Accelerating Pre-training of Multimodal LLMs via Chain-of-Sight

Ziyuan Huang[1] **Kaixiang Ji**[1] **Biao Gong**[1] **Zhiwu Qing**[2] **Qinglong Zhang**[1]
Kecheng Zheng[1] **Jian Wang**[1] **Jingdong Chen**[1] **Ming Yang**[1]
[1]Ant Group [2] Huazhong University of Science and Technology
https://chain-of-sight.github.io/

Abstract

This paper introduces Chain-of-Sight, a vision-language bridge module that accelerates the pre-training of Multimodal Large Language Models (MLLMs). Our approach employs a sequence of visual resamplers that capture visual details at various spacial scales. This architecture not only leverages global and local visual contexts effectively, but also facilitates the flexible extension of visual tokens through a compound token scaling strategy, allowing up to a 16× increase in the token count post pre-training. Consequently, Chain-of-Sight requires significantly fewer visual tokens in the pre-training phase compared to the fine-tuning phase. This intentional reduction of visual tokens during pre-training notably accelerates the pre-training process, cutting down the wall-clock training time by ∼**73%**. Empirical results on a series of vision-language benchmarks reveal that the pre-train acceleration through Chain-of-Sight is achieved without sacrificing performance, matching or surpassing the standard pipeline of utilizing all visual tokens throughout the entire training process. Further scaling up the number of visual tokens for pre-training leads to stronger performances, competitive to existing approaches in a series of benchmarks.

1 Introduction

Recently, Large Language Models [70, 6, 80, 5, 3] have received unprecedented attention, owing to their remarkable capabilities in text comprehension and generation. Riding on the success of LLMs, Multimodal Large Language Models (MLLMs) [74, 90, 63, 56, 26, 88, 73] demonstrate impressive zero-shot transferability across a wide range of vision-language tasks, such as image captioning, visual question answering, and visual grounding.

The exceptional generalization ability exhibited by the contemporary MLLMs can be largely attributed to their extensive pre-training on a massive amount of data [16, 14, 69, 31, 18]. However, as the volume of data escalates, so does the wall-clock training time, which has become a major obstackle in further explorations. According to [63], 60,000 GPU hours are needed for training a 7B model on just 96 million image-text pairs. This intensive computational demand is not only prohibitive to many researchers, but also leads to a significant carbon footprint.

One of the key reasons for the prolonged training time is the extensive length of visual tokens Typically, the image-text pairs in the pre-training phase involve around 23 text tokens (see Table 1). In contrast, most MLLMs handle substantially more visual tokens during pre-training, *e.g.*, 144 [10, 11], 256 [4, 50, 97], or even higher [63, 16, 55, 56, 64, 48]. Reducing the number of visual tokens presents a straightforward way to speed up training, as it allows for an increase in batch size and a concurrent decrease in step time. Meanwhile, the reduced memory consumption allows for better optimization stages [81], further reducing time requirements. However, training with fewer visual tokens often results in compromised performance for existing vision-language models.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

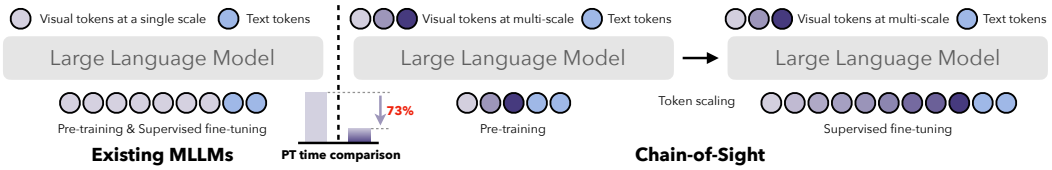

Figure 1: **Chain-of-Sight concept overview**. Recent current MLLMs maintain a constant set of visual tokens in both pre-training and fine-tuning. These tokens typically represent visual contents at a single visual scale. In contrast, our Chain-of-Sight approach leverages the idea of visual hierarchy, producing multi-scale visual tokens. Moreover, the token scaling strategy enabled by our multi-scale visual resamplers allow us to start with a small pool of visual tokens for pre-training, before increasing the number of tokens during fine-tuning. This considerably accelerates the pre-training phase.

To resolve this dilemma, this work introduces Chain-of-Sight, a vision-language bridging module for efficient pre-training of MLLMs. Unlike existing approaches that maintain a constant token count throughout both pre-training and fine-tuning, Chain-of-Sight allows for a marked increase in the number of tokens after the pre-training stage, thereby reducing the tokens needed during pre-training. The core mechanism is our multi-scale visual resampler, which produces visual tokens of multiple visual scales. Inspired by the classical concept of multi-scale feature hierarchy in visual understanding [106, 42, 33, 107, 83, 53], we partition the visual features produced by the visual backbone using windows of multiple sizes. For each window size, a visual resampler is implemented to produce a specified number of visual tokens per window. Subsequently, the visual tokens from various window sizes are gathered and linked in a global-to-local manner, forming a chain of reasoning steps from coarse views gradually to fine-grained perspectives.

On top of this, we propose a post-pretrain token scaling strategy, which compounds the elements of input resolution and window size manipulation to enable a significant escalation in the token count for our Chain-of-Sight, reaching up to 16× increase during fine-tuning. Such adaptability allows for the fine-tuning of the model with a flexible granularity or complexity as required, without the the necessity for an additional pre-training phase.

By intentionally reducing the number of visual token by ∼90% in the pre-training, a 2.5× batch size is allowed with a step time reduction of 30%, leading to a 3.7× faster pre-training in terms of wall-clock time (∼73% less) for the same amount of data, when compared with using all visual tokens during pre-training. Meanwhile, our observations indicate that this acceleration does not come at the expense of performance. The results achieved by our Chain-of-Sight model pre-trained with 32 tokens match or surpass those obtained using 336 visual tokens throughout the training process, when both models use the same tokens during fine-tuning. Further scaling up the tokens in the fine-tuning stage leads to enhanced performance at small additional training costs. This scaling showcases the potential of Chain-of-Sight to capitalize on the initial efficiency gains and adapt its framework to achieve even greater levels of accuracy and effectiveness in visual understanding for MLLMs.

2 Method

Our objective is to accelerate the pre-training of MLLMs. To this end, we resort to reducing the number of visual tokens inputted into the language model. To mitigate the performance drop associated with fewer visual tokens, we introduce a versatile bridge module within our framework, named Chain-of-Sight. This module is designed to enable the increase in the token count on demand after pre-training. With this capability, we are able to substantially lower the number of visual tokens during the pre-training phase, while retaining the ability to scale up and capture a rich level of visual detail during fine-tuning. The concept of Chain-of-Sight is illustrated in Fig. 1.

2.1 Re-examining the efficiency bottleneck in MLLM pre-training

Modern MLLMs are typically constructed by three core components: (1) a visual encoder, (2) a vision-language bridge module, and (3) a language model. Given that the language models often have a much larger size than the visual encoder, they account for the majority of computation during pre-training [90, 4, 73, 19, 58]. Consequently, the number of input tokens processed by the language model is a crucial factor determining the total computational workload.

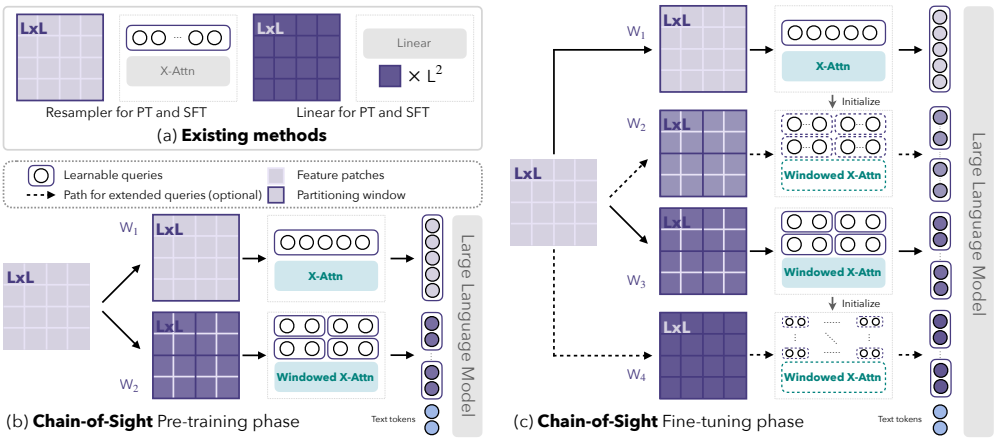

Figure 2: **The Chain-of-Sight framework.** Through partitioning visual features into windows and restricting cross-attention to the windowed features associated with the learnable tokens, our Chain-of-Sight approach produces visual tokens that encompass multiple scales. Thanks to the post-pretrain token scaling strategy, Chain-of-Sight reduces the required number of visual tokens in pre-training, thus accelerating the process. In contrast, the number of visual tokens remains constant in resampler-based methods [44, 2, 4, 99] for pre-training and fine-tuning, and the linear-layer [56, 63, 97, 15] produce a large number of visual tokens, incurring a high cost for pre-training.

As detailed in Table 1, the pre-training data predominantly comprise image-text pairs that contain fewer than 50 text tokens. In contrast, existing MLLMs are designed to handle 2× more visual tokens, often requiring such as 144 [10, 11], 256 [4, 50, 97], or even more visual tokens [63, 16, 55, 56, 64, 48]. The imbalance between the visual tokens and text tokens means that processing these visual tokens has become the main efficiency bottleneck in MLLM pre-training. This prompts our exploration for more efficient vision-language bridging structures, which is capable of reducing the number of visual tokens in pre-training without compromising performance.

2.2 Multi-scale visual resamplers

The multi-scale visual resamplers serve as the foundational mechanism enabling the flexible extension of visual tokens after the pre-training phase. This subsection focuses on the architectural details of the multi-scale visual resamplers, as visualized in Fig. 2(b), while the the extension of visual tokens is discussed in the subsequent subsection.

Essentially, the idea of exploiting multi-scale or pyramid structures to handle natural hierarchy of visual contents has been long established as a standard practice [30, 42], proving effective in countless visual tasks [33, 61, 83, 53]. Despite this, the potential for harnessing multi-scale visual hierarchies remains under-explored in the context of MLLMs.

Visual resampler. Visual resampler is a Perceiver [36]-like structure that introduces a set of learnable queries and uses cross-attention to condense visual knowledge into a predetermined set of visual tokens [4, 2, 95, 99]. We construct Chain-of-Sight with visual resamplers due to their flexibility in selecting the token count for a specified feature, independent of the features' dimensionality.

Multi-scale visual resamplers. One of the effective strategies for building multi-scale features within a network involves combining operations that spans diverse fields of views [13, 45, 101, 21]. Given that the resampler structure inherently gathers visual cues on a global scale across the entire feature map, our strategy focuses on enhancing the perception of the fine-details in the image.

To this end, we partition the visual features into non-overlapping local windows of various sizes. More precisely, given a visual feature $\mathbf{X} \in \mathbb{R}^{L \times L \times C}$ extracted by the visual encoder, where L and C denote the feature size and channel, respectively, we define a set of window sizes, denoted as $\mathbf{W} = [W_1, ..., W_m]$. This setup leads to a collection of windowed visual features $\mathbf{X}_{\text{win}} = [\mathbf{X}_1, ..., \mathbf{X}_m]$. Each $\mathbf{X}_i$ represent a set of L^2/W_i^2 windowed features obtained by applying the partition operation on the original visual feature maps with a corresponding window size W_i. This naturally forms features of multiple visual scales.

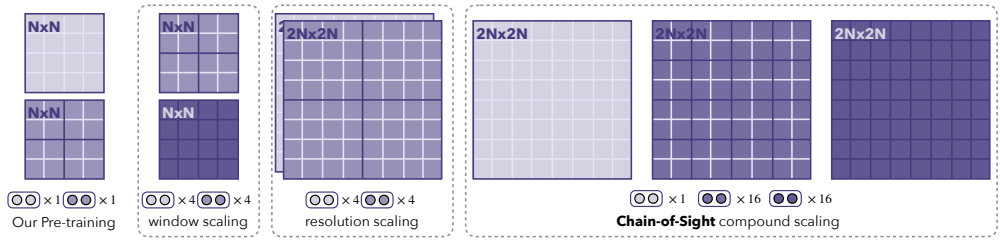

Figure 3: **Detailed illustration of our post-pretrain token scaling strategy.**

At every scale level, each windowed feature is allocated with N_i learnable queries. These learnable queries are then utilized within the visual resampler to perform cross-attention solely on their corresponding windowed feature. This yields N tokens, where the N can be calculated as follows:

$$N = \sum_i L^2/W_i^2 * N_i. \tag{1}$$

The learnable queries within the same scale share the parameters of the visual resampler, despite their different spatial locations. However, because the queries at various scales are intended to capture features from varying fields of view, distinct sets of parameters are used for each scale. This results in a group of visual resamplers operating across multiple scales. On top of this, we enable the resamplers to aggregate visual features from multiple feature levels, as in [8] (see Appendix for details).

Coarse-to-fine integration. Upon acquiring a series of multi-scale visual tokens from the multi-scale visual resamplers, our method integrates these prompts in a structured coarse-to-fine fashion. The final token sequence fed to the language model begins with tokens derived from larger windows, which presents an overall view of the image, and proceeds with tokens obtained from smaller windows that contains fine-grained details. Our priliminary experiments reveal a substantial difference in the overall performance between the coarse-to-fine and the reversed order.

2.3 Post-pretrain token scaling strategy

Reducing the number of visual tokens can effectively accelerate pre-training, but typically at the expense of performance. To address this dilemma, we enhance the token count after pre-training, which allows accelerated pre-training with fewer visual tokens, while a subsequent increase in tokens ensures the final performance after fine-tuning, as demonstrated in Fig. 2(c). Specifically, based on the multi-scale visual resamplers, the increase in the token count is accomplished via our compound token scaling strategy that integrates two core mechanisms: resolution scaling and window scaling.

Resolution scaling. Enhancing the input resolution stands as the most direct way to augment the number of visual tokens. At the cost of additional computation overhead in the visual backbone, it allows for a quadratic rise of the token count with the resolution enhancement. The concept of resolution scaling is investigated in many existing approaches based on linear projectors [55, 69, 26] or visual resamplers [50]. They can broadly be viewed as particular instances within our Chain-of-Sight framework, which regards the window size as a fixed factor. In this context, linear projectors use the smallest possible window size for visual token generation, whereas visual resamplers employ the resolution in the pre-training phase as their window size.

Window scaling. The windowing mechanism in our multi-scale visual resamplers enable scaling up token numbers by manipulating the window sizes. As in Eq. 1, reducing the window sizes can further produce a quadratic increase in the number of visual tokens on top of the resolution enhancement.

Compound scaling. Combining the above token scaling strategies, our compound scaling is capable of producing a 16× increase in the tokens during fine-tuning, as in Fig. 3. This allows us to fine-tune the scale at which visual features are represented and sampled, improving the model's capability of leveraging varying levels of detail and abstraction inherent in the visual content. Consequently, the Chain-of-Sight framework significantly boosts the visual comprehension capability of the model during the fine-tuning stage, effectively compensating for the performance drop incurred by the low number of visual tokens during pre-training.

Initialization. Inspired by [9], we initialize the parameters of the newly introduced visual resamplers by simply inflating the pre-trained parameters, as in Fig. 2. As for the new visual queries, we apply a nearest neighbor strategy to initialize them based on the pre-trained queries.

3 Experiments

In this section, we provide our experimental setup, empirical results, and the comparisons with existing methods.

3.1 Experimental setup

Model details. We instantiate our MLLM with CLIP-ViT-L/14 [78] as the visual encoder and Vicuna [20] as the language model. For efficiency, we adapt Vicuna with LoRA [34] during all training stages, instead of fully fine-tuning the language model. For the number of visual tokens, we experimented with 32, 48, and 80 during pre-training for our Chain-of-Sight model, where 16 tokens are global tokens (with a window size of 16 for an input resolution of 224) and the rest are local tokens (with a window size of 4 by default). These models are configured to be extended to at most 528, 784, and 1296 visual tokens during fine-tuning using compound token scaling.

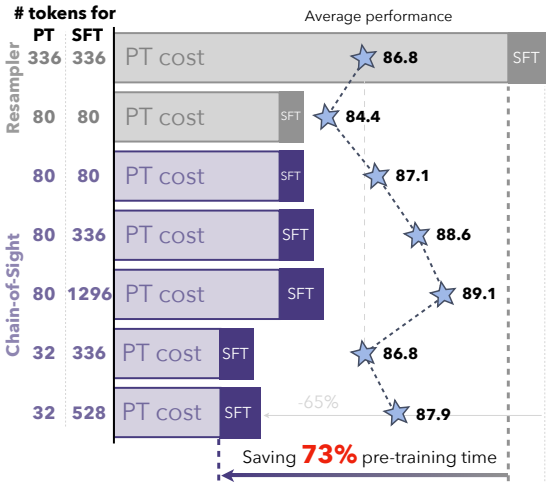

Figure 4: **Pre-train acceleration by Chain-of-Sight**, in comparison with standard resamplers. The average performance is computed over the reported benchmarks in Table 2. Our method achieves a pre-train acceleration of 73% without compromising performance.

Training settings. The training of Chain-of-Sight is divided into two stages. For the first stage, we sample around 65M image-text data involving multiple tasks, as detailed in Table 1. The multi-scale visual resamplers and the LoRA paramters [34] are unlockded for training. For the first 120,000 iterations, we use the input resolution of 224 and unlock the resamplers and the LoRA parameters [34] for training. During the last 30,000 iterations, the input resolution is raised to 448, where the parameters in the visual backbone is further activated and the tokens are scaled up through our compound scaling. The second stage of the Chain-of-Sight model is supervised fine-tuning, where we remove all the captioning datasets except for COCO.

Evaluation benchmarks. The evaluation of our approach involves various tasks including image captioning, visual question answering, text recognition, as well as the tasks defined in popular vision-language benchmarks. Details can be seen in Table A2.

3.2 Ablations

We first ablate the Chain-of-Sight (CoS) design for accelerating the pre-training of MLLMs. For the ablations, we omit the high resolution tuning in the first stage unless otherwise specified.

Pre-train acceleration by Chain-of-Sight. Fig. 4 shows the cost for pre-training and supervised-finetuning with various number of visual tokens, as well as the corresponding average performance over 12 benchmarks. We make several key findings. (a) Though reducing visual tokens for the resampler from 336 to 80 significantly reduces the training time, the average performance drops from 86.8 to 84.4. (b) Using an identical number of tokens, *i.e.,* 80 visual tokens, Chain-of-Sight notably outperforms the standard resamplers, which can be mainly accredited to the multi-scale visual tokens generated by our method. (c) Using the pre-trained model with 80 visual tokens, Chain-of-Sight can be fine-tuned with higher token counts. Using 336 tokens for fine-tuning, our method achieve an average improvement of 1.8pt over the standard resampler with 336 tokens. (d) Notably, Chain-of-Sight with 32 tokens can save up to 73% of the pre-training time, and maintain the same performance as the standard resampler with 336 tokens. (e) Taking fine-tuning into consideration, our method is capable of saving 65% of the total wall-clock time for training a MLLM with improved performance. Note that the percentage is based on a 65M pre-training dataset, and the overall gains in efficiency are expected to grow with the increase of the pre-training dataset scale.

Image captioning, visual question answering, and text recognition. Table 2 compares the performance of Chain-of-Sight with its baselines. Overall, our method delivers competitive performance against pre-training with a full set of visual tokens, while substantially accelerating training speed.

Table 1: **Multitask pretraining data** for pre-training Chain-of-Sight. MeanL., 50%L., and 90%L. indicates the mean length, the 50 percentile, and the 90 percentile length of the input text tokens. We use the tokenizer from Vicuna [20], which is the same tokenizer we use for pre-training.

Task	MeanL.	50%L.	90%L.	Dataset
Caption	23.87	19	43	COYO [7], CC3M&12M [85], COCO [54], VG Cap [39], SBU [75]
General VQA	12.17	11	17	VQAv2 [32], GQA [35]
KB QA	38.67	39	48	OK-VQA [68], AOK-VQA [84], ScienceQA [65]
Text	15.32	13	25	TextVQA [87], OCRVQA [71], TextCaps [86]
REC	12.65	16	19	RefCOCO [38], RefCOCO+ [103], RefCOCOg [67]
Total	23.32	19	42	

Table 2: **Image captioning, visual question answering, text recognition, and vision-language benchmarks**, compared with our baselines. † indicates fine-tuning with 224×224 resolution. ∗ denotes token extended through existing strategies [57, 50, 55]. S-I denotes the image subset of SEEDBench [43]. The best and second-best performances are marked **bold** and underlined.

	pre-train		FT	Captioning			VQA				Text		VLM-Bench		
Bridge	tokens	time	tokens	COCO	Flickr	NoCaps	v2	OK	GQA	SQA	Caps	VQA	MMB	POPE	S-I
224 resolution fine-tuning															
Linear	256	0.82×	256†	140.1	78.6	116.7	79.7	57.3	62.2	90.2	120.0	48.7	67.9	83.2	66.3
CoS	80	0.42×	80†	139.6	78.0	115.8	79.0	58.0	61.6	90.4	120.8	48.3	68.6	84.4	64.6
	80	0.42×	336†	140.8	80.0	117.2	79.8	58.4	62.3	90.0	122.6	50.2	69.2	84.8	65.9
448 resolution fine-tuning															
Resamp.	336	1.00×	336	141.2	81.9	117.2	81.0	58.4	61.9	84.3	135.5	61.3	66.9	85.3	66.2
	80	0.42×	80	138.8	81.4	115.8	80.3	58.0	61.6	84.5	115.8	59.0	68.1	84.5	64.4
	80	0.42×	400*	139.8	84.2	117.3	81.3	58.3	61.9	87.3	135.7	61.9	68.6	86.0	66.3
CoS	336	1.00×	336	141.4	83.4	116.8	80.9	58.4	62.8	88.6	133.9	60.3	68.7	84.2	66.9
	80	0.42×	80	140.7	82.6	118.0	80.7	58.4	61.6	89.6	134.5	60.4	69.0	84.5	65.0
	80	0.42×	336	141.3	**85.8**	119.2	81.7	58.9	62.1	90.5	137.8	63.8	70.2	85.9	66.4
	80	0.42×	1296	**142.8**	84.9	**119.3**	**82.5**	**59.4**	62.5	**91.5**	137.5	**65.0**	**70.3**	**86.4**	67.5
further accelerations															
CoS	48	0.35×	784	142.7	83.4	119.1	82.3	59.1	**62.7**	91.0	**138.4**	64.7	69.7	84.3	67.3
	32	0.27×	336	141.4	83.5	118.1	80.7	57.8	61.1	89.3	133.7	59.5	66.8	84.0	65.1
	32	0.27×	528	141.7	83.1	117.1	81.6	58.3	62.4	91.1	136.4	61.8	69.4	85.3	66.7

Compared to the linear projection, we fine-tune the visual backbone at a resolution of 224x224. Pre-trained with 80 tokens, CoS slightly falls behind when we use 80 tokens for fine-tuning, while achieving stronger performance when we scale up the tokens to 336. In this case, the total time for pre-training can be saved by 50% when compared to using linear projections with 256 visual tokens.

Compared with visual resamplers, when a high number of visual tokens are used during pre-training, Chain-of-Sight performs similarly. However, when pre-trained with fewer visual tokens, Chain-of-Sight have notable advantages on captioning and text recognition for 80 tokens, outperforming resamplers by 1.8 for captioning and 10.1 for text recognition. Further scaling up tokens pushes our model to have a performance stronger than the model pre-trained with 336 tokens. In addition, our compound scaling proves more effective than existing resolution scaling methods [57, 50, 69].

We also try further reducing the number of tokens for pre-training. The performance drop is mainly observed in vision-language benchmarks and question answering capabilities, with minor affect on the captioning task. Nevertheless, we observe that our model pre-trained with 32 tokens achieve a similar or stronger overall performance with resamplers pre-trained with 336 tokens, while taking only 0.27× wall-clock training time (achieving a 3.7× pre-train acceleration).

Vision-language benchmarks. We find the performances on vision-language benchmarks more heaviliy affected by the number of tokens used in the supervised fine-tuning stages than the ones used in the pre-training stage, especially for the question answering task with multiple choices, as in Table 2. This provides a strong support for using a small set of visual tokens in the pre-training stage for acceleration, and a large set of visual tokens for optimal performance.

Table 3: **Ablations on referring expression comprehension** compared with our baselines. † indicates fine-tuning with 224×224 resolution. ∗ denotes token extended through existing strategies [57, 50, 55]. The best and second-best performances are marked **bold** and underlined.

	pre-train		FT	**RefCOCO**			**RefCOCO+**			**RefCOCOg**		
Bridge	tokens	time	tokens	val	test-A	test-B	val	test-A	test-B	val	test	Avg.
224 resolution fine-tuning												
Linear	256	0.82×	256†	88.37	92.38	82.36	82.81	88.82	74.31	83.37	84.82	84.66
CoS	80	0.42×	80†	85.54	90.60	79.31	79.81	87.43	71.83	81.78	82.08	82.30
	80	0.42×	336†	88.43	92.58	83.45	82.79	89.07	75.91	83.66	85.14	85.10
448 resolution fine-tuning												
Resamp.	336	1.00×	336	86.46	90.84	79.82	80.66	87.74	71.63	82.25	82.64	82.76
	80	0.42×	80	83.59	89.27	76.19	77.28	84.77	67.11	78.84	79.57	79.58
	80	0.42×	400*	88.02	92.19	83.08	82.09	89.12	74.82	84.17	84.62	84.76
CoS	336	1.00×	336	89.20	92.96	84.73	83.83	90.57	76.97	85.48	86.26	86.25
	80	0.42×	80	86.32	91.06	81.43	80.47	87.72	74.29	82.95	82.97	83.40
	80	0.42×	336	89.37	93.21	83.96	84.21	90.22	76.58	**86.05**	85.89	86.19
	80	0.42×	1296	89.20	93.02	**85.46**	83.72	90.17	77.40	85.78	86.47	86.40
further accelerations												
CoS	48	0.35×	784	**89.61**	**93.51**	84.93	**84.65**	**90.85**	**77.79**	85.80	**86.87**	**86.75**
	32	0.27×	336	86.97	91.11	81.18	81.30	87.83	73.72	82.58	82.84	83.44
	32	0.27×	528	88.11	92.35	83.51	83.23	89.45	75.99	84.23	84.84	85.21

Table 4: **Ablations on post-pretrain compound scaling of visual tokens.** G./L. denotes scaling strategy over global tokens (window size=16) and local tokens (window size=4). We report the average performance of captioning, VQA, and text recognition. S-I denotes the image subset of SEEDBench [43]. The best and second-best performances are marked **bold** and underlined.

	G.	L.	res.	win. sizes	tokens	∑tokens	Caps	VQA	Text	MMB	POPE	S-I
baseline	-	-	224	[16, 4]	[16, 64]	80	111.1	66.2	84.5	68.6	84.4	64.6
	-	-	448	[32, 8]	[16, 64]	80	113.8	67.0	97.4	69.0	84.5	65.0
Win. Scale	✓	×	224	[16, 8, 4]	[16, 64, 64]	144	111.3	66.0	84.8	67.9	83.6	64.8
	×	✓		[16, 2]	[16, 256]	272	112.4	67.0	86.5	68.0	84.7	66.0
	✓	✓		[16, 8, 2]	[16, 64, 256]	336	112.7	66.9	86.4	69.2	84.8	65.9
Res. Scale	✓	×	448	[32, 16, 8]	[16, 64, 64]	144	114.6	67.0	97.7	68.4	84.8	64.9
	×	✓		[32, 4]	[16, 256]	272	115.0	67.4	100.4	69.9	86.0	66.5
	✓	✓		[32, 16, 4]	[16, 64, 256]	336	115.5	67.8	100.8	70.2	85.9	66.4
Com. Scale	✓	×	448	[32, 8, 4]	[16, 256, 256]	528	115.4	67.8	100.2	69.7	86.2	67.0
	×	✓		[32, 16, 2]	[16, 64, 1024]	1104	114.6	68.0	101.2	**70.8**	86.0	**67.5**
	✓	✓		[32, 8, 2]	[16, 256, 1024]	1296	**115.6**	**68.1**	**101.3**	70.3	**86.4**	**67.5**

Visual grounding. Table 3 shows the comparison between Chain-of-Sight and its baselines on referring expression comprehension (REC). The conclusions are consistent with the above findings. Since Chain-of-Sight incorporates both global and local contexts, using our method notably boost the performance of visual resamplers, achieving an improvement of 3.82 and 3.49 on the average performance when using 80 and 336 visual tokens, respectively. Notably, for the REC task, when compared with Chain-of-Sight model pretrained with 336 tokens, we can achieve at most a 2.85× acceleration (reducing the wall-clock training time to 0.35× using 48 tokens for pre-training) without performance loss. Against the standard visual resampler pre-trained with 336 tokens, our 32-token-variant perform favourably, while reducing the training cost by 73%.

Post-pre-train compound scaling of visual tokens. In Table 4, we ablate our token scaling strategy. The experiment share the same pre-training, where the baselines are models fine-tuned with 80 visual tokens at 224^2 and 448^2 resolutions. For scaling up the number of tokens, we separate the ablations on global and local tokens. Notably, we find scaling up the global tokens alone has negligible effects on the average performances, while scaling up local tokens brings notable improvement on various benchmarks. Combining both further brings slight improvements over the model with up-scaled local tokens on almost all benchmarks. Hence, we use 1296 tokens for fine-tuning our final model.

Table 5: **Comparison with SoTA methods on 10 benchmarks.** Despite that we have only employed LoRA to fine-tune the language model, our model achieves a competitive performance against existing approaches in many benchmarks. *PT tks.* indicates the number of visual tokens used for pre-training and *Parm.* indicates the trainable parameters for the whole model. * indicates at least part of the training set is observed during training. Best performance is marked **bold**. Gray fonts indicate models of larger sizes than ours.

Model	LLM	PT tks.	Parm.	VQA^{v2}	GQA	VizWiz	SQA^{I}	VQA^{T}	POPE	MME	MMB	$SEED^{I}$
InstructBLIP-13B [22]	Vicuna-13B	32	188M	-	49.5	33.4	63.1	50.7	78.9	1212.8	-	-
LLaVA-1.5-13B [56]	Vicuna-13B	576	13B	80.0*	63.3*	53.6	71.6	61.3	85.9	1531.3	67.7	68.1
CogVLM-17B [97]	Vicuna-7B	256	10B	82.3*	-	-	91.2*	70.4	87.9	-	77.6	72.5
VILA-13B [52]	Vicuna-13B	576	13B	80.8*	63.3*	60.6	73.7	66.6	84.2	1570.1	70.3	-
Honeybee-13B [10]	Vicuna-13B	256	13B	-	-	-	-	-	85.5	1629/315	73.2	68.2
Mini-Gemini-13B [48]	Vicuna-13B	576	13B	-	-	-	-	65.9	-	1565/322	68.5	-
InstructBLIP-7B [22]	Vicuna-7B	32	188M	-	49.2	34.5	60.5	50.1	-	-	36.0	58.8
Shikra [12]	Vicuna-7B	-	7B	77.4*	-	-	-	-	-	-	58.8	-
IDEFICS-9B [41]	-	64	9B	50.9	38.4	35.5	-	25.9	-	-	48.2	-
Qwen-VL [4]	Qwen-7B	256	8B	78.8*	59.3*	35.2	67.1	63.8	-	-	38.2	62.3
LLaVA-1.5-7B [56]	Vicuna-7B	576	7B	78.5*	62.0*	50.0	66.8	58.2	85.9	1510.7	64.3	-
mPLUG-Owl2 [99]	LLaMA2-7B	64	7B	79.4*	56.1	54.5	68.7	58.2	85.8	1450.2	64.5	57.8
Honeybee-7B [10]	Vicuna-7B	144	7B	-	-	-	-	-	83.2	1584/307	70.1	64.5
VILA-7B [52]	Vicuna-7B	576	7B	79.9*	62.3*	**57.8**	68.2	64.4	85.5	1533.0	68.9	-
Mini-Gemini-7B [48]	Vicuna-7B	576	7B	-	-	-	-	65.2	-	1523/316	69.3	-
CoS-7B	Vicuna-7B	80	532M	82.9*	64.0*	50.7	93.9*	65.1	85.9	1549/301	72.8	68.9
CoS-8B	LLaMA3-8B	80	540M	**84.3***	**65.3***	-	**95.7***	**67.6**	**86.9**	**1598/308**	**76.6**	**73.1**

In terms of training efficiency in the fine-tuning stage, the fastest model (resolution 224 with 80 tokens) is twice as fast as the medium model (resolution 448 with 336 tokens), and uses around 25% of the time spent on training the model with the 1296 visual tokens. However, since the wall-clock time required for supervised fine-tuning is substantially smaller than the pre-training stage, such an increment on the training time during fine-tuning is acceptable.

3.3 Comparison with existing approaches

Visual question answering and vision-language benchmarks. Table 5 compare the performance of our model with existing approaches. Since the majority of them fine-tunes the whole language model during fine-tuning, the trainable parameters of existing approaches are substantially larger than our approach. Nevertheless, our Chain-of-Sight has achieved competitive performance against existing approaches on many benchmarks, reaching top performance on visual question answering and MMBench among models of the same scale with less than 10% of the trainable parameters. Since the model did not go through an instruction tuning stage, the performance on MME and Vizwiz is not satisfactory. We include the results for more benchmarks in the appendix.

Visual grounding. We compare our model with the existing approaches on visual grounding in Table 6. Despite that the only data source of object localization for training our Chain-of-Sight model is the RefCOCO datasets [38, 67, 103], and that our language model is adapted with LoRA [34], our model achieves a leading performance on these three benchmarks, when compared to existing approaches of a similar scale.

4 Related work

Multi-modal large language models. Since the introduction of the Transformer arhictecture [94] and large-scale pre-training [25, 79], language models have been advancing rapidly [91, 92, 108, 80, 70, 5, 3]. Recently, they are shown to be able to handle various types of data, such as vision [72, 58, 44, 2] and audio [66, 89], leading to a series of multi-modal language models (MLLMs) [4, 12, 99, 109]. The visual capabilities of MLLMs are mainly enabled through transforming visual features into visual tokens, which can be roughly categorized into two types. One uses linear projection to feed image patches into LLMs [58, 11, 16, 93, 97], and the other uses learnable prompts and cross-attentions to aggregate information from the whole feature map [44, 4, 2, 99, 50]. Alternatively, Honeybee [10] proposes a convolutional model for combining the benefit of both. Most existing approaches use an identical number of visual tokens throughout pre-training and fine-tuning. Though some of the recent works have exploited raising the visual tokens during fine-tuning with increased resolution to

Table 6: **Performance comparison on referring expression comprehension** compared with existing approaches. † indicates models fine-tuned with LoRA. Best performance is marked **bold**. Gray fonts indicate models of larger sizes than ours.

		RefCOCO			RefCOCO+			RefCOCOg		
Model	LLM	val	test-A	test-B	val	test-A	test-B	val	test	Avg.
Shikra-13B [12]	Vicuna-13B	87.83	91.11	81.81	82.89	87.79	74.41	82.64	83.16	83.95
Ferret-13B [100]	Vicuna-13B	89.48	92.41	84.36	82.81	88.14	75.17	85.83	86.34	85.57
Griffon v2 [105]	LLaMA2-13B	89.60	91.80	86.50	81.90	85.50	76.20	85.90	86.00	85.42
CogVLM-17B [97]	Vicuna-7B	92.76	94.75	88.99	88.68	92.91	83.39	89.75	90.79	90.25
MAttNet [102]	-	76.40	80.43	69.28	64.93	70.26	56.00	66.67	67.01	68.87
OFA-L [96]	-	79.96	83.67	76.39	68.29	76.00	61.75	67.57	67.58	72.65
UNITER [17]	-	81.41	87.04	74.17	75.90	81.45	66.70	74.02	68.67	76.17
MDETR [37]	-	86.75	89.58	81.41	79.52	84.09	70.62	81.64	80.89	81.81
Shikra-7B [12]	Vicuna-7B	87.01	90.61	80.24	81.60	87.36	72.12	82.27	82.19	82.93
Ferret-7B [100]	Vicuna-7B	87.49	91.35	82.45	80.78	87.38	73.14	83.93	84.76	83.91
MiniGPTv2† [11]	LLaMA2-7B	88.69	91.65	85.33	79.97	85.12	74.45	84.44	84.66	84.29
Qwen-VL-7B [4]	Vicuna-7B	89.36	92.26	85.34	83.12	88.25	77.21	85.58	85.48	85.83
CoS-7B †	Vicuna-7B	90.72	93.83	85.83	86.03	91.02	78.63	87.46	87.94	87.68
CoS-8B †	LLaMA3-8B	**92.67**	**95.14**	**88.89**	**89.12**	**93.63**	**83.25**	**89.56**	**90.42**	**90.33**

enhance downstream performance [50, 55, 57, 69], the large set of visual tokens for each image still presents a major bottleneck for the pre-training stage.

Efficient model pre-training. As the model size consistently expands, the efficiency of training large models has become increasingly important. Beyond efforts in the system optimizations [81, 82, 24, 23], the pre-training of large models can be accelerated by sparse computation, such as masking [46, 77] or mixture of experts [28, 51]. Our approach presents a novel perspective for accelerating pre-training for MLLMs by reducing visual tokens required.

Multi-scale hierarchy in vision. Multi-scale hierarchy is a fundamental property in vision, which has led to the introduction and evolution of convolutional networks [30, 42, 40, 33] as well as its application in various vision problems [61, 83, 53, 13]. Recently, transformers are also shown to benefit from multi-scale hierarchy [98, 60, 27, 47]. This work extends multi-scale hierarchy to language models for stronger visual capabilities and higher training efficiency.

5 Discussions

Limitations. Despite the strong performance and the notable acceleration achieved by Chain-of-Sight, our approach leverages parameter efficient fine-tuning (PEFT) for adapting language models. Hence, the generality of the final model might be limited, compared to approaches that fine-tunes the whole language model during supervised fine-tuning process [57, 10, 52] or even the pre-training stage [63, 4, 26]. This is mainly due to the limited training resources and is exactly what motivates us to explore efficient pre-training methods. We believe the pre-train acceleration achieved by the presented approach has stronger potentials beyond our results.

Conclusions. In this work, we set out to accelerate the pre-training phase of MLLMs. Motivated by the unbalance between the number of visual and text tokens during pre-training, we present Chain-of-Sight to reduce the number of token required for pre-training. Chain-of-Sight produces visual tokens of multiple visual scales, providing various level of granularity for the MLLMs to have better perception capabilities. The proposed compound token scaling strategy in the fine-tuning stage can substantially increase the number of tokens post pre-train, such that the model can achieve competitive performance despite the low token count during pre-training. Empirical results have shown that our Chain-of-Sight is capable of achieving a 3.7× speed up in the pre-training process with on-par or better downstream performances. We hope our research can facilitate further investigations in efficient pre-training of MLLMs.

References

[1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957, 2019.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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

[5] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] 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.

[8] Yun-Hao Cao, Kaixiang Ji, Ziyuan Huang, Chuanyang Zheng, Jiajia Liu, Jian Wang, Jingdong Chen, and Ming Yang. Towards better vision-inspired vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13537–13547, 2024.

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[10] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. *arXiv preprint arXiv:2312.06742*, 2023.

[11] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun 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.

[12] 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.

[13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[14] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023.

[15] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023.

[16] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

[17] 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 *European conference on computer vision*, pages 104–120. Springer, 2020.

[18] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024.

[19] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.

[20] 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. https://lmsys.org/blog/2023-03-30-vicuna/, 2023.

[21] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[22] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[24] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[26] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, et al. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *arXiv preprint arXiv:2404.06512*, 2024.

[27] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021.

[28] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[29] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[30] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[31] Peng Gao, Renrui Zhang, Chris Liu, Longtian Qiu, Siyuan Huang, Weifeng Lin, Shitian Zhao, Shijie Geng, Ziyi Lin, Peng Jin, et al. Sphinx-x: Scaling data and parameters for a family of multi-modal large language models. *arXiv preprint arXiv:2402.05935*, 2024.

[32] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[34] 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. *arXiv preprint arXiv:2106.09685*, 2021.

[35] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[36] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[37] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021.

[38] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[39] 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. *International journal of computer vision*, 123:32–73, 2017.

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[41] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36, 2024.

[42] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[43] 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.

[44] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[45] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6054–6063, 2019.

[46] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400, 2023.

[47] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022.

[48] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024.

[49] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

[50] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607*, 2023.

[51] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024.

[52] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. *arXiv preprint arXiv:2312.07533*, 2023.

[53] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[54] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[55] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023.

[56] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[57] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[58] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[59] 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.

[60] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[61] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[62] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[63] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

[64] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. *arXiv preprint arXiv:2312.17172*, 2023.

[65] Pan Lu, Swaroop Mishra, Tanglin 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. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

[66] Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *arXiv preprint arXiv:2306.09093*, 2023.

[67] 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*, pages 11–20, 2016.

[68] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[69] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[70] Meta. Introducing meta llama 3: The most capable openly available llm to date. 2024.

[71] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE, 2019.

[72] Yao Mu, Junting Chen, Qinglong Zhang, Shoufa Chen, Qiaojun Yu, Chongjian Ge, Runjian Chen, Zhixuan Liang, Mengkang Hu, Chaofan Tao, et al. Robocodex: Multimodal code generation for robotic behavior synthesis. *arXiv preprint arXiv:2402.16117*, 2024.

[73] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 36, 2024.

[74] OpenAI. Gpt-4v(ision) system card. 2023.

[75] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011.

[76] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[77] Zhiwu Qing, Shiwei Zhang, Ziyuan Huang, Xiang Wang, Yuehuan Wang, Yiliang Lv, Changxin Gao, and Nong Sang. Mar: Masked autoencoders for efficient action recognition. *IEEE Transactions on Multimedia*, 2023.

[78] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[79] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *Technical Report*, 2018.

[80] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[81] 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.

[82] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020.

[83] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[84] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer, 2022.

[85] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[86] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 742–758. Springer, 2020.

[87] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

[88] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, et al. Generative multimodal models are in-context learners. *arXiv preprint arXiv:2312.13286*, 2023.

[89] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*, 2023.

[90] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[91] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[92] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[93] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

[94] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[95] Guangzhi Wang, Yixiao Ge, Xiaohan Ding, Mohan Kankanhalli, and Ying Shan. What makes for good visual tokenizers for large language models? *arXiv preprint arXiv:2305.12223*, 2023.

[96] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022.

[97] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

[98] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[99] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv preprint arXiv:2311.04257*, 2023.

[100] 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.

[101] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[102] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1307–1315, 2018.

[103] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016.

[104] 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.

[105] Yufei Zhan, Yousong Zhu, Hongyin Zhao, Fan Yang, Ming Tang, and Jinqiao Wang. Griffon v2: Advancing multimodal perception with high-resolution scaling and visual-language co-referring. *arXiv preprint arXiv:2403.09333*, 2024.

[106] Qinglong Zhang and Yu-Bin Yang. Rest: An efficient transformer for visual recognition. *Advances in Neural Information Processing Systems*, 34:15475–15485, 2021.

[107] Qinglong Zhang and Yu-Bin Yang. Rest v2: simpler, faster and stronger. *Advances in Neural Information Processing Systems*, 35:36440–36452, 2022.

[108] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[109] 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.

A Broader impact

The Chain-of-Sight model employs off-the-shelf pre-trained language models. As such, all CoS models inherits the shortcomings from the language model, including certain model biases or hallucinations. While to some extent CoS enhances the visual capabilities of language models, proper assessment and safety precautions are still required before deploying our model.

B Details on multi-level feature aggregation

Our multi-scale visual resamplers consider visual hierarchy in two aspects. In addition to the multiple spatial scales detailed in the manuscript, we also enable the visual resamplers to aggregate from multiple feature levels in the visual backbone, which is useful in a wide range of visual tasks [47, 53] but often neglected in current MLLMs.

In Fig. A1, we demonstrate the structure for the multi-level feature aggregation. Essentially, we exploit features from multiple layers in the visual backbone, and the learnable queries aggregate sequentially from lower level to higher level features.

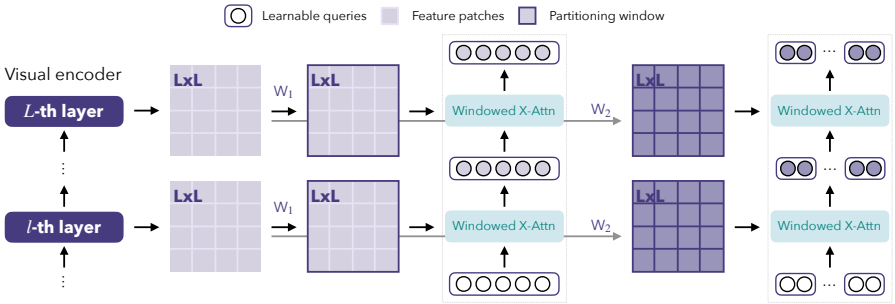

Figure A1: **Multi-level feature aggregation** in the multi-scale visual resamplers of Chain-of-Sight.

C Details on training data and evaluation benchmarks

Here, we provide details on the training data and the evaluation benchmarks.

Table A1: **Dataset statistics used for Chain-of-Sight pre-training**. MeanL., 50%L., and 90%L. indicates the mean length, the 50 percentile, and the 90 percentile length of the input text tokens.

Task	Dataset	MeanL.	50%L.	90%L.	Statistical count
Captioning	COYO [7]	24.8	20	44	46M
	CC3M&CC12M [85]	22.5	17	43	10M
	COCO [54]	13.4	14	19	0.6M
	VG Caption [39]	7.9	7	11	1M
	SBU [75]	18.2	19	43	0.8M
General VQA	VQAv2 [32]	10.6	10	14	0.6M
	GQA [35]	13.3	13	19	0.9M
Knowledge-based VQA	OK-VQA [68]	13.4	13	18	9k
	AOK-VQA [84]	40.1	40	46	68k
	ScienceQA [65]	45.5	38	77	19k
Text	TextVQA [87]	13.1	13	17	35k
	OCRVQA [71]	15.1	13	25	1M
	TextCaps [86]	18.2	17	25	0.1M
REC/REG	RefCOCO [38] RefCOCO+ [103] RefCOCOg [67]	18.0/7.3	18/6	19/13	321k

Table A1 shows per-dataset length distributions of our training data. The majority of the data used in the pre-training stage have an average length lower than 25, which provides strong motivation for our approach that reduces the number of visual tokens for pre-training acceleration.

Table A2 shows the detailed description on the benchmarks we use to evaluate our model.

Table A2: Summary of the evaluation benchmarks.

Task	Dataset	Description	Split	Metrics
Captioning	NoCaps [1]	Captioning of natural images.	val	CIDEr (↑)
	Flickr [76]	Captioning of natural images.	karpathy-test	CIDEr (↑)
	COCO [54]	Captioning of natural images.	karpathy-test	CIDEr (↑)
General VQA	VQAv2 [32]	VQA on natural images.	test-dev	VQA Score(↑)
	GQA [35]	VQA on scene understanding and reasoning	test-balanced	VQA Score (↑)
	OK-VQA [68]	VQA on natural images requiring outside knowledge.	val	VQA Score (↑)
	ScienceQA-Img [65]	Multi-choice VQA on a diverse set of science topics	test	Accuracy (↑)
Text-rich benchmarks	TextVQA [87]	VQA on natural images containing text.	val	VQA Score (↑)
	TextCaps [86]	Captioning of natural images containing text.	test	CIDEr (↑)
LVLM Benchmarks	SEED-Bench [43]	Multi-choice VQA on a diverse set of topics	IMG	Accuracy (↑)
	MMBench [59]	Multi-choice VQA on a diverse set of topics	test	Accuracy (↑)
	MME [29]	Open-ended VL Benchmark by yes/no questions	Perception & Cognition	Accuracy (↑)
	POPE [49]	Multi-choice VQA for testing hallucinations	overall	F1-Score (↑)
	MMMU [104]	VQA on a diverse set of topics	val	Accuracy (↑)
Grounding	RefCOCO [38]	Refer grounding on natural images.	overall	Accuracy (↑)
	RefCOCO+ [103]	Refer grounding on natural images.	overall	Accuracy (↑)
	RefCOCOg [67]	Refer grounding on natural images.	overall	Accuracy (↑)

D Detailed training settings

We include the detailed parameters for training Chain-of-Sight in Table A3.

Table A3: Training hyperparameters of the chain-of-sight models.

Configuration	Multi-task pre-training	Supervised fine-tuning
Image resolution	224^2 | 448^2	448^2
ViT initialization	CLIP ViT-L/14	CLIP ViT-L/14
ViT freeze	yes | no	no
LLM adaptation	LoRA (r=64)	LoRA (r=64)
Optimizer	AdamW [62]	
Optimizer hyperparameter	$\beta_1 = 0.9, \beta_2 = 0.98$	
Peak learning rate	$2e^{-4}$ | $3e^{-5}$	$3e^{-5}$
Minimum learning rate	$1e^{-6}$	$1e^{-6}$
ViT learning rate decay	- | 0.9	0.9
ViT Drop path rate	0	
Learning rate schedule	cosine decay	
Weight decay	0.1	
Training steps	120000 | 30000	20000
Warm-up steps	2000	2000
Global batch size	512	256
Numerical precision	`bfloat16`	

E Further results

We provide further empirical results for CoS-7B and CoS-8B in Table A4.

Table A4: Further empirical results. 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.

	Regular				MMBench							Other
Model	OK	COCO	NoCaps	Flickr	LR	AR	RR	FP-S	FP-C	CP	Total	$MMMU_v$
CoS-7B	60.3	143.0	119.7	86.0	46.6	80.9	69.5	74.1	62.2	82.7	72.8	35.4
CoS-8B	62.7	142.5	119.7	85.0	46.6	82.9	80.0	77.8	71.3	84.1	76.6	39.7

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract have indicated that the aim of the paper is to accelerate the pre-training of MLLMs, and the empirical results achieved by CoS (reducing wall-clock training time by 73% without performance loss) is validated in the experiments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are provided at the end of the manuscript.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We do not have theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided details for training our model in description and using tables.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the data are publicly accessable, and the details of the structure is clear enough to reproduce the result.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, please refer to the appendix for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: Even though our approach uses LoRA for fine-tuning, the whole training process is still computational intensive. Hence, it is difficult to provide error bars for the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided that in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Definitely.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We have discussed broader impacts in our appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: We do not plan to release any data. For the model, we currently do not have safeguards for releasing it. We will make sure that the guidelines and instructions are in place when we release the model.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, we have.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets are provided.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.