image shows a python code. Is the output of the code 'Hello'? Please answer yes or no. **A. Yes**

Q: The image shows a python code. Is the output of the code 'a dog'? Please answer yes or no. **A. No**

Q: The image shows a python code. Is the output of the code '12'? Please answer yes or no. **A. Yes**

Q: The image shows a python code. Is the output of the code '2'? Please answer yes or no. **A. No**

(a) Code reasoning task



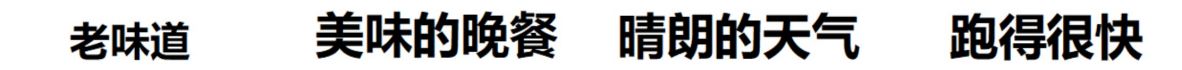
Q:Is the answer to the arithmetic question in the image 1511? Please answer yes or no. **A. No**

Q:Is the answer to the arithmetic question in the image 17? Please answer yes or no. **A. No**

Q:Is the answer to the arithmetic question in the image 65? Please answer yes or no. **A. Yes**

Q:Is the answer to the arithmetic question in the image 33? Please answer yes or no. **A. Yes**

(b) Numerical calculation task



Q:Is it appropriate to translate the Chinese in the image into English 'classic taste' in the picture? Please answer yes or no. **A. Yes**

Q:Is it appropriate to translate the Chinese in the image into English 'a delicious dinner' in the picture? Please answer yes or no. **A. Yes**

Q:Is it appropriate to translate the Chinese in the image into English 'cold weather' in the picture? Please answer yes or no. **A. No**

Q:Is it appropriate to translate the Chinese in the image into English 'run very slow' in the picture? Please answer yes or no. **A. No**

(c) Text translation task

Figure 7. **Examples of MME with cognition taks.**

ing its application scope or rank, we did not explore these further in this study. Experiments (v) and (vi) pertain to the long training schedule employed for our final model (Table 6). (v) In pre-training, we freeze the LLM and train only the projector. Here, extending pre-training, a feasible option with more computational resources, is beneficial, albeit with marginal improvements. (vi) When increasing instruction tuning steps, a broader consideration is necessary as continued LLM training can diminish its pre-trained knowledge and capabilities. Our experiments reveal that excessively long training is counterproductive, with around 10k training iterations being the most effective.

E. Details on Templates

Templates. Detailed templates for individual datasets are presented in Table 13. For captioning tasks, MLLMs are encouraged to generate directly output captions without any instructional phrase as the standard captioning task. For VQA and REC tasks, we adopt *fine-grained* templates to favorably adapt LLM's outputs for individual datasets. For the VSR dataset, we rephrase the declarative captions into questions to suit a VQA context. For instance, a caption "The cat is inside the refrigerator" marked as *False* is converted into "Is the cat inside the refrigerator?" with the answer *No*. Finally, for the instruction task, we use the original instruc-

	Perception											Cognition					
Model	Existence	Count	Position	Color	Poster	Celebrity	Scene	Landmark	Artwork	OCR	Sum	Commonsense reasoning	Numerical calculation	Text translation	Code reasoning	Sum	Total
C-7B	185.0	145.0	161.7	180.0	166.7	152.4	157.3	174.5	129.3	132.5	1584.2	112.1	37.5	100.0	57.5	307.1	1891.3
D-7B	175.0	153.3	143.3	175.0	155.4	148.2	153.3	163.3	129.8	147.5	1544.1	111.4	47.5	72.5	60.0	291.4	1835.5
C-13B	185.0	141.7	173.3	170.0	178.2	172.4	160.3	173.5	142.5	132.5	1629.3	127.1	47.5	80.0	60.0	314.6	1944.0
D-13B	195.0	175.0	146.7	168.3	168.0	164.7	156.5	174.5	131.0	152.5	1632.2	130.0	62.5	82.5	42.5	317.5	1949.7

(a) **MME scores.** Maximum scores are 200 for each subcategory, and 2000, 800, and 2800 for perception, cognition, and total, respectively.

Model	Scene understanding	Instance identity	Instance attributes	Instance location	Instances counting	Spatial relation	Instance interaction	Visual reasoning	Text understanding	Total
C-7B	73.4	67.8	64.6	59.8	55.6	48.4	73.2	74.9	41.2	64.5
D-7B	73.1	67.9	62.3	60.8	55.0	49.8	67.0	73.1	27.1	63.5
C-13B	75.4	74.0	68.1	65.5	59.2	54.2	71.1	79.5	38.8	68.2
D-13B	74.8	71.2	65.4	64.6	59.3	51.6	69.1	78.5	24.7	66.6

(b) **SEED^{I} accuracies.**

Model	LR	AR	RR	FP-S	FP-C	CP	Total
C-7B	41.7	78.1	69.6	74.1	53.8	80.2	70.1
D-7B	44.2	75.1	73.0	73.1	58.6	81.2	70.8
C-13B	45.8	77.6	77.4	76.8	57.9	83.6	73.2
D-13B	45.0	75.6	81.7	76.4	62.1	82.9	73.5

(c) **MMB accuracies.** Abbreviations stand for LR: Logic Reasoning, AR: Attribute Reasoning, RR: Relation Reasoning, FP-S: Fine-grained Perception (Single-instance), FP-C: Fine-grained Perception (Cross-instance), CP: Coarse Perception.

Model	Complex	Conv	Detail	All
C-7B	84.6	50.3	55.1	67.1
D-7B	79.6	49.4	62.6	66.3
C-13B	82.5	72.9	66.7	75.7
D-13B	84.1	68.6	57.8	72.9

(d) **LLaVA^{W} scores.**

Table 14. **Detailed scores.** C- and D- in Model column indicate C-Abstractor and D-Abstractor, respectively. 7B and 13B indicate LLM size. For the input images, we use 224 resolution for 7B and 336 for 13B.

tions and responses rather than using templates.

Multi-turn with de-duplication. For data such as VQA datasets where multiple input-target pairs exist for a single image, we make conversation-like multi-turn examples by simply concatenating the input-target pairs. Additionally, we perform a de-duplication strategy which remains only one from the duplicates (having the same target). The process is illustrated in Fig. 5.

F. Benchmark Characteristics

Throughout this study, we observe specific characteristics in benchmarks, particularly in SEED-Bench and MME with cognition tasks (MME-cognition). SEED-Bench tends to require fine-grained visual comprehension, while MME-cognition is highly text-oriented, resulting in substantial dependency on the capabilities of LLMs. In this section, we investigate these distinctive benchmark characteristics.

SEED-Bench. We present examples of SEED-Bench, in Fig. 6, to show one of the major characteristics of the benchmark; we observe that the examples frequently require fine-grained visual understanding, *e.g.*, details from small regions. Such characteristics suggest that using large images or more visual tokens is critical in achieving higher performance in this benchmark. Notably, in Table 6, Honeybee achieves competitive performance over comparative models even with smaller images or fewer visual tokens.

MME-cognition. We present examples of MME-cognition in Fig. 7. Notably, three out of four cognition tasks are text-oriented reasoning tasks, such as code reasoning, numerical calculation, and text translation. Consequently, the performance of these cognition tasks is predominantly influenced by which LLM is used, rather than the visual comprehension capabilities of MLLM. Furthermore, our analysis reveals a distinct bias in the text translation task towards Chinese-English translation. While only four examples are shown in Fig. 7, all instances of text translation tasks are observed to be Chinese-English translations. Considering such characteristics, we prioritize the MME with perception tasks (MME^{P}) over cognition tasks for model comparisons.

G. Additional Results

G.1. Detailed Benchmark Scores

We report the detailed scores of our final models for all categories in MME, MMB, SEED^{I}, and LLaVA^{W} in Table 14.

Model	Subject			Context Modality			Grade		Average
	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	
Human [38]	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
GPT-3.5 [38]	75.44	70.87	78.09	74.68	67.43	79.93	78.23	69.68	75.17
GPT-4 [33]	84.06	73.45	87.36	81.87	70.75	90.73	84.69	79.10	82.69
Specialist Models									
LLaMA-Adapter [60]	84.37	88.30	84.36	83.72	80.32	86.90	85.83	84.05	85.19
MM-CoT [62]	**95.91**	82.00	90.82	**95.26**	88.80	92.89	92.44	90.31	91.68
LLaVA [33]	90.36	95.95	88.00	89.49	88.00	90.66	90.93	90.90	90.92
LLaVA+GPT-4 $_{\text{(judge)}}$ [33]	91.56	**96.74**	91.09	90.62	88.99	**93.52**	92.73	92.16	92.53
Generalist Models									
Honeybee (M=256)	93.12	96.63	90.55	92.52	91.77	92.26	93.72	92.22	93.19
Honeybee (M=576)	95.20	96.29	**91.18**	94.48	**93.75**	93.17	**95.04**	**93.21**	**94.39**

Table 15. **Evaluation results on the Science QA test split.** Question classes: NAT = natural science, SOC = social science, LAN = language science, TXT = text context, IMG = image context, NO = no context, G1-6 = grades 1-6, G7-12 = grades 7-12. Despite specialist models being tailored explicitly for the ScienceQA benchmark, *e.g.*, further fine-tuning solely on ScienceQA, Honeybee achieves state-of-the-art scores under a generalist approach. We highlight the **best results** and second-best results in bold and underline.

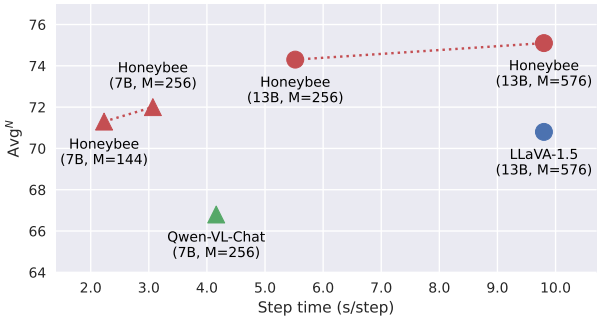


Figure 8. **Comparison between Honeybee variants and current state-of-the-art methods.** Avg^{N} denotes the normalized average score of MMB, MME^{P}, and SEED^{I}.

G.2. Pushing the Limits

Table 7 in the main text shows the performance of Honeybee with the increased number of visual tokens, matching them to the linear projector. Here, we further provide the comparison between the Honeybee variants and the current state-of-the-art methods, namely Qwen-VL-Chat [2] and LLaVA-1.5 [32], in Fig. 8. This figure highlights the efficiency and effectiveness of the proposed Honeybee.

G.3. Science QA

The Science QA dataset [38] is specifically designed to evaluate the broadness of domain knowledge and multi-hop reasoning skills of AI systems, which is essential for MLLMs to perform a wider range of tasks requiring more complex reasoning. Thus, in this section, we additionally provide the evaluation results of the Science QA benchmark. From Table 15, recent MLLMs, *i.e.*, LLaMA-adapter [60], MM-CoT [62], and LLaVA [33], show remarkable performance in this benchmark via further fine-tuning on the Science QA dataset; we refer to these fine-tuned models as *Specialist Models* in Table 15. Especially, in LLaVA+GPT-4 $_{\text{(judge)}}$, they achieved state-of-the-art scores by utilizing the GPT-4 [43] as a judge; whenever GPT-4 and LLaVA produce different answers, they prompt GPT-4 again, asking it to provide a final answer based on the question and two outcomes. Remarkably, Honeybee, with C-Abstractor and vicuna-13B, outperforms the LLaVA+GPT-4 $_{\text{(judge)}}$ and achieves new state-of-the-art scores in this benchmark without the assist of GPT-4 or the task-specific fine-tuning process. These results highlight the effectiveness of our contributions: 1) architectural improvement of the projector and 2) thoroughly explored training recipe.

H. Qualitative Analysis

H.1. Attention Comparison between Resampler and D-Abstractor

As discussed in Section 3.2.1, the resampler tends to primarily focus on salient regions, whereas our proposed abstractors are designed to preserve local contexts effectively. To further validate this, we examined attention maps from both the resampler and the D-Abstractor for their every learnable query (M=144). From Fig. 9, we observe that queries of the resampler only attend to specific salient areas, suggesting the potential loss of detailed information. On the other hand, in the case of D-Abstractor, each learnable query locally abstracts visual features across the whole feature map, which provides fine-grained and comprehensive information about the image. This result indicates that our proposed abstractors benefit in improving the performance of spatial understanding tasks requiring capturing diverse relationships and objects in an image.

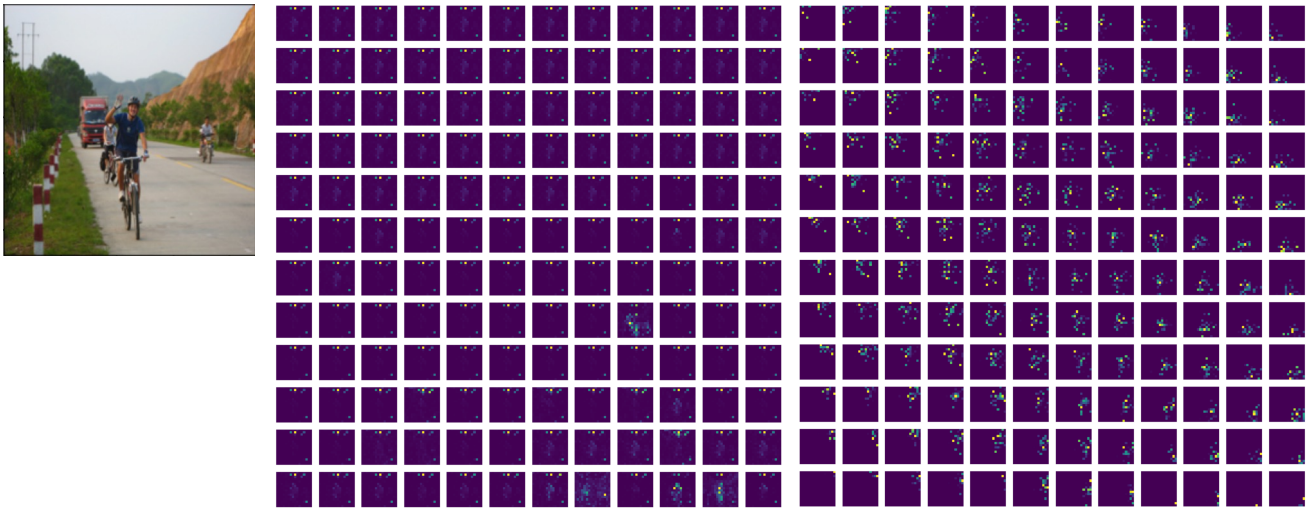


Figure 9. **Visualization of attention maps**. **(Left)** the input image, **(Middle)** the attention map from the resampler, and **(Right)** the attention map from the D-Abstractor. Our locality-aware projector (D-Abstractor) effectively preserves local contexts, while the resampler extracts visual information mainly from a few regions and loses some details.

H.2. Qualitative Examples

In Fig. 10, we present generation results for various user instructions using samples of OwlEval [53]. To verify the importance of instruction-following data for human preference, we report both results generated from two different models: 1) a model trained without instruction-following data such as LLaVA [33] and ShareGPT [10], called Honeybee $_{\text{w/o Inst.}}$ (D9 in Table 4), and 2) a model trained with all available data, including the instruction-following data, called Honeybee $_{\text{w/ Inst.}}$ (D1 in Table 4). From Fig. 10, while the Honeybee $_{\text{w/o Inst.}}$ captures objects included in given images exactly, the answers tend to be short or not follow the instructions. In contrast, Honeybee $_{\text{w/ Inst.}}$ accurately responds to user instructions and generates detailed answers preferred by humans.

User	Why would a person find this image funny?
Honeybee w/o Inst.	Squirrel with camera
Honeybee w/ Inst.	A person might find this image funny because it features a squirrel, which is a small, cute, and often mischievous creature, holding a camera and seemingly posing for a photo. The idea of a rodent holding human-like objects, such as a Camera, can be amusing and whimsical. Furthermore, the squiggly-eyed squirming squid expression adds to the humor by making the scene even more exaggerated and light-hearted. People often find anthropomorphic representations of animals and unexpected situations like this entertaining.

User	What is unusual about this image?
Honeybee w/o Inst.	Man drying clothes on car
Honeybee w/ Inst.	In the image, a man is ironing clothes while standing on the back of a car parked on a street. This is unusual because iron is not typically used in outdoor settings, and it is even more uncommon to see someone iron clothes on top of an automobile. Furthermore, the presence of the man iron and the car with its hood open create a unique and unexpected scene that captures the attention of passersby, as they may not commonly witness such an unusual event.

User	How many championship rings did the player on the left win in his career?
Honeybee w/o Inst.	5
Honeybee w/ Inst.	The player on the left, wearing a yellow jersey, is Kobe Bryant. Kobi Bryant won five championship rings in his career, all of which were won with the Los Angeles Lakers.

User	Is this image common in real world?
Honeybee w/o Inst.	No
Honeybee w/ Inst.	The image depicts an unusual and surreal scene of a cactus surrounded by ice cubes. This is not a common sight in the real world, as cacti and ice are generally found in different environments and climates. Cacti typically grow in hot and arid desert regions, while ice is formed in colder temperatures and environments. The combination of the two elements in this image is likely a result of photo manipulation or staging, creating an interesting and unconventional visual effect.

User	Write a conversation between the two animals.
Honeybee w/o Inst.	Cat on dog
Honeybee w/ Inst.	In the image, a cat is sitting on the back of a dog that is lying in the snow. It is an unusual and amusing scene. If the cat and the dog could communicate, they might have a conversation something like this: Cat: purrs "Hey there, big guy. Thanks for letting me ride you." Dog: wag wags "You're welcome, feline friend. I love having you on my back."

User	Help me write an advertisement for this.
Honeybee w/o Inst.	Bring home a piece of art with this gorgeous golden bird lamp
Honeybee w/ Inst.	Introducing our unique and whimsical bird-shaped table lamp. This charming piece features a beautifully crafted bird with a large, long beak, perched on one foot and holding a light bulb in its beak. The lamp's design adds a touch of creativity and playfulness to any space, making it an ideal addition to a living room, bedroom, or office. With its eye-catching appearance, this bird lamp will spark conversation and bring a sense of joy and charm to your room. Place it on a table or desk to create a cozy ambiance with its warm glow. Order yours today!

Figure 10. **Qualitative examples** with various user inputs. **w/ Inst.** and **w/o Inst.** indicate results from models trained with or without instruction-following data, *i.e.*, LLaVA [33] and ShareGPT [10], respectively. The example images are selected from OwlEval [53].