FastVLM: Efficient Vision Encoding for Vision Language Models

Supplementary Material

A. Training Setup

For experiments presented in Tab. 1, Tab. 2, Tab. 4, Tab. 5, we perform 2-stage training with the hyperparameters listed in Tab. 7. The model is trained for a single epoch in all the stages. Note, in Tab. 5, we do not re-train other token pruning works, we simply report the performance of the respective methods as they adhere to the 2-stage training setup described in Tab. 7, which was originally introduced in LLaVA-1.5 [49].

To showcase our model's performance in the presence of additional dataset, we scale both pretraining and instruction tuning datasets in Sec. 4. For results presented in R13, R17, R18, R25, R26 in Tab. 6, we still perform 2-stage training described in Tab. 7, for R18 and R26, we use instruction tuning dataset of size 1.1 million samples in Stage-2. For results presented in R3, R4, R6, R7, R10, R11, R14, R19, R20, R21, R27, R28, R38 and R39, we scale-up both instruction tuning dataset and pretraining dataset. We also introduce and additional stage of pretraining with the scaled-up dataset as described in Tab. 8. Details of 1.1 million, 6.5 million and 11.9 million instruction tuning dataset is presented in Sec. D.

	Stage-1	Stage-2
Data	LLaVA-1.5 558K	LLaVA-1.5 665k
Learning Rate	1e-3	2e-5
Batch size	256	128
LR. schedule	cosine decay	cosine decay
LR. warmup ratio	0.03	0.03
Optimizer	AdamW	AdamW
Trainable modules	Projector	Full Model

Table 7. 2-Stage training setup used in ablations for Sec. 3.

	Stage-1	Stage-1.5	Stage-2
Data	LLaVA-1.5 558K	Recap-CC3M + Recap-CC12M [39]	1.1M / 6.5M / 12.5M
Learning Rate	1e-3	2e-5	2e-5
Batch size	256	128	128
LR. schedule	cosine decay	cosine decay	cosine decay
LR. warmup ratio	0.03	0.03	0.03
Optimizer	AdamW	AdamW	AdamW
Trainable modules	Projector	Full Model	Full Model

Table 8. 3-Stage training setup used for results with scaled-up data in Tab. 6.

B. Architecture Details

The patch embedding layers shown in Fig. 2, consists of 7×7 depthwise convolutions with [76] style train-time over-parameterization, followed by 1×1 pointwise convolution. The stride for 7×7 depthwise convolution is set to 2 in order to downsample the input tensor. In [78], squeeze-excite layers were incorporated into this block; however, we found them to negatively impact inference latency, especially for high image resolutions, so we opted not to include them in our model. We use the same ConvFFN layer defined in [77], i.e. 7×7 depthwise convolutions preceding a typical FFN layer. The stem downsamples the input tensor by factor of 4 on each side, and each patch embedding layer downsamples the input tensor by a factor 2. Although recent architectures like ViTamin [11] recommend an overall downsampling factor of only 16, FastViTHD incorporates an additional patch embedding layer compared to FastViT, resulting in an overall downsampling factor of 64× for the input tensor. In each stage, we increase the number of channels by a factor of 2 as done in FastViT and other convolutional and hybrid transformer architectures. This results in a Stage-5 with the widest MLP layers in the architecture, performing self-attention on an input tensor which is downsampled by a factor of 64.

B.1. Naive Scaling

In order to scale the model size of FastViT, we simply increased the embedding dimensions per stage to `[128, 256, 512, 1024]`, and set the number of layers per stage to `[2, 12, 16, 6]`. Patch embedding layers in each stage use squeeze-excite layers and the MLP expansion ratio is set to 3.0, following the design in [78].

C. Additional Results

We present the performance of FastVLM on text-rich benchmarks under various training settings in Tab. 10. FastVLM surpasses MM1 and Cambrian-1 across a wide range of benchmarks by scaling up pretraining and instruction tuning datasets. This result highlights the quality of visual tokens produced by FastViTHD, as FastVLM is able to achieve these improvements with 2.8× less visual tokens than MM1 and with a vision encoder that is 5.1× smaller.

C.1. Dynamic Resolution (AnyRes) Results

From Fig. 6, it is evident that VLMs prefer visual encoding with fewer semantic breaks. Variants with more tiles typically underperform compared to those with fewer tiles and a static resolution. To further scale up input resolution,

Row Ann.	Method	Vision Encoder	LLM	Input Res.	#Visual Tokens	Vis. Enc. Size(M)↓	Vision Enc. Latency(ms)↓	LLM Prefilling(ms)↓
0.5B Model Comparison								
R1	nanoLLaVA	ViT-SO400M	Qw.1.5	384	729	430	272.1	263.3
R2	LLaVAOV [41]*	ViT-SO400M	Qw.2	1152	7290	430	2721.4	11402.4
R3	**FastVLM (Ours)**	FastViTHD	Qw.2	1024	256	**125**	**116.3**	50.5
R3	**FastVLM (Ours)***	FastViTHD	Qw.2	2048	1280	**125**	581.5	336.4
1-2B Model Comparison								
R4	MobileVLMv2 [18]	ViT-L/14	ML.	336	144	304	127.4	458
R5	**FastVLM (Ours)**	FastViTHD	Qw.2	768	144	**125**	**54.8**	97.1
R6	DeepSeekVL [54]	ViT-SO400M	DS.	384	576	430	272.1	-
R7	MM1 [61]*	ViT-H	-	1344	720	632	-	-
R8	**FastVLM (Ours)**	FastViTHD	Qw.2	1024	256	**125**	**116.3**	116.1
R8	**FastVLM (Ours)***	FastViTHD	Qw.2	2048	1280	**125**	581.5	681.7
7B Model Comparison								
R9	InstructBLIP [19]	ViT-g/14	Vic.	224	32	1012	149.5	152.1
R11	**FastVLM (Ours)**	FastViTHD	Vic.	256	16	**125**	**6.8**	143.4
R12	MobileVLMv2 [18]	ViT-L/14	Vic.	336	144	304	127.4	332.1
R13	ConvLLaVA [25]	ConvNeXT-L	Vic.	768	144	200	164.3	332.1
R14	**FastVLM (Ours)**	FastViTHD	Vic.	768	144	**125**	**54.8**	332.1
R17	**FastVLM (Ours)**	FastViTHD	Qw.2	768	144	**125**	**54.8**	391.2
R20	ConvLLaVA [25]	ConvNeXT-L	Vic.	1024	256	200	696.1	461.1
R26	LLaVA-1.5 [49]			336	576	304	127.4	1170.0
R27	MobileVLMv2 [18]	ViT-L/14	Vic.	336	576	304	127.4	1170.0
R28	ShareGPT4V [12]			336	576	304	127.4	1170.0
R29	ViTamin [11]	ViTamin-L	Vic.	384	576	333	137.6	1170.0
R30	ConvLLaVA [25]	ConvNeXT-L	Vic.	1536	576	200	1569.7	1170.0
R31	VILA [46]	ViT-L/14	L-2	336	576	304	127.4	1169.5
R33	MM1 [61]*	ViT-H	-	1344	720	632	-	-
R34	LLaVA-NeXT*	ViT-L/14	L-3	672	2880	304	637.0	19709.7
R21	**FastVLM (Ours)**	FastViTHD	Vic.	1024	256	**125**	**116.3**	461.1
R36	**FastVLM (Ours)**	FastViTHD	Qw.2	1024	256	**125**	**116.3**	524.5
R36	**FastVLM (Ours)***	FastViTHD	Qw.2	2048	1280	**125**	581.5	3139.5
VLMs with Multiple Vision Encoders and 8B LLM								
R35	MiniGemini-HD	ConvNeXT-L ViT-L/14	L-3	1536 672	2880	200 304	1569.7 552.6	19709.7
R36	Cambrian-1 [73]	ViT-SO400M ConvNeXt-XXL DINOv2-ViT-L/14 ViT-L/14	L-3	384 1024 518 336	576	430 846 304 304	272.1 2290.4 1171.5 127.4	1223.6

Table 9. **Breakdown of prefilling latencies for recent methods.** The models are grouped based on total number of visual tokens. For models that were difficult to export or unavailable, we mark them as '-' in the table. "Vic." refers to Vicuna [91], "Qw.2" refers to Qwen2 [80] and "Qw." refers to Qwen [3]. "L-2" refers to LLaMA-2. "L-3" refers to LLaMA-3. "ML." refers to MobileLLaMA [17, 18]. "DS." refers to DeepSeek LLM [20]. * For input resolution and visual tokens, we report the highest supported resolution by the respective models as some models like LLaVA-OneVision [41] and MM1 [61] use dynamic input resolution. FastVLM models using dynamic resolution employs a simple 2×2 grid, with tile size set to 1024. For VLMs that use multiple vision encoders, the size of each encoder is listed independently, for TTFT, the latency from each encoder is summed up.

we train variants of FastVLM with support for dynamic input resolution, where we use a tile size of 1024×1024 and use a simple 2×2 grid. This enables the model to process a peak input resolution of 2048×2048 using only 4 tiles, unlike models like InternVL2 [15] which uses roughly 36 tiles to process images of resolution 2688×2688. We report performance of FastVLM with dynamic resolution support on text-rich benchmarks in Tab. 10.

C.2. CVBench and MathVista Results

Results in Tab. 11 show that, in comparison to Cambrian-1 [73], FastVLM is significantly better on MathVista [56] and competitive on CVBench [73], even though we have a single backbone and significantly fewer tokens. Results on both CVBench and MathVista benchmarks further improve as we scale the SFT dataset by including LLaVA-OneVision [41].

D. Datasets

D.1. Pretraining Datasets

For Stage-1 training, we only use LLaVA-1.5 558K [49] dataset. For Stage-1.5 training, we use densely captioned versions of CC3M [66] and CC12M [10] introduced in [39]. The total size of this dataset is 15 million image-text pairs. We generated 300 generic questions, such as "What is in this photo?". For each (image, dense-caption) pair, we randomly selected a generic question to form a triplet of (ques-

Row Ann.	Method	Vision Encoder	LLM	Data (M) (PT+IT)	Input Res.	#Visual Tokens	Vis. Enc. Size(M)↓	TTFT (ms)↓	ChartQA	OCRBench	TextVQA	DocVQA	InfoVQA
0.5B Model Comparison													
R1	LLaVAOV [41]*	ViT-SO400M	Qw.2	4.5+3.2	1152	7290	430	14124	61.4	-	-	70.0	46.3
R2	**FastVLM (Ours)**	FastViTHD	Qw.2	15+12.5	1024	256	**125**	**166**	63.4	54.9	62.9	70.4	35.8
R3	**FastVLM (Ours)***	FastViTHD	Qw.2	15+12.5	2048	1280	**125**	918	**68.8**	**59.0**	**65.4**	**82.1**	**49.3**
1-2B Model Comparison													
R4	MM1 [61]*	ViT-H	-	3000+1.5	1344	720	632	-	61.8	56.6	68.2	68.4	38.5
R5	**FastVLM (Ours)**	FastViTHD	Qw.2	15+12.5	1024	256	**125**	**233**	69.6	62.9	69.0	75.6	41.7
R6	**FastVLM (Ours)***	FastViTHD	Qw.2	15+12.5	2048	1280	**125**	1263	**76.4**	**63.2**	**71.5**	**87.6**	**60.0**
7B Model Comparison													
R7	MM1 [61]*	ViT-H	-	3000+1.5	1344	720	632	-	72.6	62.6	72.8	76.8	45.5
R8	LLaVA-NeXT†*	ViT-L/14	L-3	-	672	2880	304	20347	69.5	49.0	64.6	72.6	-
R9	Cambrian-1 [73]	ViT-L/14 ViT-SO400M ConvNeXt-XXL DINOv2-ViT-L/14	L-3	2.5+7	336 384 1024 518	576	304 430 846 304	5085	73.3	62.4	71.7	77.8	-
R4	**FastVLM (Ours)**	FastViTHD	Vic.	0.5+0.6	768	144	**125**	**387**	17.1	30.0	62.9	32.9	28.7
R5	**FastVLM (Ours)**	FastViTHD	Vic.	0.5+1.1	768	144	**125**	**387**	59.1	38.4	67.5	57.3	29.7
R6	**FastVLM (Ours)**	FastViTHD	Vic.	15+1.1	768	144	**125**	**387**	65.4	45.3	69.4	65.5	32.0
R7	**FastVLM (Ours)**	FastViTHD	Qw.2	15+1.1	768	144	**125**	446	69.3	45.9	69.5	66.9	34.3
R8	**FastVLM (Ours)**	FastViTHD	Qw.2	15+11.9	768	144	**125**	446	74.2	59.0	72.8	72.0	44.3
R9	**FastVLM (Ours)**	FastViTHD	Vic.	0.5+0.6	1024	256	**125**	**577**	19.2	29.3	64.4	35.6	28.9
R10	**FastVLM (Ours)**	FastViTHD	Vic.	0.5+1.1	1024	256	**125**	**577**	61.0	38.3	67.4	62.8	32.0
R11	**FastVLM (Ours)**	FastViTHD	Vic.	15+1.1	1024	256	**125**	**577**	66.9	47.1	70.6	72.4	34.7
R12	**FastVLM (Ours)**	FastViTHD	Qw.2	15+1.1	1024	256	**125**	641	71.0	49.7	72.1	73.3	37.5
R13	**FastVLM (Ours)**	FastViTHD	Qw.2	15+6.5	1024	256	**125**	641	76.6	52.9	73.1	78.7	44.2
R14	**FastVLM (Ours)**	FastViTHD	Qw.2	15+11.9	1024	256	**125**	641	77.0	63.3	**74.8**	78.9	49.7
R15	**FastVLM (Ours)**	FastViTHD	Qw.2	15+12.5	1024	256	**125**	641	**77.5**	**65.7**	73.4	**82.7**	**51.2**
R16	**FastVLM (Ours)***	FastViTHD	Qw.2	15+12.5	1024	1280	**125**	3721	**82.4**	**67.3**	**76.6**	**92.3**	**68.3**

Table 10. **Comparison with recent methods on text-rich benchmarks.** The models are grouped based on total number of visual tokens. "-" indicates that performance was not reported in the respective paper. For the dataset column, "-" indicates that the dataset size for pretraining ("PT") or instruction tuning ("IT") is not explicitly mentioned in the respective paper. For methods that have more than 2 stages of training, we report the total samples used for all the pretraining stages as part of "PT". "TTFT" means time to first token (the sum of the vision encoder latency and the LLM prefilling time), we report latency only for models that are publicly available and in a format favorable to MLX [27] "Vic." refers to Vicuna [91], "Qw.2" refers to Qwen2 [80]. "L-3" refers to LLaMA-3. * - For input resolution and visual tokens, we report the highest supported resolution by the respective models that use dynamic input resolution. †- performance numbers reported from [73]. For VLMs that use multiple vision encoders, the size of each encoder is listed independently, for TTFT, the latency from each encoder is summed up.

Method	LLM Decoder	Data (M) (PT+IT)	Input Res.	#Visual Tokens	Latency Enc.(ms)	CVBench 2D	CVBench 3D	Math Vista
Cambrian-1	LLama3-8B	2.5+7	Mult.	576	3861.4	72.3	72.0	49.0
FastVLM	Qwen2-7B	15+6.5	1024	256	116.3	71.8	69.6	57.6
FastVLM	Qwen2-7B	15+11.9	768	144	54.8	75.9	79.3	64.6
FastVLM	Qwen2-7B	15+11.9	1024	256	116.3	76.7	80.9	64.8

Table 11. **Evaluation on CVBench and MathVista.** "6.5M" instruction tuning dataset is from Cambrian-1. "11.9" instruction tuning dataset is concatenation of Cambrian-1 and LLaVA-OneVision datasets.

tion, image, dense-caption). With a 0.5 probability, we placed the image's special token `<image>` either before or after the question. From recent works like [39, 61, 73] and our results in Tab. 6, scaling dataset in Scale-1.5 is beneficial to improve the performance of VLM across a wide range of evaluations. Even though FastViTHD is smaller than ViT-L/14 and ViT-H used in [39, 61] respectively, we see similar scaling trends.

D.2. Visual Instruction Tuning Datasets

We use 3 different version of instruction tuning datasets. The smallest scale is LLaVA-1.5 665K dataset [49]. We further scale up this dataset by including training splits of the following datasets; AI2D [33], ScienceQA [55], ChartQA [58], COCO [47], DocVQA [60], DVQA [31], GeoQA+ [8], OCRVQA [62], SegmentAnything [35], SynthDoG-EN [34], TextVQA [69] and Visual Genome [36]. The conversational data for the listed datasets is sourced from [14]. The total number of samples in this dataset is 1.1 million and is referred to as "1.1M" in all the tables. We further scale-up instruction tuning dataset using image-based conversational data from Cambrian-7M [73], which amounts to 5.4 million samples. Filtered Cambrian-7M [73] is merged with "1.1M " dataset to obtain "6.5M" instruction tuning dataset. We then append all available single-image instruction tuning

	GQA	SQA	TextVQA	POPE	LLaVA Bench^W	MMVet	VQAv2	DocVQA	Seed Bench^I
	62.69	64.25	60.71	85.8	59.4	29.6	77.27	27.57	53.31
	62.68	64.95	60.61	86.1	60.1	31.6	77.39	28.37	53.55
	62.69	65.64	60.68	85.3	61.4	31.1	77.31	28.26	53.46
Std.	0.0047	0.57	0.041	0.33	0.83	0.85	0.049	0.35	0.099

Table 12. VLM benchmarks across three independent runs with frozen FastViT image encoder. Training setup is LLaVA-1.5 with Vicuna 7B as LLM. Standard deviation across runs is listed in the bottom row.

data open-sourced by LLaVA-OneVision [41] to "6.5M" to obtain "11.9M" instruction tuning dataset. We then include roughly 0.6M samples from DocMatix [37] dataset to obtain "12.5M" instruction tuning dataset. From Tab. 6, we see further improvements in VLM benchmarks when instruction tuning dataset is scaled, following trends exhibited by image encoders much bigger than FastViTHD.

D.3. Evaluations

In addition to evaluations listed in Sec. 4, we report performance of FastVLM on ChartQA [58], OCRBench [52] and InfoVQA [59] to compare FastVLM against recent methods on text-rich benchmarks. In Tab. 12, report performance of FastViT model (with architectural interventions) from multiple training runs and compute the standard deviation of metrics reported in Tab. 6. As described in Sec. 4, for ablations we are interested in benchmarks that are quick to evaluate and exhibit lower variance to different initializations. From Tab. 12, GQA, TextVQA, POPE, DocVQA and SeedBench fit the criteria. While VQAv2 also exhibits lower variance it is substantially larger and takes long time to evaluate. The standard deviation across the selected metrics is below 0.5, so we use the average of these metrics as a reliable indicator for our analysis in Sec. 3.

E. Qualitative Analysis

We analyzed failures across benchmarks and found: In text-rich benchmarks (e.g., DocVQA, ChartQA), failures occur when text is too small or precise alignment is needed (e.g., reading tables). In majority of cases where the text is too small, simply using higher input resolution can reduce errors as shown in Tab. 14. Some cases require broader general knowledge to obtain a correct response as seen in Tab. 13, in such cases using a bigger LLM reduces failures. Some cases require reasoning about a higher resolution image, in which case using a bigger LLM can decrease failures as seen in Tab. 15. Some failures result from misjudgment, where correct responses are misclassified by the LLM judge or incorrect labels, we ignore these cases in our analysis.

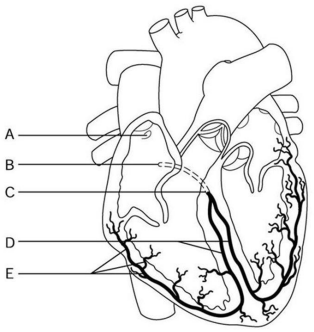
When to Use a Larger LLM	
	
Dataset: MMMU [87]	
User	What is the common term for the yellow area surrounding the site of an infection? Options: ["I don't know and I don't want to guess", 'Corona', 'Border', 'Halo', 'Toxin zone']
Ground Truth	D
FastVLM-0.5B @ 256	E ✗
FastVLM-0.5B @ 1024	E ✗
FastVLM-1.5B @ 256	D ✓
FastVLM-1.5B @ 1024	D ✓
	
Dataset: MMMU [87]	
User	The sinoatrial (SA) node is indicated by ___. Options: ['A', 'B', 'C', 'D', 'E']
Ground Truth	A
FastVLM-0.5B @ 256	E ✗
FastVLM-0.5B @ 1024	D ✗
FastVLM-1.5B @ 256	A ✓
FastVLM-1.5B @ 1024	A ✓

Table 13. Cases that require broader general knowledge, in which case a larger LLM is preferred.

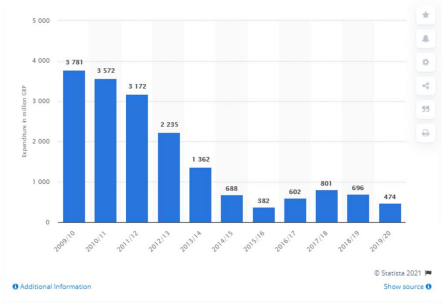
When to Use Higher Resolution	
	
Dataset: GQA [30]	
User	What is sitting inside the bowls?
Ground Truth	squash
FastVLM-0.5B @ 256	Sculpture ✗
FastVLM-0.5B @ 1024	Squash ✓
FastVLM-1.5B @ 256	Potato ✗
FastVLM-1.5B @ 1024	Squash ✓
	
Dataset: ChartQA [58]	
User	What was the highest expenditure on foreign military aid in 2009/10?
Ground Truth	3781
FastVLM-0.5B @ 256	Germany ✗
FastVLM-0.5B @ 1024	3781 ✓
FastVLM-1.5B @ 256	275 ✗
FastVLM-1.5B @ 1024	3781 ✓

Table 14. Cases where increasing the resolution is sufficient to obtain a better response.

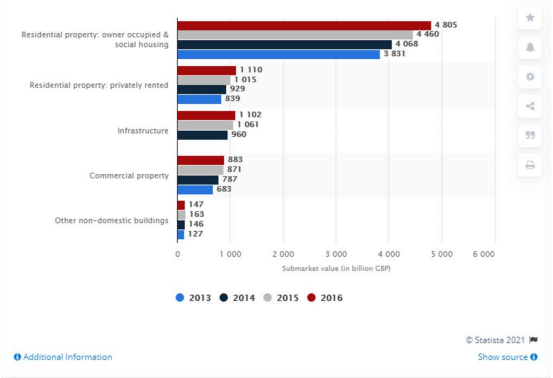
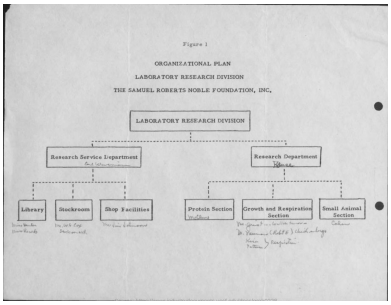
When to Use Higher Resolution and a Larger LLM	
	
Dataset: ChartQA [58]	
User	What was the value of the commercial property market in 2016?
Ground Truth	883
FastVLM-0.5B @ 256	10000 ✗
FastVLM-0.5B @ 1024	871 ✗
FastVLM-1.5B @ 256	1100 ✗
FastVLM-1.5B @ 1024	883 ✓
	
Dataset: DocVQA [60]	
User	Under which department ‘Stockroom’ is organized?
Ground Truth	Research Service Department
FastVLM-0.5B @ 256	Department of Chemistry ✗
FastVLM-0.5B @ 1024	Library ✗
FastVLM-1.5B @ 256	Research Department ✗
FastVLM-1.5B @ 1024	Research Service Department ✓

Table 15. Cases where increasing the resolution alone is not sufficient to obtain a better response. Larger LLM in combination with higher resolution is required to obtain a correct response.