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 [34] 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 [56, 59]. Such success leads to increasing attention for research into MLLMs taking multimodal inputs (*e.g.*, videos [28], audio [13], 3d world [17], point cloud [52]) beyond text.

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, 34] or abstractors [2, 11, 27, 54, 66].

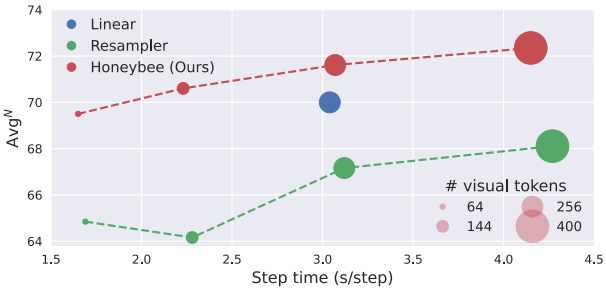

Figure 1. **Performance *vs.* efficiency for projectors** where Avg^{N} means an average of normalized benchmark scores (MME, MM-Bench, and SEED-Bench) and step time is a single step execution time during pre-training. Honeybee with the locality-enhanced projector (*i.e.*, C-Abstractor) offers a more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

	MMB	SEED^{I}	MME^{P}	MME	LLaVA^{W}
Previous SoTA	67.7 [33]	68.1 [33]	1531 [33]	1848 [2]	70.7 [33]
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.

*Equal contribution

Notably, recent MLLMs prefer abstractors (*e.g.*, resampler, Q-former) to linear projectors; 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 performance. However, as shown in Fig. 3, the abstractors face more difficulties in learning spatial understanding tasks compared to the linear projectors. This difficulty stems from the abstraction process lacking a locality-aware design, which causes it to primarily focus on a few regions, leading to a loss of finer details essential for spatial comprehension. In contrast, linear projectors excel at preserving all local contexts 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 projectors, 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 projectors by employing two powerful operations in locality modeling—convolution and deformable attention. Such injection of locality-aware design into the abstraction process 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 [34] 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 crucial properties of projector, 1) locality preservation of visual features and 2) flexibility to manage the number of visual tokens, and propose locality-enhanced abstractors to 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, 61] and enhancing visual instruction-following data [32, 34, 40, 63, 65, 66]. Also, several studies focus on grounding capabilities of MLLMs by utilizing additional datasets specifically designed for these tasks [7, 45, 53, 55]. 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 [43] 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 [44], to generate visual instruction data for complex VL tasks, such as generating stories, poems, detailed captions from given images [32, 34, 63, 65, 66]. Another line of studies has explored transforming existing VL task datasets into an instruction-following format using pre-defined templates, called *instructization* [2, 11, 33, 61]. 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 [35], 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, 36], several studies leverage powerful LLMs, *e.g.*, GPT-4 [44], to assess the response quality of MLLMs [3, 34, 58]. This approach aims for a more detailed evaluation of the profi-

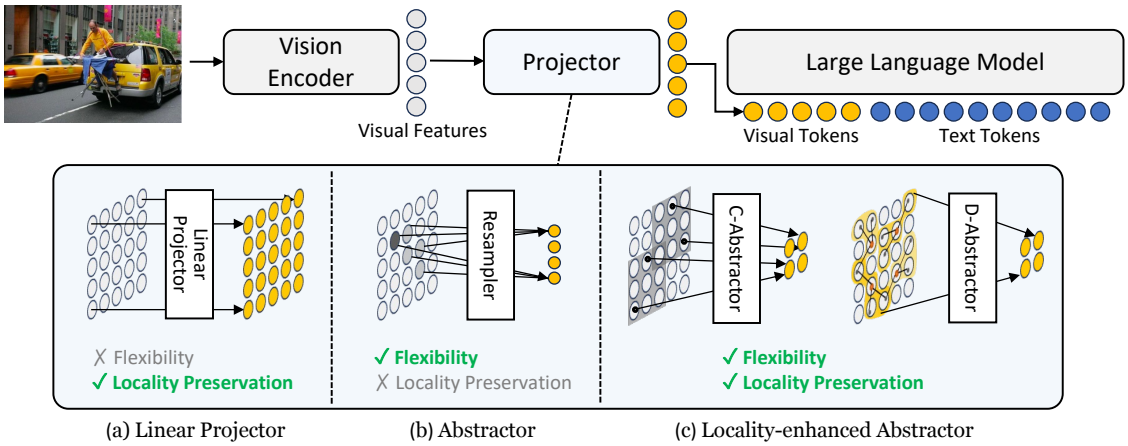

Figure 2. **Conceptual comparison between projectors** in terms of how to convert visual features into visual tokens. (a) Linear projector performs a one-to-one transformation, thus effective in preserving all local contexts of visual features, but limited in flexibility. (b) Abstractor such as resampler 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.

ciency 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 an additional modality input 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, 54, 66]. 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. Considering its role, when designing a projector, the most impor-

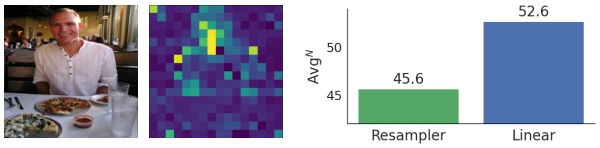

Figure 3. (Left) an example of an attention map from the resampler and (Right) a comparison of spatial understanding capability for the resampler and linear projector where Avg^{N} is computed using six spatial understanding tasks from MME, MMB, and SEED^{I}.

tant 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, 54].

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, 49, 51]. Thus, we design Convolutional Abstractor, C-Abstractor, for effective local context modeling. Fig. 4a depicts the entire architecture, comprising L ResNet blocks [51] 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 [37, 49] in Appendix B, but ResNet [51] shows the best performance.

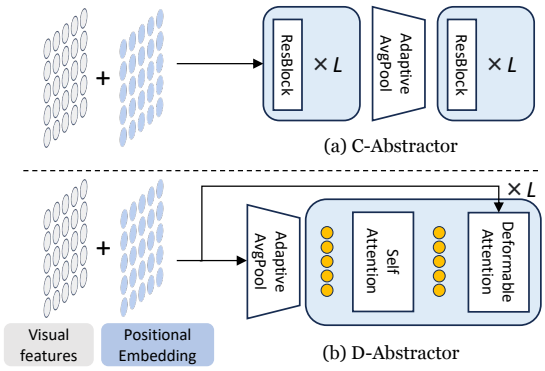

Figure 4. Conceptual architecture of our proposed projectors.

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 [67] 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 projector. 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 [42], VSR [31]	2.2M
VQA (MC)	ScienceQA [39], A-OKVQA [48]	0.03M
REC	RefCOCO [21], RefCOCO+ [57], RefCOCOg [41], VG [23]	5.7M
Instruction	LLaVA150K [34], 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 [34] and ShareGPT [10]. In addition, to enhance visual understanding, we instructize a wide range of existing datasets using templates, as listed in Table 2. Specifically, our approach includes: 1) employing a range of tasks such as open-ended VQA [16, 20, 31, 42], multiple-choice VQA [39, 48], captioning [5, 26], and referring expression comprehension (visual grounding and grounded captioning) [21, 23, 41, 57]; 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 instructization 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 instruction tuning using existing datasets with the template-based instructization is beneficial [2, 11, 33], 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 following 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, 33, 61]. 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 [50], 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, 33, 50, 61], 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 [50], and 2) *coarse-grained*: applying the shared templates across datasets within the same task category [11, 33].

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, 38, 61]. However, the introduction of GPT-assisted datasets has seemingly diminished the emphasis on the diversity of templates [33]. 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. Settings

Benchmarks. We adopt four benchmarks specifically designed for Multimodal LLM (MLLM) evaluation, including MME [14], MMBench [35], SEED-Bench [25] and LLaVA-Bench (In-the-Wild) [34]. 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.

Implementation details. We use 7B and 13B Vicuna-v1.5 [10] as LLM. We leverage the pre-trained CLIP ViT-L/14 [46] with 224 and 336 resolutions for 7B- and 13B-LLM, respectively; we use 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. For in-depth ablations, we use a short training schedule (50k pre-training, 4k instruction tuning) with Vicuna-7B, CLIP ViT-L/14, and C-Abstractor with M=144 visual tokens unless stated otherwise. For the final models, we adopt a long training schedule (200k pre-training, 10k instruction tuning). More details are in Appendix C.

5.2. Analysis on Locality-Enhanced Projector

To showcase the value of the proposed projector, we assess and compare both performance and efficiency against existing projectors in Table 3 using six spatial understanding tasks from MME, MMBench, and SEED-Bench. First, Resampler (B2, B5) shows poor performance due to its lack of consideration for local context preservation, despite being flexible to the number of visual tokens M. Second, Linear projector is limited to M=256 (B4) due to its inflexibility (B1), leading to intractable computational costs in high-resolution of larger M. Third, for the same computational budget (M=256), our C-Abstractor offer significantly improved performance compared to linear one (52.6 (B4) *vs.* 56.3 (B6)). Lastly, with fewer visual tokens (M=144), our C-Abstractor demonstrate improved performance (+0.9 point) and greater efficiency (3.04 (B4) *vs.* 2.23 (B3) s/step). This improvement suggests our locality-enhanced projector excels at abstracting visual features where it integrates local contexts from neighboring features and provides context-enriched visual tokens, thus enabling our projectors to outperform linear counterparts even with fewer visual tokens.

	Projector	M	s/step	MME	MMB			SEED		Avg^{N}
				POS	SR	OL	PR	SR	IL	
B1	Linear	144	-	*Unavailable due to inflexiblity*						-
B2	Resampler	144	2.28	75.0	22.2	43.2	62.5	47.5	50.6	43.9
B3	C-Abstractor	144	2.23	135.0	24.4	54.3	66.7	49.0	58.8	53.5
B4	Linear	256	3.04	140.0	24.4	40.7	70.8	48.9	60.9	52.6
B5	Resampler	256	3.12	73.3	24.4	37.0	79.2	44.4	51.8	45.6
B6	C-Abstractor	256	3.07	136.7	26.7	55.6	75.0	52.7	59.3	56.3

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. M means the number of visual tokens and s/step indicates the execution time for a single step during pre-training.

5.3. Hidden Recipe for Visual Instruction Tuning

Dataset combination. Table 4 shows a comprehensive ablation study to identify the individual impact of datasets on various multimodal benchmarks. First, we investigate *the impact of dataset diversity within each task* by leveraging only a single dataset for each task group (D1 *vs.* D2). The overall performance drop highlights the importance of the dataset diversity within each task. Second, we explore *the impact of each task* by sequentially excluding specific tasks (D1 *vs.* D3-8). This reveals that task diversity is crucial for learning how to handle a variety of tasks; each task improves the performance of relevant benchmarks, VQA (Open) $\rightarrow$ MME, VQA (MC) $\rightarrow$ MMB and SEED^{I}, and captioning and instruction-following data $\rightarrow$ LLaVA^{W}. Third, we inspect *the impact of using existing vision-language data* (D9 *vs.* D10). Excluding such data leads to significant decreases in MME, MMB and SEED^{I} benchmarks. This suggests that rich knowledge in existing vision-language datasets enhances MLLM's perception understanding or visual reasoning capabilities. In summary, these experiments emphasize the importance of diversity in both tasks and datasets within each task.

Dataset balancing. The necessity of hand-crafted dataset balancing is addressed in previous studies [11, 38]. Based 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.

	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.

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, 50], 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 [33], 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 datasets 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.

Final recipe. In summary, our *final recipe* is summarized as 1) adopting flexible, locality-preserving C-Abstractor or D-Abstractor; 2) leveraging diverse datasets for various tasks (Table 4); 3) applying selected ablation options in Table 5 and Appendix D—the application of per-dataset balancing with hand-crafted tuning, fine-grained templates, and multi-turn interactions with deduplication.

Method	LLM	Projector	Vision Encoder	Res.	MMB	MME^{P}	MME	SEED^{I}	LLaVA^{W}
Approaches using 7B LLM									
LLaVA (v1) [34]	LLaMA-7B	Linear	CLIP ViT-L/14	224	38.7	502.8	717.5	33.5	-
MiniGPT-4 [66]	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 [54]	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 [33]	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 [66]	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 [33]	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.

5.4. Putting It Altogether

Comparison with existing MLLMs. In Table 6, we compare our Honeybee, trained using the final recipe and a long training schedule, 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 [33] use larger vision encoders (*e.g.*, ViT-bigG for Qwen-VL) or larger images (448 and 336) with more visual tokens (M=256 and 576). In contrast, Honeybee employs ViT-L/14 with 224 resolution and 144 visual tokens striking a balance between performance and efficiency (Figure 8). For tasks requiring detailed visual understanding, such as SEED^{I} (see Appendix F), using larger images or more visual tokens can be beneficial. When the number of visual tokens is increased from 144 to 256, Honeybee achieves the best score in SEED^{I} (65.5) among 7B-scale LLMs, as shown in Table 7. When scaled up to 13B, Honeybee surpasses all previous methods in every benchmark. The detailed scores are available in Appendix G.1.

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 that of linear projector (Table 7), show performance enhancement at the cost of efficiency. Additional comparisons with previous methods are in Appendix G.2.

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* is pre-training step time.

Additional results. We additionally present (i) the detailed scores for MME, MMB, SEED^{I}, and LLaVA^{W} in Appendix G.1, (ii) ScienceQA [39] results in Appendix G.3, (iii) additional benchmark (MM-Vet [58], MMMU [60], POPE [30]) results in Appendix G.4, and (iv) qualitative examples 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. Inspired by this, we identify the desirable but overlooked projector property, *i.e.*, locality preservation, and propose the locality-enhanced projector that offers a preferable performance-efficiency balance. 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, Honeybee remarkably outperforms previous state-of-the-art methods on 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. 2

[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] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *EMNLP*, 2023. 8, 18

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

[32] 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

[33] 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

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

[35] 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

[36] 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

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

[38] 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

[39] 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

[40] 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

[41] 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

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

[43] OpenAI. ChatGPT, 2023. 2

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

[45] 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

[46] 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

[47] 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

[48] 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

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

[50] 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

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

[52] 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

[53] 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

[54] 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

[55] 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

[56] 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

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

[58] 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. 2, 8, 18

[59] 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

[60] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023. 8, 18

[61] 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

[62] 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

[63] 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

[64] 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. 17

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

[66] 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

[67] 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