Projector	M	s/step	MMB	SEED$^{\text{I}}$	MME$^{\text{P}}$	Avg$^{\text{N}}$
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.

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) [33] 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-Abstractor and D-Abstractor achieve better or comparable benchmark scores while using fewer visual tokens, indicating our projectors' 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 projectors, C-Abstractor and D-Abstractor.

Architectures	MMB	SEED$^{\text{I}}$	MME$^{\text{P}}$	Avg$^{\text{N}}$
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.

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 [51] with squeeze-excitation [19], 2) ConvNext block [37], 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 [67] 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 [67] 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.

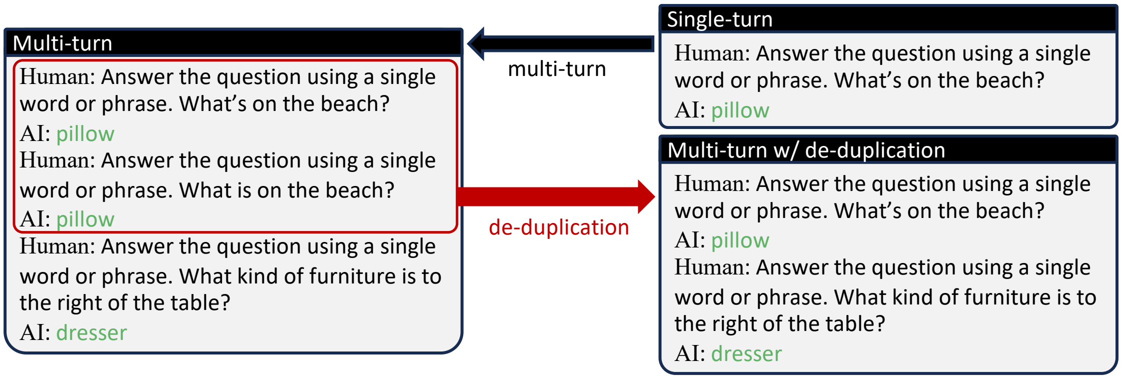
C. Implementation Details

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 × 8, employing deepspeed zero-2 [47] and flash-attention v2 [12] for all experiments, except for the pre-training of long schedule where we use multi-node 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	Human: 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) [33]. {*} is replaced depending on dataset examples where red-colored one means a target output. Note that *bbox* is expressed as normalized coordinates $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$.



Multi-turn:

Human: Answer the question using a single word or phrase. What's on the beach?
AI: pillow
Human: Answer the question using a single word or phrase. What is on the beach?
AI: pillow
Human: Answer the question using a single word or phrase. What kind of furniture is to the right of the table?
AI: dresser

Single-turn:

Human: Answer the question using a single word or phrase. What's on the beach?
AI: pillow

Multi-turn w/ de-duplication:

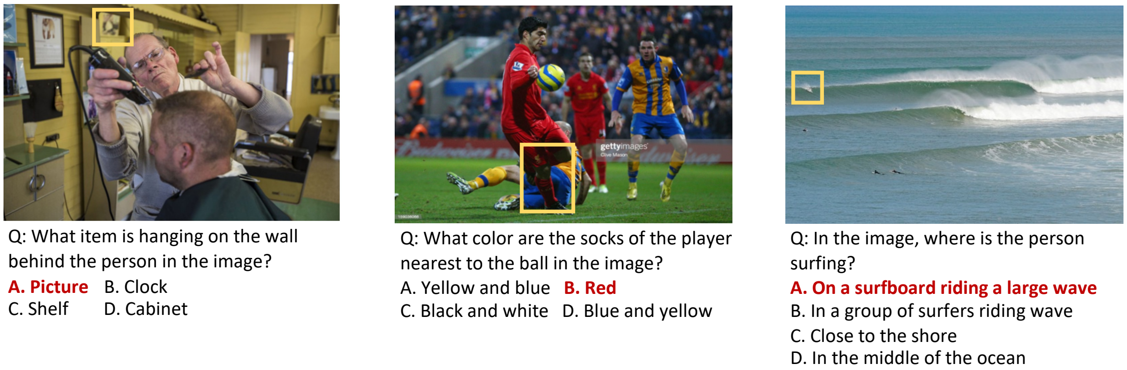
Human: Answer the question using a single word or phrase. What's on the beach?
AI: pillow
Human: Answer the question using a single word or phrase. What kind of furniture is to the right of the table?
AI: dresser

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, 54], 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 [46]. 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-

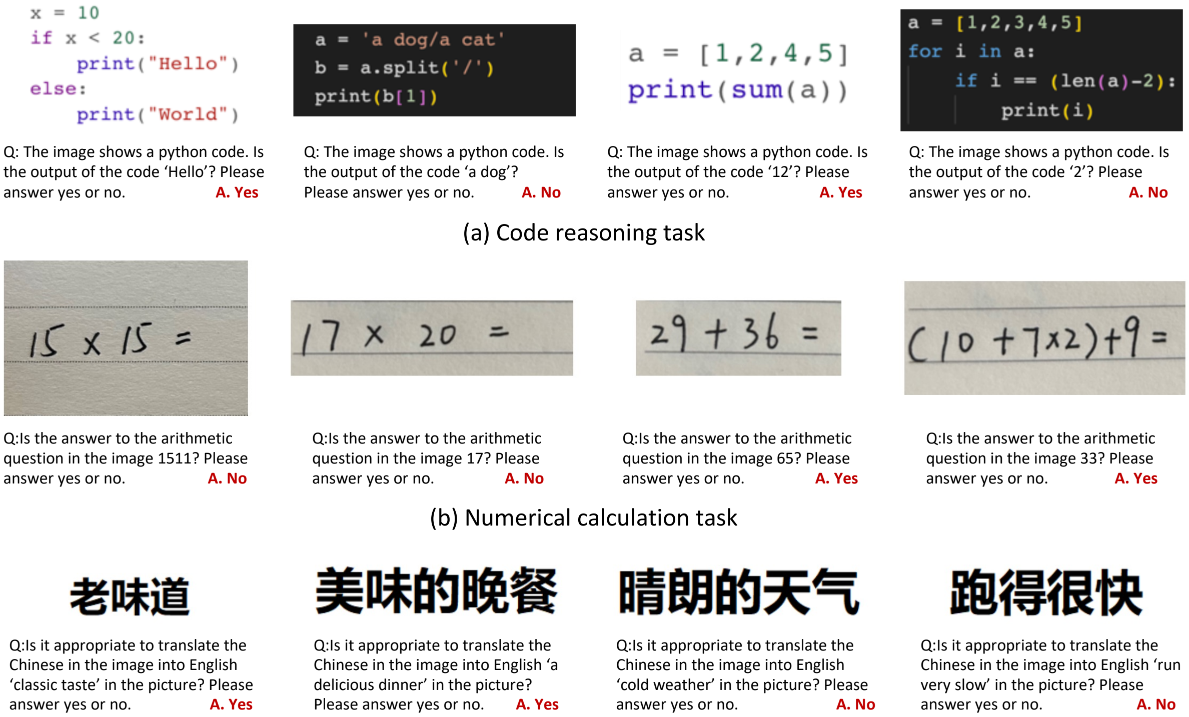


Q: What item is hanging on the wall behind the person in the image?
A. Picture B. Clock
C. Shelf D. Cabinet

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).



```
x = 10
if x < 20:
    print("Hello")
else:
    print("World")
```

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

```
a = 'a dog/a cat'
b = a.split('/')
print(b[1])
```

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

```
a = [1,2,4,5]
print(sum(a))
```

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

```
a = [1,2,3,4,5]
for i in a:
    if i == (len(a)-2):
        print(i)
```

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

15 x 15 =

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

17 x 20 =

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

29 + 36 =

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

(10 + 7x2)+9 =

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

(b) Numerical calculation task

老味道

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

美味的晚餐

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

晴朗的天气

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

跑得很快

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 [39]	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
GPT-3.5 [39]	75.44	70.87	78.09	74.68	67.43	79.93	78.23	69.68	75.17
GPT-4 [34]	84.06	73.45	87.36	81.87	70.75	90.73	84.69	79.10	82.69
Specialist Models									
LLaMA-Adapter [62]	84.37	88.30	84.36	83.72	80.32	86.90	85.83	84.05	85.19
MM-CoT [64]	**95.91**	82.00	90.82	**95.26**	88.80	92.89	92.44	90.31	91.68
LLaVA [34]	90.36	95.95	88.00	89.49	88.00	90.66	90.93	90.90	90.92
LLaVA+GPT-4 $_{\text{(judge)}}$ [34]	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.

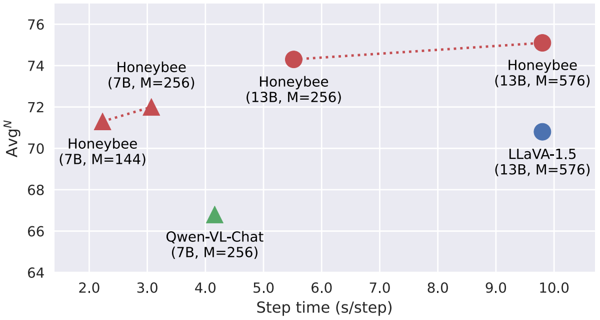


Figure 8. **Comparison between Honeybee variants and current state-of-the-art methods.** Avg^{N} denotes the normalized average score of MMB, MME^{P}, and SEED^{I}.

	MM-Vet	MMMU	POPE
Approaches using 7B LLM			
LLaVA (v1)	23.8	-	66.5
MiniGPT-4	22.1	-	-
LLaMA-AdapterV2	31.4	29.8	-
mPLUG-Owl	-	-	67.4
InstructBLIP	26.2	-	-
Qwen-VL-Chat	-	35.9	-
LLaVA-1.5	30.5	-	**85.9**
Honeybee (C-7B-144M)	34.9	35.3	83.2
Honeybee (C-7B-256M)	**35.6**	**36.4**	84.3
Approaches using 13B LLM			
MiniGPT-4	24.4	26.8	74.5
BLIP-2	22.4	35.4	85.3
InstructBLIP	25.6	35.7	83.8
LLaVA-1.5	35.4	36.4	**85.9**
Honeybee (C-13B-256M)	38.1	**37.3**	85.5
Honeybee (C-13B-576M)	**42.2**	36.2	85.6

Table 16. **Additional benchmark comparison.** Numbers are collected from each paper and official leaderboard, selecting the best results for each method when multiple exist.

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 [33], in Fig. 8. This figure highlights the efficiency and effectiveness of the proposed Honeybee.

G.3. Science QA

The Science QA dataset [39] 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 [62], MM-CoT [64], and LLaVA [34], 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 [44] 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.

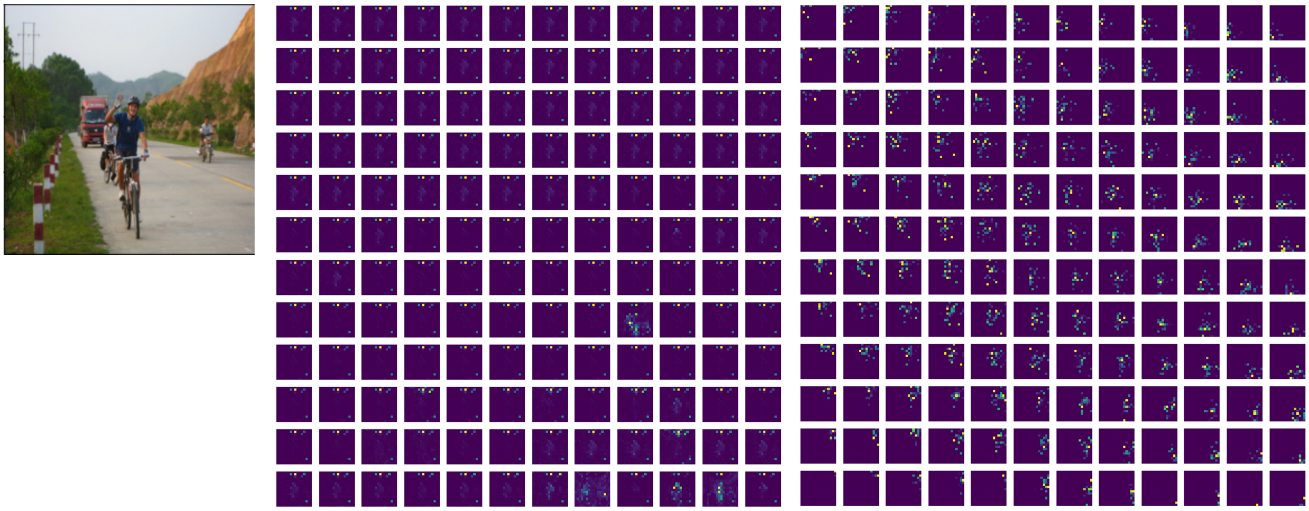


Figure 9. **Visualization of attention maps**. **(Left)** the input image, **(Middle)** the attention map from the resampler, and **(Right)** the attention map from D-Abstractor. Our locality-aware projector (D-Abstractor) effectively preserves local contexts, while the resampler extracts visual information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

G.4. Additional Benchmark Results

In addition to four benchmarks used in the main paper, we perform further evaluation using three additional benchmarks—1) MM-Vet [58] and MMMU [60] for visual understanding capability evaluation, and 2) POPE [30] for evaluation of object hallucination. In Table 16, similar to the experimental results in the main text, Honeybee shows superior comprehensive visual understanding on MM-Vet and MMMU benchmarks. On the other hand, in the hallucination aspect, the performance of 7B-scale Honeybee slightly falls short compared to LLaVA-1.5. However, when using larger images (336 resolution) with 13B LLM, it achieves competitive performance, suggesting the importance of higher visual understanding and reasoning to tackle hallucination issue.

H. Qualitative Analysis

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

As discussed in Section 3.2.1, the vanilla abstractor (*i.e.*, resampler) tends to primarily focus on salient regions, whereas our locality-enhanced abstractor (*i.e.*, D-Abstractor) is 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 projectors benefit in improving the performance of spatial understanding tasks that require to capture diverse relationships and objects in an image.

H.2. Qualitative Examples

In Fig. 10, we present generation results for various user instructions using samples of OwlEval [54]. 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 [34] 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 [34] and ShareGPT [10], respectively. The example images are selected from OwlEval [54].