ConvLLaVA: Hierarchical Backbones as Visual Encoder for Large Multimodal Models

Chunjiang Ge¹, Sijie Cheng³, Ziming Wang², Jiale Yuan², Yuan Gao² Jun Song², Shiji Song¹, Gao Huang^{1,⊠}, Bo Zheng² ¹Department of Automation, Tsinghua University ²Alibaba Group ³Department of Computer Science and Technology, Tsinghua University gecj20@mails.tsinghua.edu.cn

Abstract

High-resolution Large Multimodal Models (LMMs) encounter the challenges of excessive visual tokens and quadratic visual complexity. Current high-resolution LMMs address the quadratic complexity while still generating excessive visual tokens. However, the redundancy in visual tokens is the key problem as it leads to more substantial compute. To mitigate this issue, we propose ConvLLaVA, which employs ConvNeXt, a hierarchical backbone, as the visual encoder of LMM to replace Vision Transformer (ViT). ConvLLaVA compresses high-resolution images into information-rich visual features, effectively preventing the generation of excessive visual tokens. To enhance the capabilities of ConvLLaVA, we propose two critical optimizations. Since the low-resolution pretrained ConvNeXt underperforms when directly applied on high resolution, we update it to bridge the gap. Moreover, since ConvNeXt's original compression ratio is inadequate for much higher resolution inputs, we train a successive stage to further compress the visual tokens, thereby reducing redundancy. These optimizations enable ConvLLaVA to support inputs of 1536×1536 resolution generating only 576 visual tokens, capable of handling images of arbitrary aspect ratios. Experimental results demonstrate that our method achieves competitive performance with state-of-the-art models on mainstream benchmarks. The ConvLLaVA model series are publicly available at https://github.com/alibaba/conv-llava.

1 Introduction

Large Multimodal Models (LMMs; [40, 45, 1]) have achieved notable advancements in recent years, demonstrating superior performance in diverse domains, including image and video understanding [54, 11], digital agent development [53], and robotics [24]. The imperative to comprehend a wide range of tasks and intricate scenes underscores the critical role of the visual encoder, which is mostly a Vision Transformer (ViT; [12]). However, ViT's quadratic spatial complexity and output of excessive visual tokens limit its application in diverse and high-resolution tasks [54, 21, 11, 8]. The excessive visual tokens lead to a significant computational burden in the Large Language Model (LLM; [46, 47]), far exceeding the computational cost imposed by the quadratic spatial complexity in the visual encoder. Such redundancy in the visual tokens not only sacrifices efficiency but also impedes the effective extraction of visual information [31, 11]. While a range of methods (Tab. 1; [31, 27, 49]) have been proposed to remedy the quadratic spatial complexity of ViT, they fail to mitigate the key problem, the redundancy in the visual tokens [5, 28].

[™] Corresponding author.

Table 1: Comparison with previous methods. Res., VE, #V Tokens denote resolution, visual encoder, and the number of visual Tokens. Enumerate aspect ratio (Enum) indicates that the model supports a set of predefined aspect ratios. Fix aspect ratio means the model supports fixed resolution input. Any aspect ratio means the model supports arbitrary aspect ratio input. *: OtterHD does not actually have a visual encoder. The spatial complexity for its visual tokens is quadratic.

Method	Res.	#V Tokens	Complex Cropping	x Design Extra VE	Vi Model	isual Encoder Complexity	Aspect Ratio
LLaVA-1.5 OtterHD	336 1024	576 1225			ViT None	Quadratic Quadratic*	Fix Any
LLaVA-NExT	672	2880	1		ViT	Linear	Enum
MiniGemini-HD	1536	2880	1	1	ViT,ConvNeXt	Linear	Enum
ConvLLaVA ConvLLaVA	1024 1536	256 576			ConvNeXt ConvNeXt	Linear Linear	Any Any

Hierarchical visual backbones [15, 16, 10], which can be considered as counterparts to ViT, can well address the problem of excessive visual tokens due to their inherent *Information Compression* process. Specifically, features are sequentially compressed across stages in hierarchical backbones. They compress visual features by $32 \times [15, 34]$ compared to ViT with only $14 \times [12]$. Therefore, at the same resolution they generate fewer than 1/4 visual tokens compared to ViT, significantly alleviating computational burdens on the LLM. Moreover, hierarchical visual encoders, typically designed with linear spatial complexity [34, 10, 15], effectively tackle both the issue of excessive visual tokens and the quadratic visual complexity.

We choose to employ ConvNeXt among the hierarchical visual encoders due to its excellent performance [48, 56] and the availability of off-the-shelf contrastive language-image pretrained weights (CLIP; [41]), which mainstream visual encoders of LMMs adopt [23, 32, 2, 39]. However, directly replacing ViT with ConvNeXt leads to inferior performance on general capabilities benchmarks (Section 3.2). This can be attributed to the fact that ConvNeXt is pretrained on low resolution, whereas we directly apply it to high-resolution [17, 43]. Moreover, the pretraining data for ConvNeXt is considered to be of low quality [51, 17, 43] compared to ViT's pretraining data [41]. To address these issues, we propose to update the visual encoder rather than freezing it. Surprisingly, updating the visual encoder enables ConvNeXt to perform comparably to ViT on general benchmarks. On fine-grained benchmarks, we observe that ConvNeXt outperforms ViT. These findings indicate that even when compressing visual tokens to an equal quantity, the higher resolution model's features still contain more fine-grained information. This observation inspires us to further scale up the resolution. However, further scaling the resolution beyond 1024 leads to the generation of excessive visual tokens. To mitigate this issue, we further compress the visual information with an additional ConvNeXt stage to enhance the inherent *information compression* of hierarchical backbones. The visual inputs would be compressed by $64 \times$ rather than $32 \times$ to further reduce the redundancy. Hence, ConvLLaVA generates only 576 visual tokens when processing 1536 resolution inputs, which is equivalent to the number of visual tokens generated by ViT when processing 336 resolution inputs (Section 3.3).

In summary, we introduce ConvLLaVA whose visual encoder is a five-stage ConvNeXt. ConvLLaVA compresses high-resolution images into information-rich visual features, effectively avoiding the generation of excessive visual tokens (in Tab. 1; [31, 27, 25, 36]). Furthermore, thanks to the translation equivalence of convolution, ConvLLaVA can be trained on low-resolution and evaluated on higher resolutions, and it can also handle images of arbitrary aspect ratio. Extensive experiments have demonstrated the effectiveness of our method. ConvLLaVA 7B outperforms LLaVA-1.5-13B across various benchmarks, including MME [13], MMBench [33], SEEDBench [22], RealWorldQA [50], TextVQA [44], DocVQA [38], POPE [26], and MMVet [57].

2 Related Work

Large Multimodal Models. To harness the potential of Large Language Models and incorporate visual information, BLIP series models [23, 9] propose the Q-former, which generates visual tokens for LLMs to interpret visual data. Meanwhile, LLaVA [32] employs a single linear layer to map



Figure 1: We show the structure for LLaVA and ConvLLaVA in (a) and (b). ConvNeXt has a hierarchical structure which compresses visual tokens between stages. The training procedure is composed of three training stages and the trainable parameters for each stage are shown in (c).

visual features to the word embedding space, allowing LLMs to perceive vision features. These approaches utilize the ViT as the visual encoder [41, 12, 3, 28, 60], primarily tailored for low-resolution visual data (e.g., 224 or 336 resolution). Moreover, Qwen-VL [2] and mPLUG-owl2 [55] scale the resolution of ViT to 448 by updating the weights of ViT. However, these methods fail to further scale up resolution due to the quadratic spatial complexity of ViT, while ConvNeXt can scale up the resolution with the linear cost increase. Qwen-VL [2] and mPLUG-owl2 [55] also explore to reduce the visual tokens via resampler. However, recent studies [3, 11] show that convolution or simply concatenation performs better than resampler.

High-resolution LMMs with Cropping. The representative cropping method for high-resolution LMMs is introduced in LLaVA-NExT [31], which partitions an image into four patches, each encoded separately by ViT and subsequently concatenated for LLM processing. A collection of methods have adopted cropping to scale up resolution [54, 29, 27, 11]. While effective in reducing ViT complexity, cropping compromises the structural integrity of the image, thus potentially impacting overall performance. Moreover, the proliferation of visual tokens introduced by cropping poses significant complexity on LLMs and challenges the retrieval capabilities of LLMs [11].

High-resolution LMMs with Extra Visual Encoders. Incorporating an auxiliary visual encoder for high-resolution image understanding would not significantly increase the number of visual tokens. Vary [49] and Deepseek-VL [35] utilize SAM [20] as a high-resolution visual encoder to augment the feature of ViT. MiniGemini-HD [25] and LLaVA-HR [36] employ ConvNeXt [17] to process high-resolution images and use cross-attention or adapters to extract features from the high-resolution input. However, these methods introduce additional complexity through supplementary visual encoders and associated hyperparameters. Furthermore, extracting features from low-quality representations (e.g., LAION-CLIP-ConvNeXt) may potentially compromise LMMs' performance [14, 51].

3 ConvLLaVA

We present ConvLLaVA, as illustrated in Fig. 1 (b), whose visual encoder is a five-stage ConvNeXt. We first introduce the overall architecture and the advantages of our ConvLLaVA in Section 3.1. The two major optimizations: updating the visual encoder and training an additional stage are introduced in Section 3.2 and Section 3.3.

3.1 ConvNeXt as Standalone Visual Encoder

The architecture of ConvLLaVA is identical to most popular general LMMs, *e.g.*, LLaVA [32, 30], Qwen-VL [2], and VILA [28]. These models comprise three components as shown in Fig. 1 (a): a vision encoder g(), a large language model f(), and a vision-language projector h(). Specifically, the vision model encodes the visual inputs x into latent visual embeddings g(x). The vision-language projector then maps the latent visual embeddings into the embedding space of the language model z = h(g(x)). Given the visual embeddings z and text embeddings t encoded by the language tokenizer, these embeddings are concatenated along the sequence dimension and then passed to the

language model. Finally, the vision language model is trained with language modeling loss [42]. Considering that our study mainly focuses on the visual encoder, we employ a two-layer MLP and Vicuna-7B [59] as the projector and language model following LLaVA-1.5 [30]. Rather than using CLIP-VIT [41], we introduce CLIP-ConvNeXt [34, 17] as the standalone visual encoder.

ConvNeXt. The basic block of ConvNeXt comprises a depth-wise convolution and a feed-forward network [34]. The depth-wise convolution has a 7×7 kernel size, and the computation complexity is $\mathcal{O}(k^2 CN)$, where k, C, and N are the kernel size, number of channels, and number of visual tokens, respectively. In contrast, the complexity of self-attention in ViT is $\mathcal{O}(4C^2N + 2CN^2)$. Consequently, the spatial complexity of ConvNeXt is significantly lower than ViT. The input is initially processed by a 4×4 non-overlapping convolution downsampling layer. Subsequently, the features are successively fed into the four stages of ConvNeXt, while each stage comprises several ConvNeXt blocks. Feature maps are downsampled by $2\times$, and dimensions are expanded by $2\times$ between stages. The output of the ConvNeXt is downsampled by $32\times$, rather than $14 \times$ of ViT-L. Hence, ConvNeXt produces less than 1/4 visual tokens compared to ViT, which alleviates the computation load of the language model. Benefiting from the linear spatial complexity and fewer visual tokens,





Figure 2: Estimated FLOPs v.s. image resolution for different backbones. We estimate total FLOPs with the number of visual tokens.

the computation reduction of LMMs from ViT-L (red line) to ConvNeXt (blue line) is almost $8 \times$ as illustrated in Fig. 2.

Five-stage ConvNeXt[†]. Leveraging ConvNeXt as the visual encoder is efficient for encoding 768 resolution images, while scaling resolutions to higher than 768 produces excessive visual tokens. Previous studies [31, 25] neglect to explore compressing visual tokens, while compressing visual tokens has been proven to be reasonable since there is redundancy in the visual representation [28, 5]. These studies suggest that we can further downsample visual features using ConvNeXt. We propose to compress visual features by incorporating ConvNeXt blocks for stage 5 into the original four-stage model. We prefer using ConvNeXt blocks over other structures due to the following three reasons (1) The five-stage ConvNeXt, as a whole, could be transferred as a visual encoder for other LMMs, whereas downsampling in the projector does not offer such flexibility (2) ConvNeXt blocks maintain translation equivariance, allowing them to effectively process images of any aspect ratio, unlike attention blocks. (3) The impact on performance from the downsampling stage is minimal, except that the resampler consistently underperforms compared to other methods, as evidenced by [3, 11, 39]. Finally, we denote the overall five-stage ConvNeXt as ConvNeXt[†]. At 1536 resolution, ConvNeXt[†] reduces the number of visual tokens to 576, equivalent to that of ViT at 336 resolution. This would reduce the total computation by $6 \times w.r.t$. the original ConvNeXt (blue line) to ConvNeXt[†] (green line) as shown in Fig. 2. Our approach is more computationally efficient than cropping methods, which often produce an excessive number of visual tokens [39, 31, 27]. Furthermore, by eliminating the need for cropping and merging, ConvLLaVA avoids the global view, thereby further reducing the number of visual tokens.

3.2 Updating ConvNeXt is Essential

The mainstream optimization approach [32, 28] freezes the vision encoder during training, as it has better performance and is more efficient than updating the visual encoder [18]. However, freezing ConvNeXt during training is sub-optimal. Hence, we conduct depth analysis to prove that freezing the visual encoder (i.e., ConvNeXt) would inherit the defects from pretraining, and updating ConvNeXt may both improve the quality of representations and adapt them to high-resolution inputs.

Setups of Freezing ConvNeXt. The optimization procedure is the same as LLaVA-1.5 [30]. For training the projector and instruction tuning, we use the same 558k caption dataset and 665k instruction data, respectively. Our visual encoder CLIP-ConvNeXt-L is pretrained on 256 resolution and fine-tuned with 320 resolution based on LAION-2B [34, 17]. We directly increase the resolution to 512 and 768 when applying ConvNeXt as the vision encoder. As for the baseline, we use ViT

Visual Encoder	#Params	Res.	PT Data	MMBench	SEEDBench	TextVQA	DocVQA
ViT-L	304M	336	WIT	66.5	65.3	45.5	21.2
ConvNeXt-L ConvNeXt-L	200M 200M	512 768	LAION-2B LAION-2B	64.5 63.2	63.8 65.7	49.3 53.7	23.0 29.8

Table 2: Comparison on different visual encoders. Visual encoders are frozen during training. #Params, Res., PT Data is short for Number of Parameters, Resolution, and Pretraining Data.

Table 3: Comparison on different visual encoders. The visual encoders are updated during training. The number in the bracket is the improvement compared with freezing the visual encoder. The absolute improvement shown in green is according to the results in Tab. 2. #V Tokens stands for the number of visual tokens.

Visual Encoder	Res.	#V Tokens	MMBench	SEEDBench	TextVQA	DocVQA	Δ
ViT-L	336	576	66.8 (+0.3)	68.1(+2.8)	50.1(+4.6)	26.4(+5.2)	+3.2
ConvNeXt-L ConvNeXt-L	512 768	256 576	65.5(+1.0) 66.5(+3.3)	67.9 (+4.1) 68.6 (+2.9)	55. 1(+5.8) 60.0 (+6.3)	31.4 (+8.4) 40.2 (+10.4)	+4.8 + 5.7

which is pretrained on 336 resolution with OpenAI WIT dataset [41]. The training and inference speed for ConvNeXt on 768 resolution is on par with ViT on 336 resolution. Hence, we consider the comparison between 768-resolution ConvNeXt and 336-resolution ViT to be fair. Detailed training procedure is shown in Tab. 12.

Benchmarks. We use four standard benchmarks to evaluate the results: two general capability benchmarks, MMbench [33], SEEDBench [22], and two fine-grained OCR benchmarks, TextVQA [44] and DocVQA [38]. It is worth noting that our evaluation procedure for TextVQA differs slightly from LLaVA-1.5 [30], as we use VLMEVALKIT which does not include OCR tokens in the question.

Results for Freezing the Visual Encoder. As shown in Tab. 2, we observe the following results:

(1) ConvNeXt has significant advantages over ViT on OCR benchmarks. On TextVQA and DocVQA, both 512 and 768 resolution ConvNeXt outperforms ViT due to their higher resolution [18, 55]. Even with fewer visual tokens, the 512-resolution ConvNeXt still outperforms the 336-resolution ViT.

(2) The overall general capability of ConvNeXt is inferior to ViT. For general benchmarks, on SEEDBench, 768-resolution ConvNeXt performs comparably with ViT. While on MMBench, ConvNeXt underperforms ViT. We hypothesize that there are two reasons for the performance gap on MMbench: First, ConvNeXt is pretrained on low resolution but directly applied on high resolution. Such employment affects the quality of visual features. Second, the pretrained representation for ConvNeXt may be inferior to OpenAI's ViT [41].

The results imply that increasing resolution without training could affect the quality of representation and hamper the performance of LMMs. However, studies have shown that simply updating the visual encoder during instruction tuning can hinder performance [18]. To mitigate this issue, ShareGPT4V [6] provides an effective training protocol and a high-quality dataset for updating the visual encoder. Therefore, we adopt this effective method to update the visual encoder.

Setups of Updating ConvNeXt. To update the visual encoder, we first leverage the 558k caption dataset for projector initialization [30]. Then, we apply a high-quality caption dataset, ShareGPT4V-PT [6], to train the entire vision-language model including the visual encoder. Finally, the LLaVA 665k instruction tuning dataset is used for visual instruction tuning. The detailed training procedure is shown in Tab. 13. The last 12 layers of ViT-L are trainable (according to ShareGPT4V [6]). For ConvNeXt, we update the last 18 blocks (ConvNeXt-L has a total of 36 blocks).

Results for Updating the Visual Encoder. As shown in Tab. 3, we observe the following results:

(1) ConvNeXt has significant advantages over ViT on the OCR benchmark. The improvement for 768 resolution ConvNeXt is larger than 336 resolution ViT (6.3/10.4 v.s. 4.6/5.2). These results demonstrate the idea of compressing high-resolution visual inputs to a small number (*e.g.*, 576) of

Visual Encoder	Resolution	#Visual Tokens	MMBench	SEEDBench	TextVQA	DocVQA
ConvNeXt-L†	768	144	65.3	67.7	54.7	31.1
ConvNeXt-L†	1024	256	65.1	68.3	59.4	35.8
ConvNeXt-L†	1536	576	64.3	69.1	60.7	42.5

Table 4: Results of training stage 5 ConvNeXt model. The number of visual tokens does not greatly increase when scaling up resolution.

information-rich visual tokens is feasible. Compressing does not lead to great information loss. Even with the same number of tokens, ConvNeXt preserves more fine-grained visual information and significantly outperforms ViT.

(2) For general benchmarks, ConvNeXt performs on par with ViT. Specifically, ConvNeXt outperforms ViT on SEEDBench and performs on par with ViT on MMBench. Notably, the performance gap between the 768 resolution ConvNeXt and the 336 resolution ViT on MMBench is narrowed from 3.3 to 0.3 compared with freezing the visual encoder. This implies that updating the visual encoder is essential. To further support this, we show the results of updating the visual encoder with more data in Appendix A.

Generally, the updated ConvNeXt performs better than ViT on these 4 benchmarks. This evidences that updating the ConvNeXt significantly enhances the performances, underscoring its critical importance. Previous methods employ ConvNeXt as an auxiliary visual encoder and directly increase the resolution to 1024 [36] or 1536 [25]. They fail to identify the problem that scaling up the resolution without updating ConvNeXt would compromise the performance. Our method, delving deeper into the root of the issue, provides a simple yet effective solution to scaling up the resolution.

3.3 Training with Stage 5 Scales up Resolution to 1536

As we mentioned in Section 3.1, scaling resolution to higher than 768 would generate excessive visual tokens. To reduce the redundancy and mitigate the excessive computational demands on the large language model (LLM), we propose training stage 5 for the ConvNeXt model to compress the visual information (training protocol shown in Fig. 1 (c)).

Implementation Details. We employ a three-stage training protocol. In the projector initialization stage, we train the fifth stage layers and the projector with the ShareGPT4V-PT data [6]. In the second stage, we train the entire model with the ShareGPT4V-PT data. For instruction tuning, we utilize the 665k LLaVA instruction data to train the LLM and the projector. The training protocol is similar to the protocol for updating the visual encoder. The only difference is that we train the fifth stage and projector with ShareGPT4V-PT data, while experiments in Section 3.2 train the projector with the 558k caption data in the first training stage. We add 6 layers in stage 5 and tune the last three stages in the second training phase. Ablation studies on these hyper-parameters are included in Appendix B.

Results for ConvNeXt[†]. We present the results of adding stage 5 to ConvNeXt in Tab. 4. Scaling up the resolution consistently improves performance on SEEDBench, TextVQA, and DocVQA, which require fine-grained understanding and benefit from the higher resolution. These results highlight the effectiveness of our method of training stage 5. However, on MMBench, the performance of ConvNeXt[†] exhibits a slight drop when scaling the resolution from 1024 to 1536. The resolution of 1536 is approximately six times higher than the pretraining resolution (256). Adapting the pretrained visual encoder to effectively extract global information from such a significant increase in resolution requires a substantial amount of training data. In Section 4, we verify this hypothesis by providing sufficient data to the visual encoder in the second training stage.

On Scaling Resolution. When we increase the resolution, the number of visual tokens also increases. These two factors are entangled, and there has been a lack of in-depth investigation into the relationship between them. Previous work claims that raw resolution matters more than the number of visual tokens [28]. We experiment on the general benchmark SEEDBench and OCR benchmark DocVQA to investigate these assumptions. Our method provides control experiments to reveal the relationship between resolution and the number of visual tokens. We compare the results of ConvNeXt (trained in Section 3.2) and ConvNeXt† (trained in Section 3.3) as the visual encoder for LMMs under the same number of visual tokens. The two series of models are pretrained with ShareGPT4V-PT and



Figure 3: Comparisons of ConvNeXt and ConvNeXt[†] on SEEDBench and DocVQA. The marked number above the line shows the resolution of the model.

instruction-tuned with 665k LLaVA instruction data. ConvNeXt[†] has an additional stage to compress the number of visual tokens to 1/4. Hence, the differences between these two series models have been largely reduced. Our control experiments reveal novel findings:

(1) When the number of visual tokens is the same, the higher resolution model exhibits better performance on SEEDBench and DocVQA. In the Fig.3, the green line consistently outperforms the blue line. This is because that high-resolution model provides finer-grained and higher-quality visual features even if the output number of visual tokens is the same. Previous work [31, 27, 11] which scales up the resolution by splitting the image into patches would generate excessive visual tokens. Such cropping methods significantly sacrifice efficiency and challenge the retrieval capability of LLM. Our core discovery presents a promising approach to enrich the information contained in visual features without compromising efficiency. Compressing high-resolution images into information-rich visual tokens is more efficient than the cropping method. Training a stage to further compress visual features provides a manner to increase resolution and maintain a moderate computational cost.

(2) The importance of the number of visual tokens varies across different benchmarks at equivalent resolution. For general benchmarks like SEEDBench, the performance drop brought by compressing visual tokens for the 768-resolution models is marginal (0.9 on SEEDBench). However, for OCR benchmarks like DocVQA, the performance drop for the model with fewer visual tokens is substantial (9.1 on DocVQA). Overall, these results demonstrate that while compressing visual tokens causes only slight information loss on general benchmarks, but leads to significant information loss on fine-grained OCR benchmarks.

4 Experiments

Our results demonstrate that scaling up the resolution of ConvNeXt and updating the visual encoder are two effective approaches to training an advanced, high-resolution Language-Multimodal Model. However, we found that the available training data was insufficient to fully unleash the potential of these approaches. Consequently, we scaled up the high-quality training data to address this limitation.

4.1 Training Setups

Training Stages. We adopt a three-stage training protocol to train ConvLLaVA as shown in Fig. 1 (c). The training process is categorized into three stages: (1) *Projector Initialization*. We train the fifth stage of the ConvNeXt model and the vision-language projector. We utilize caption data including ShareGPT4V-PT [6], ShareGPT4V [6], and ALLaVA captions [4], totaling approximately 2M examples. (2) *Vision-Language Pretraining*. We employ caption data including ShareGPT4V-PT [6], ShareGPT4V [6], ALLaVA [4], and a 190k open-sourced subset of VFLAN [52], amounting to 2.9M data. (3) *Visual Instruction Tuning*. We fine-tune the model with the 665k LLaVA instruction dataset [30]. In each stage, we train the model for 1 epoch with the AdamW optimizer. The cosine learning rate schedule is also applied.

Implementation Details. We utilize the LAION-2B pretrained ConvNeXt-L model as our visual encoder [17]. In the three training stages, the resolution is scaled up to a fixed value. We train

Table 5: Comparisons with different resolution multi-modality models. *The results are measured by VLMEVALKIT with official checkpoints. [†]The results are measured with the original image aspect ratio and the short side of the image is resized to 1536. [‡]OtterHD is tested with the original resolution of the images and the number of tokens varies. We mark the best performance of the 7B model **bold** and the second-best underlined.

Method	Res.	#V Tokens	LLM	MME	MMB	SEED	RWQA	MMMU	MMVet	Text	Doc	POPE
LLaVA-1.5	336	576	13B	1531	68.2	68.2	55.3	36.4	38.3	48.7*	23.7*	85.9
VILA	336	576	13B	1570	70.3	_	_	-	38.8	54.5^{*}	39.2^*	84.2
LLaVA-Next	672	2880	13B	1575	70	71.9	_	36.2	48.4	67.1*	-	86.7
OtterHD	_‡	_‡	8B	1223	58.3	_	_	_	26.3	_	_	86
Qwen-VL-Chat	448	256	7B	1488	60.6	64.8	_	-	_	61.5	62.6	_
LLaVA-1.5	336	576	7B	1510	64.3	66.2	54.8	_	30.5	45.5^{*}	21.6^*	85.9
ShareGPT4V	336	576	7B	<u>1567</u>	<u>68.8</u>	<u>69.7</u>	56.5	-	37.6	51.1^*	26.6^*	86
VILA	336	576	7B	1533	68.9	_	_	_	35.1	53.2^{*}	35.8^{*}	86.3
LLaVA-Next	672	2880	7B	1519	67.4	70.2	_	35.8	43.9	64.4^{*}	-	86.5
MiniGemini-HD	1536	2880	7B	1546	65.8	_	-	36.8	41.3	-	-	-
ConvLLaVA	768	144	7B	1541	68	68.8	55.9	<u>36.3</u>	<u>44.8</u>	59.1	44.8	<u>87.3</u>
ConvLLaVA	1024	256	7B	1553	<u>68.8</u>	69.3	<u>58.8</u>	35.1	44.4	62.5	48.5	87.7
ConvLLaVA	1536	576	7B	1575	68.7	70.2	59.9	35.8	45.9	65.8	59/ 65	<u>87.3</u>

Table 6: Results on referring expression comprehension tasks. The models in this table are trained with the same grounding data. We mark the best performance of the model **bold**.

Mathad Dag		#V Tokons	ттм	I	RefCOCO		R	RefCOCO+		RefCOCOg		Ava
Wiethou	Res.		LLIVI	val	test-A	test-B	val	test-A	test-B	val	test	Avg
LLaVA-1.5	336	576	7B	76.3	83.2	67.9	66.8	77.0	56.8	70.4	70.0	71.1
LLaVA-1.5	336	576	13B	84.0	89.5	77.1	76.3	84.3	66.1	78.8	78.3	79.3
ConvLLaVA	768	144	7B	84.5	89.0	79.2	77.7	84.9	69.7	79.8	79.7	80.6
ConvLLaVA	1024	256	7B	85.5	89.6	78.8	79.3	86.1	70.3	80.6	81.2	81.4
ConvLLaVA	1536	576	7B	86.5	90.6	80.5	80.0	86.8	71.5	82.0	82.4	82.3

ConvLLaVA at 768, 1024, and 1536 resolutions. The learning rates in the three training stages are 3e-4, 2e-5, and 2e-5, respectively. Meanwhile, the batch sizes are 256, 256, and 128. Training the ConvLLaVA 768 resolution model takes approximately 18 hours on 2 A800 machines. The instruction tuning costs 20 hours for LLaVA-NExT 7B on an A100 machine [31], while it tasks only 9 hours for our 1536 resolution ConvLLaVA on a single machine.

Evaluation Benchmarks. To systematically investigate the performance of our model, we include more benchmarks for evaluation, including MME [13], MMBench [33], SEEDBench [22], MMMU [58], MMVet [57], RealWorldQA [50], TextVQA [44], DocVQA [38], and POPE [26]. Our results are measured by VLMEVALKIT. We also assess the performance on grounding benchmarks, including RefCOCO [19], RefCOCO+, and RefCOCOg [37].

4.2 Quantitative Results

We perform a comprehensive comparison with state-of-the-art models on 7 different benchmarks (Tab. 5). Our model achieves consistent improvements compared to LLaVA-1.5. Our 7B model even exhibits comparable performance with LLaVA-1.5 13B and LLaVA-NExT 7B [31]. On OCR benchmarks like TextVQA and DocVQA, our model outperforms the LLaVA-1.5 7B and 13B models. Since OCR benchmarks are sensitive to resolution, our ConvLLaVA series models demonstrate consistent improvement on TextVQA and DocVQA with higher resolution, showcasing the effectiveness of scaling up resolution. Notably, our model surpasses Qwen-VL-Chat on DocVQA which has millions of document training data. While there is only a limited number of document data in our training dataset. This shows the benefits of the high-resolution design of our model. ConvLLaVA outperforms LLaVA-NExT on MMBench, TextVQA, POPE, and MMVet.

For grounding benchmarks, our model and LLaVA are trained with the same set of grounding data. The comparison between them is fair. On RefCOCO, RefCOCO+, and RefCOCOg, ConvLLaVA exhibits consistent improvement when increasing resolution (Tab. 6). ConvLLaVA outperforms LLaVA-7B and 13B model on all 8 test splits. This demonstrates the benefits of higher resolution for grounding tasks. Our 7B model also surpasses 13B LLaVA model on all 8 benchmarks.

4.3 Understanding Any Aspect Ratio Images and Highre Resolutions

Thanks to the translation equivalence of convolution neural network, our model could be trained on a fixed resolution but inference on higher resolution and with an arbitrary aspect ratio. We test such ability on our 1536 resolution model ConvLLaVA.

The original image preprocessing process is padding the image to a square, resizing the image to 1536, and center cropping [30]. We canceling padding and center cropping. Hence, the short side of the image should just be resized to 1536 and keep the original aspect ratio. This is the setting of how we test images of any aspect ratio. The results are shown in Tab. 7. We observe that on the general benchmark, SEEDBench, the performance slightly decreases. On OCR benchmarks, especially on DocVQA, the performance is improved. The reason for this we think

Table 7: Results for different aspect ra-
tios and higher resolution.

Input Shape	SEED	Text	Doc
(1536, 1536)	70.2	65.8	59.0
short side=1536	68.9	64.6	65.0
short side=1664	67.3	64.2	65.7

is that the image aspect ratio in DocVQA is not 1:1, forcely transforming the image into a square would lower the resolution of the image.

We also test ConvLLaVA when resizing the short side of images to 1664 resolution which is higher than its pretrained 1536 resolution. We observe that on DocVQA the performance could be further improved to 65.7.

4.4 Discussions

Architectures and data. While we have demonstrated the effectiveness of our method, there remains room for further improvement. The ConvNeXt architecture we use is tailored for low-resolution image understanding (e.g., 256), with a kernel size of 7 optimized for such resolutions. However, as the resolution increases to 1536, the relatively small kernel size may limit the model capacity when the resolution is extremely high. Besides, the number of layers in the ConvNeXt four stages (3, 3, 27, 3) is designed for a 4-stage model and may not be optimal for our 5-stage model. Therefore, a potential future direction could involve designing a five-stage, linear spatial complexity, hierarchical high-resolution vision encoder. We emphasize the critical role of the five-stage visual encoder since it is fit for high-resolution LMM. It compresses visual features by $64 \times$, greatly reducing the redundancy in its visual tokens. In contrast, four-stage visual encoders, designed for traditional computer vision tasks, output excessive tokens when resolution is high.

Linear spatial complexity and information compression. We identify *linear spatial complexity* and *information compression* procedure as two critical properties for future visual encoders of LMMs. These properties ensure the efficiency of both the visual encoder and the LLM, respectively. Furthermore, they are crucial for multi-image, interleaved image and text, and video understanding tasks, as these tasks commonly result in numerous visual tokens. We anticipate that future research will focus more on these two directions to further advance the research of LMMs.

Trade-off between compression and retrieval for high-resolution understanding. Our method, ConvLLaVA, compresses a 1536-resolution image to 576 visual tokens with a $64 \times$ compression ratio. While concurrent work [11, 7] explores retrieving fine-grained image information from long visual token sequences. In the context of high-resolution image understanding, compressing visual information maintains computational efficiency, but excessive compression may lead to information loss. Conversely, retaining a large number of visual tokens avoids information loss but sacrifices efficiency and challenges the retrieval capabilities of LLMs. Consequently, a trade-off emerges

between visual information compression and retrieval capabilities for high-resolution understanding. Future research should explore an optimal balance between these two factors.

5 Conclusion

In this paper, we have critically examined the limitations of the visual encoder for current LMMs: quadratic spatial complexity and numerous visual tokens. The excessive visual tokens are the more fundamental problem. These drawbacks hinder LMMs from efficiently understanding high-resolution images. Consequently, we propose ConvLLaVA, whose visual encoder is a hierarchical backbone, ConvNeXt, to mitigate this issue. ConvLLaVA compresses high-resolution visual information into information-rich visual representation rather than preserving all the redundancy in the visual representation. Extensive experimental results have demonstrated the efficacy of our proposed method. Our 7B parameter model exhibits superior performance compared to the LLaVA-1.5 13B model. Furthermore, our method is flexible in encoding images with arbitrary shapes and resolutions. Our work highlights the advantages of hierarchical visual backbones for LMMs, addressing critical challenges while maintaining simplicity and efficiency.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants 62321005 and 62276150.

References

- [1] Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2023.
- [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.
- [3] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Localityenhanced projector for multimodal llm. *arXiv preprint arXiv:2312.06742*, 2023.
- [4] Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4vsynthesized data for a lite vision-language model. arXiv preprint arXiv:2402.11684, 2024.
- [5] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. arXiv preprint arXiv:2403.06764, 2024.
- [6] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. arXiv preprint arXiv:2311.12793, 2023.
- [7] 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.
- [8] Sijie Cheng, Zhicheng Guo, Jingwen Wu, Kechen Fang, Peng Li, Huaping Liu, and Yang Liu. Can vision-language models think from a first-person perspective? arXiv preprint arXiv:2311.15596, 2023.
- [9] 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, 2023.
- [10] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *European conference on computer vision*, pages 74–92. Springer, 2022.
- [11] 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.
- [12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.
- [13] 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.
- [14] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.
- [15] 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.
- [16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [17] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021.

- [18] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. arXiv preprint arXiv:2402.07865, 2024.
- [19] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.
- [20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. arXiv preprint arXiv:2304.02643, 2023.
- [21] Bo Li, Peiyuan Zhang, Jingkang Yang, Yuanhan Zhang, Fanyi Pu, and Ziwei Liu. Otterhd: A high-resolution multi-modality model. arXiv preprint arXiv:2311.04219, 2023.
- [22] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seedbench: Benchmarking multimodal llms with generative comprehension. arXiv preprint arXiv:2307.16125, 2023.
- [23] 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:2301.12597, 2023.
- [24] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. arXiv preprint arXiv:2311.01378, 2023.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.
- [31] 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.
- [32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.
- [33] 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.
- [34] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision* and pattern recognition, pages 11976–11986, 2022.
- [35] 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.
- [36] Gen Luo, Yiyi Zhou, Yuxin Zhang, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. Feast your eyes: Mixture-of-resolution adaptation for multimodal large language models. arXiv preprint arXiv:2403.03003, 2024.
- [37] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In CVPR, 2016.

- [38] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, 2021.
- [39] 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.
- [40] OpenAI. Gpt-4v(ision) system card, 2023.
- [41] 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.
- [42] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018.
- [43] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems, 35:25278–25294, 2022.
- [44] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *ECCV*, 2020.
- [45] 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.
- [46] 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.
- [47] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.
- [48] Kirill Vishniakov, Zhiqiang Shen, and Zhuang Liu. Convnet vs transformer, supervised vs clip: Beyond imagenet accuracy. arXiv preprint arXiv:2311.09215, 2023.
- [49] Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Vary: Scaling up the vision vocabulary for large visionlanguage models. arXiv preprint arXiv:2312.06109, 2023.
- [50] x.ai. Grok-1.5 vision preview, 2024.
- [51] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. arXiv preprint arXiv:2309.16671, 2023.
- [52] Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, Di Jin, Yu Cheng, Qifan Wang, and Lifu Huang. Vision-flan: Scaling human-labeled tasks in visual instruction tuning. arXiv preprint arXiv:2402.11690, 2024.
- [53] Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users. *arXiv preprint arXiv:2312.13771*, 2023.
- [54] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. arXiv preprint arXiv:2310.05126, 2023.

- [55] 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.
- [56] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. Advances in Neural Information Processing Systems, 36, 2024.
- [57] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.
- [58] 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.
- [59] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.
- [60] 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 Training Visual Encoder with More Data

In Section 3.2, we observe that updating the visual encoder is essential for ConvNeXt as the standalone encoder. We compare the two visual encoders with more training data in Tab. 8. For the visual language training stage, we use ALLaVA and ShareGPT4V-PT. We train the last two stages for ConvNeXt and the last 12 layers for ViT. With more training data, ConvNeXt outperforms ViT on all the 4 benchmarks. These results validate the advantages of ConvNeXt over ViT. This ConvNeXt model even outperforms the 768-resolution ConvLLaVA model on some benchmarks due to its higher number of visual tokens. However, the training and inference speed is much slower than the 768-resolution ConvLLaVA model due to the increased number of visual tokens. The 1536 resolution ConvLLaVA, featuring outputting the same number of visual tokens, outperforms this model. This shows higher resolution model may have a higher model capacity to learn from data.

Table 8: Comparison on different visual encoders. The visual encoders are updated during training. #V Tokens stands for the number of visual tokens.

Visual Encoder	Res.	#V Tokens	MMBench	SEEDBench	TextVQA	DocVQA
ViT-L	336	576	68.1	68.0	49.6	29.2
ConvNeXt-L	768	576	68.5	69.5	62.5	51.6

B Hyperparameters for 5-stage ConvNeXt

We discuss the choice of hyperparameters in this section.

Number of Trained Stages. We conduct an ablation study to determine the optimal number of stages for vision-language pretraining at 768 resolution. We find that fine-tuning from stage 3 yields better results than fine-tuning from stage 4 (Tab. 9). While the performances of fine-tuning from stage 2 and stage 3 are comparable, we opt for fine-tuning from stage 3 due to its fewer trainable parameters.

Number of Layers in Stage 5. We ablate on the number of ConvNeXt layers in stage 5. Given that the number of layers in each stage is a multiple of 3 in ConvNeXt-L, we experiment with 3, 6, and 9 layers in stage 5. For simplicity, we perform the experiments on ConvNeXt 768. We observe a slight decrease in performance when adding 9 layers in stage 5 (Tab. 10). However, it's hard to determine whether adding 3 or 6 layers is more beneficial for these four benchmarks. Hence, we conduct experiment on the 1536 resolution to further investigate this hyperparameter (Tab. 11). The results show that adding 6 layers could be better. We opt for 6 layers in our experiments.

C Training protocol for each experiment

The detailed training hyper-parameters are shown in the following tables.

Table 9: Ablation on the number of trainable stages.

Visual Encoder	Tune from Stage	MMBench	SEEDBench	TextVQA	DocVQA
ConvNeXt-L†	2	65.1	67.7	54.8	31.1
ConvNeXt-L†	3	65.3	67.7	54.7	31.1
ConvNeXt-L†	4	66.2	67.0	52.2	28.2

Table 10: Ablation on number of layers in stage 5.

Visual Encoder	#Layers Added	MMBench	SEEDBench	TextVQA	DocVQA
ConvNeXt-L†	3	65.2	67.9	55.6	29.6
ConvNeXt-L†	6	65.3	67.7	54.7	31.1
ConvNeXt-L†	9	64.6	67.9	54.6	30.1

Table 11: Experiments on the number of layers in stage 5 on 1536 resolution.Visual Encoder #Layers in Stage 5 MMBench SEEDBench TextVQA DocVQAConvNeXt-L†364.668.460.638.8ConvNeXt-L†6664.369.160.742.5

Training Stage	1	2						
Visual Encoder								
Projector	\checkmark	\checkmark						
LLM		\checkmark						
data	LLaVA LCS-558K	LLaVA SFT 665k						
lr	1e-3	2e-5						
batch size	256	128						
lr schedule	cosine decay	cosine decay						
lr warmup ratio	0.03	0.03						
epoch	1	1						
optimizer	AdamW	AdamW						

Table 12: The training protocol for Tab. 2.

Table 13: The training protocol for Tab. 3.

Training Stage	1	2	3
Visual Encoder		1	
Projector	\checkmark	\checkmark	1
LLM		\checkmark	1
data	LLaVA LCS-558K	ShareGPT4V-PT	LLaVA SFT 665k
lr	1e-3	2e-5	2e-5
batch size	256	256	128
lr schedule	cosine decay	cosine decay	cosine decay
lr warmup ratio	0.03	0.03	0.03
epoch	1	1	1
optimizer	AdamW	AdamW	AdamW

Tuble 11. The training protocol for Tub. 1, Tub. 9, and Tub. 10					
Training Stage	1	2	3		
ConvNeXt		✓			
Stage 5	\checkmark	\checkmark			
Projector	✓	\checkmark	\checkmark		
LLM		\checkmark	\checkmark		
data	ShareGPT4V-PT	ShareGPT4V-PT	LLaVA SFT 665k		
lr	3e-4	2e-5	2e-5		
batch size	256	256	128		
lr schedule	cosine decay	cosine decay	cosine decay		
lr warmup ratio	0.03	0.03	0.03		
epoch	1	1	1		
optimizer	AdamW	AdamW	AdamW		

Table 14: The training protocol for Tab. 4, Tab. 9, and Tab. 10

Table 15: The training protocol for Tab. 5, and Tab. 6						
Training Stage	1	2	3			
ConvNeXt		1				
Stage 5	✓	✓				
Projector	✓	✓	1			
LLM		1	1			
	ShareGPT4V-PT	ShareGPT4V-PT				
data	ShareGPT4V	ShareGPT4V	LLaVA SFT 665k			
	ALLaVA Caption	ALLaVA, VFLAN				
lr	3e-4	2e-5	2e-5			
batch size	256	256	128			
lr schedule	cosine decay	cosine decay	cosine decay			
lr warmup ratio	0.03	0.03	0.03			
epoch	1	1	1			
optimizer	AdamW	AdamW	AdamW			