A. Related Works

Autoregressive pre-training. Autoregressive modeling has been a foundational idea in machine learning and statistics for decades, long before deep learning [11]. However, it has been popularized and scaled by works such as GPT [12, 92, 93], and LLaMAs [31, 116, 117] which have demonstrated the power of autoregressive pre-training in natural language processing tasks. In vision, autoregressive principles have been applied through models like iGPT [19], which flattens images into a sequence of discretized pixels and then treats them analogously to language tokens. Similarly, Yu et al. [129] also discretize the patches with a VQGAN model [34] and then predicts them autoregressively. AIM [33] brings back the more practical continuous approach and scales to very large vision models. However, AIM still lags behind other state of the art models in sheer performance, as it uses vision-only data and requires large model capacities to perform optimally. This paper addresses these limitations by introducing multimodal pre-training in the AIMv2 family. Concurrent works [77, 104, 113, 123, 125, 126, 132] have also investigated similar multimodal autoregressive approaches that predict text and images. However, they often focus on multimodal generation quality rather than representation quality, and therefore use discrete tokens or leverage diffusion models [98] as decoders [70, 110, 111].

Pre-training in vision. For many years, the computer vision community predominantly focused on supervised pre-training [58, 97, 108], with ImageNet [61] checkpoints serving as the backbone for most visual tasks. This eventually hit a wall in terms of scalability, as labels are expensive to acquire. The community therefore focused on self-supervised methods. Earlier models used pretext tasks such as rotation prediction and patch deshuffling [39, 86, 136]. More sophisticated models like SimCLR [20], BYOL [42], SwAV [15] and DINO [16] leverage variations of contrastive learning to train models that are quasi-invariant to a broad range of image augmentations. This turns out to learn strong and general visual representations without supervision. However they require carefully handcrafted data augmentations, which also makes them computationally expensive, especially at scale. On the other hand, MAE and BEiT [8, 48] introduced masking strategies to reconstruct input data, reducing the reliance on augmentations and increasing efficiency but sacrificing performance. In practice, the best performing self-supervised vision-only models use a mixture of augmentations and masking [4, 87, 138]. Unfortunately, they are challenging to scale as they still need multiple forward passes for each training step. AIM [33] departs from these methods by employing a reconstruction-based autoregressive framework that exhibits strong scalability but requires high capacity models to attain optimal performance. Leveraging large-scale, noisy, weakly supervised datasets from the internet [13, 35, 100], an efficient paradigm emerged that aligns vision and text features through contrastive learning [54, 94]. Nevertheless, CLIP-like models require large batch sizes and meticulous dataset filtering [35, 100]. Subsequent works, such as SigLIP [134], EVA CLIP [109], and Fini et al. [37], have addressed these issues by optimizing training processes and improving data filtering [35]. Unlike these approaches, AIMv2 does not perform explicit feature space alignment but aligns training objectives through autoregressive modeling for better multimodal synergy.

Captioning. Image captioning has been extensively studied prior to the computer vision literature. Early works [56, 121, 127] focused on aligning visual features with text to generate descriptions using CNNs and RNNs. VirTex [28] and ICMLM [99] utilize captioning for visual pre-training. SimVLM [124] employs a PrefixLM approach, encoding images and partial text tokens with a multimodal encoder and decoding the remaining text. LEMON [51] scales the language model in both parameters and dataset size. Approaches such as [65, 66] combines generative captioning with discriminative contrastive objectives to enhance multimodal learning, which led to scaling to billion-parameter models [62, 67, 130]. Similarly, CapPa [118] trains a captioning model that functions as both a masked and causal decoder, and Caron et al. [17] re-purposes a captioning model for web-scale entity recognition. Different from most previous approaches, AIMv2 does not use cross-attention and treats vision and text tokens symmetrically, similar to large multimodal models (e.g. LLaVA [73], MM1 [85], and Wang et al. [122]). Additionally, AIMv2 incorporates an autoregressive image modeling loss on vision tokens, further enhancing performance beyond captioning-only methods.

B. Hyperparamters

Pre-training. We outline the optimization hyperaparmeters and data augmentations used during AIMv2 pre-training in Table B1. For the captions, we adopt the tokenizer used by SigLIP [134] and truncate any text longer than 77 tokens.

Attentive probing. The optimization and data augmentations hyperaparmeters for the attentive probing stage are detailed in Table B2. We use the same set of hyperaparmeters for all AIMv2 capacities and the baselines. To ensure a fair comparison, we sweep the learning rate and weight decay using the ranges detailed in Table B2 and report the strongest results for each model.

C. Image Recognition

Evaluation benchmarks. In Table 3, we evaluate the recognition performance of AIMv2 and other baselines on a diverse set of benchmarks that encompass fine-grained

config	ViT-L	ViTs-H	ViT-1B	ViT-3B
Optimizer	Fully decoupled AdamW [76]			
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.95$			
Peak learning rate	1e-3	8e-4	8e-4	4e-4
Minimum Learning rate	1e-5			
Weight decay	1e-4			
Batch size	8192			
Patch size	(14, 14)			
Gradient clipping	1.0			
Warmup iterations	12,500			
Total iterations	1,500,000			
Learning rate schedule	cosine decay [75]			
Augmentations:				
`RandomResizedCrop`				
`size`	224px			
`scale`	[0.4, 1.0]			
`ratio`	[0.75, 1.33]			
`interpolation`	`Bicubic`			
`RandomHorizontalFlip`	$p = 0.5$			

Table B1. Pre-training hyperparameters We detail the hyperaparmeters used for pre-training all AIMv2 variants.

config	IN-1k	Others
Optimizer	Pytorch's AdamW [76]	
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.999$	
Peak learning rate grid	[5e-5, 1e-4, 2e-4, 3e-4, 5e-4, 1e-3, 2e-3]	
Minimum Learning rate	1e-5	
Weight decay	[0.05, 0.1]	
Batch size	1024	512
Gradient clipping	3.0	
Warmup epochs	5	0
Epochs	100	
Learning rate schedule	cosine decay	
Augmentations:		
`RandomResizedCrop`		
`size`	224px	
`scale`	[0.08, 1.0]	
`ratio`	[0.75, 1.33]	
`interpolation`	`Bicubic`	
`RandomHorizontalFlip`	$p = 0.5$	
`Color Jitter`	0.3	
`AutoAugment`	`rand-m9-mstd0.5-inc1`	

Table B2. Attentive probe hyperparameters. We detail the hyperparameters used during attentive probing of AIMv2 and the baselines. For AIMv2 and the baselines we sweep over the learning rates and weight decay and report the best performance for each model.

recognition, medical imaging, satellite imagery, natural environment imagery, and infographic analysis. We detail the datasets, the splits and their sizes in Table C1.

High-resolution adaptation. In Table C2, we show the performance of AIMv2 models with varying image resolutions (224px, 336px, and 448px) across a broad set of recognition benchmarks. These results extend the main paper, which primarily focuses on the 224px resolution and the 3B model at 448px. We observe that scaling both the model capacity and image resolution leads to consistent improvements across most tasks.

Dataset	train	test	classes
Imagenet-1k [27]	1,281,167	50,000	1000
iNAT-18 [119]	437,513	24,426	8142
CIFAR-10 [60]	50,000	10,000	10
CIFAR-100 [60]	50,000	10,000	100
Food101 [10]	75,750	25,250	101
DTD [25]	3,760	1,880	47
Pets [88]	3,680	3,669	37
Cars [59]	8,144	8,041	196
Camelyon17 [7]	302,436	34904	2
PCAM [120]	262,144	32768	2
RxRx1 [112]	40,612	9854	1139
EuroSAT [49]	16,200	5400	10
fMoW [24]	76,863	19915	62
Infograph [89]	36,023	15,582	345

Table C1. Recognition benchmarks. We outline the recognition benchmarks, the number of train and test images for each dataset, and the number of categories.

Linear probing and LiT tuning. We show linear probe results in Table C3. We use GAP of the features for AIMv2. AIMv2 outperforms OAI CLIP and SigLIP. Moreover, we report more results for LiT in Table C4 where AIMv2 outperforms Cap on all benchmarks.

D. Multimodal understanding

D.1. Instruction Tuning Setup

Evaluation benchmarks. We list the multimodal benchmarks we use for assessing the performance of our models and the baselines in Table D2, together with the splits, prompts, and evaluation metric utilized for each dataset.

Hyperparamters. The hyperaparmeters used for the instruction tuning stage are detailed in Table D1. We use the same hyperaparmeters for all language decoders, AIMv2 capacities, and the baselines.

D.2. Additional Results

Instruction tuning with Cambrian. Table D3 evaluates AIMv2, fine-tuned on Cambrian, across different resolutions using a tiling strategy. Unlike the main paper, which uses Llava SFT, Cambrian offers a less in-domain data mix and achieves stronger results on text-rich benchmarks. Starting with a base resolution of 336px (matching the encoder's pretraining resolution), higher resolutions (672px and 1008px) are obtained with tiling; by splitting high-resolution images into 2×2 and 3×3 grids. AIMv2 paired with tiling shows consistent improvements on text-rich benchmarks such as InfoVQA, ChartQA, DocVQA, and TextVQA. However, on benchmarks like COCO, NoCaps, TextCaps, and MME_p, no significant gains are observed with increased resolution.

Instruction tuning with DCLM-1B decoder. In Figure D2, we present the same comparison between OAI CLIP, SigLIP, and AIMv2 as in the main paper, but this time using the Llava SFT mixture paired with a DCLM 1B decoder. These results demonstrate that AIMv2 consis-

model	architecture	IN-1k	iNAT-18	Cifar10	Cifar100	Food101	DTD	Pets	Cars	CAM17	PCAM	RxRx1	EuroSAT	fMoW	Infographic
AIMv2 $_{224px}$	ViT-L/14	86.6	76.0	99.1	92.2	95.7	87.9	96.3	96.3	93.7	89.3	5.6	98.4	60.7	69.0
	ViT-H/14	87.5	77.9	99.3	93.5	96.3	88.2	96.6	96.4	93.3	89.3	5.8	98.5	62.2	70.4
	ViT-1B/14	88.1	79.7	99.4	94.1	96.7	88.4	96.8	96.5	94.2	89.0	6.7	98.8	63.2	71.7
	ViT-3B/14	88.5	81.5	99.5	94.3	96.8	88.9	97.1	96.5	93.5	89.4	7.3	99.0	64.2	72.2
AIMv2 $_{336px}$	ViT-L/14	87.6	79.7	99.1	92.5	96.3	88.5	96.4	96.7	93.8	89.4	6.7	98.4	62.1	71.7
	ViT-H/14	88.2	81.0	99.3	93.6	96.6	88.8	96.8	96.4	93.3	89.4	7.2	98.7	63.9	73.4
	ViT-1B/14	88.7	82.7	99.4	93.9	97.1	88.9	96.9	96.5	94.2	89.5	8.4	98.9	65.1	73.7
	ViT-3B/14	89.2	84.4	99.5	94.4	97.2	89.3	97.2	96.6	93.2	89.3	8.8	99.0	65.7	74.0
AIMv2 $_{448px}$	ViT-L/14	87.9	81.3	99.1	92.4	96.6	88.9	96.5	96.6	94.1	89.6	7.4	98.6	62.8	72.7
	ViT-H/14	88.6	82.8	99.4	93.6	97.0	88.9	96.8	96.5	93.4	89.6	7.8	98.7	64.8	74.5
	ViT-1B/14	89.0	83.8	99.4	94.1	97.2	88.9	97.1	96.6	93.5	89.9	9.2	99.1	65.9	74.4
	ViT-3B/14	89.5	85.9	99.5	94.5	97.4	89.0	97.4	96.7	93.4	89.9	9.5	98.9	66.1	74.8

Table C2. Frozen trunk evaluation for recognition benchmarks, high resolution AIMv2 models. We report the recognition performance of the AIMv2 high resolution family of models when compared to the base 224px models shown in the main paper. All models are evaluated using attentive probing with a frozen backbone.

	AIMv2	OAI CLIP	SigLIP
IN-1k	**85.2**	84.6	84.4

Table C3. Linear probe on IN-1k.

Model	Samples	Imagenet	C10	C100	Food101	Pets	Cars	DTD
Cap	2B	75.0	96.9	82.7	90.9	91.1	90.1	58.1
AIMv2	2B	**75.3**	**97.4**	**83.5**	**90.9**	**91.6**	**90.5**	**59.8**

Table C4. LiT results for Cap and AIMv2 across multiple datasets.

config	Llava SFT mixture	Cambrian
Optimizer	Pytorch's AdamW [76]	
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.999$	
Decoder peak learning rate	1e-5	2e-5
Connector peak learning rate	8e-5	1.6e-4
Minimum Learning rate	0	
Weight decay	0.01	
Batch size	128	512
Gradient clipping	1.0	
Warmup iterations	250	700
iterations	5000	14,000
Learning rate schedule	cosine decay	
Transformations	`[PadToSquare, Resize]`	

Table D1. Instruction tuning hyperaparmeters. We detail the hyperparameters of the instruction tuning stage, both for the Llava SFT mixture [73] and Cambrian [115].

tently outperforms the baselines, regardless of the decoder's capacity. Notably, in the practical setting of a small decoder, AIMv2 maintains its position as the preferred choice for multimodal understanding tasks.

D.3. Qualitative Results

The qualitative results in Figure D1 highlight AIMv2's strengths on multimodal evaluations compared to SigLIP [134] and OAI CLIP [94] after instruction tuning on Cambrian. In the first three examples, AIMv2 excels in text-rich tasks by correctly localizing and extracting the relevant textual information. For instance, in the example on the left, AIMv2 is able to identify the correct value for "supreme gasoline" and outputs the correct operation for finding the solution ("Divide 50 by 3.65"). This contrasts

Benchmark	Split	Prompt	Evaluation Metric
VQAv2 [41]	Val		Accuracy
GQA [52]	Val		Accuracy
OKVQA [81]	Val		Accuracy
TextVQA [106]	Val	*Answer the question using a single word or phrase.*	Accuracy
DocVQA [83]	Val		ANLS
InfoVQA [84]	Val		ANLS
ChartQA [82]	Test		Relaxed accuracy
SEED [64]	Test (image split)		Accuracy
ScienceQA [78]	Test (image split)	*Answer with the option's letter from the given choices directly.*	Accuracy
MME [38]	Test (image split)		Accuracy
COCO [71]	Val	*Provide a one-sentence caption for the provided image.*	CIDEr
TextCaps [105]	Val		CIDEr
NoCaps [2]	Val		CIDEr

Table D2. Multimodal benchmarks. We provide the list of benchmarks used during the multimodal evaluation including the reference, split, prompt, and the evaluation metric.

with OAI CLIP and SigLIP, which provide generic and incomplete answers that fail to focus on the relevant information. Similarly, AIMv2 successfully identifies the license plate number ("AED-632") in a blurry image, demonstrating robust localization and reading capabilities in challenging settings. In the luggage example, AIMv2 accurately reads the weight ("30.7"), despite the presence of multiple distracting objects in the image, while the other models make mistakes. Finally, in the calorie estimation example, AIMv2 provides a more plausible response ("1000 calories") based on its knowledge, whereas SigLIP and OAI CLIP offer less contextually plausible answers.

E. Detection, Segmentation and Grounding

E.1. Open Vocabulary Detection and Grounding

Performance on Small Objects. In Table E1 we report the breakdowns of COCO between classes that are either *small*, *medium*, or *large*. We can observe that AIMv2 consistently outperforms on the *small* classes by +0.5 AP, compared to DFN-CLIP, the second best performing model in that breakdown. This is further emphasized by the results reported on LVIS val, as objects in LVIS are more likely to

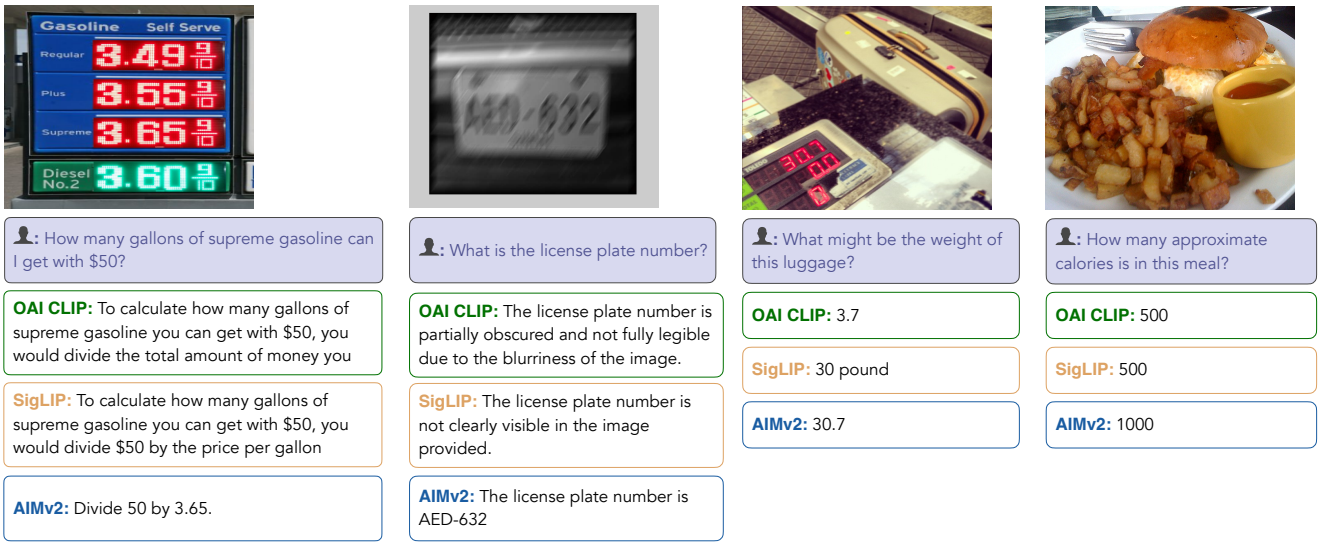


Figure D1. Qualitative comparison of AIMv2, SigLIP, and OAI CLIP on multimodal tasks after instruction tuning on Cambrian. AIMv2 demonstrates superior performance in both text-rich (e.g. extracting relevant information or reading text in cluttered scenes) and knowledge-based scenarios (e.g., estimating caloric content), showcasing its ability to focus on relevant information, accurately localize text, and provide contextually appropriate answers.

data mix	decoder	resolution	VQAv2	GQA	OKVQA	TextVQA	DocVQA	InfoVQA	ChartQA	ScienceQA	COCO	TextCaps	NoCaps	MME_p
Cambrian	Llama 3.0	336px	75.5	71.5	61.1	58.3	50.2	35.1	51.7	78.7	95.5	82.3	98.1	1594
Cambrian	Llama 3.0	672px	77.5	72.8	62.0	69.1	76.4	48.3	64.7	79.4	92.6	80.6	95.4	1482
Cambrian	Llama 3.0	1008px	77.7	73.2	62.0	72.2	79.2	53.5	65.1	81.6	93.7	81.6	97.6	1507

Table D3. Additional multimodal evaluations. We report the performance of AIMv2 using the Cambrian [115] SFT data mixture for different image resolutions (336px, 672px and 1008px).

	COCO				LVIS Val			
Model	AP_{all}	AP_s	AP_m	AP_l	AP_{all}	AP_r	AP_c	AP_f
OpenAI CLIP	59.1	43.5	63.5	74.8	31.0	17.6	**27.2**	41.2
DFN-CLIP	59.8	44.0	63.8	75.3	30.7	17.2	26.4	41.5
SigLIP	58.8	41.7	62.8	75.7	30.5	16.5	26.5	41.1
DINOv2	60.1	43.7	64.2	**75.8**	30.8	**18.5**	26.1	41.4
AIMv2	**60.2**	**44.5**	**64.3**	75.4	**31.6**	18.0	27.0	**42.8**

Table E1. Performance on OVD Benchmarks. We report the performance on mean average precision (AP) for COCO and LVIS. For COCO, we also report AP for the *small*, *medium*, and *large* subsets, while for LVIS, we report on *rare*, *medium*, and *frequent* subsets.

Model	Window Size	COCO AP_{all}	LVIS Val AP_{all}	RefCOCO Val P@1	RefCOCO+ Val P@1	RefCOCOg Val P@1
DINOv2	16	60.1	30.8	92.2	85.9	**89.1**
AIMv2		**60.2**	**31.6**	**92.6**	**86.3**	88.9
DINOv2	24	59.6	29.6	92.1	85.0	88.7
AIMv2		**59.8**	**31.2**	**92.3**	**85.8**	**89.1**
DINOv2	32	60.2	30.7	**92.5**	86.1	**89.5**
AIMv2		**60.3**	**32.9**	**92.5**	**86.3**	88.9
DINOv2	37	60.2	31.1	92.2	85.9	88.4

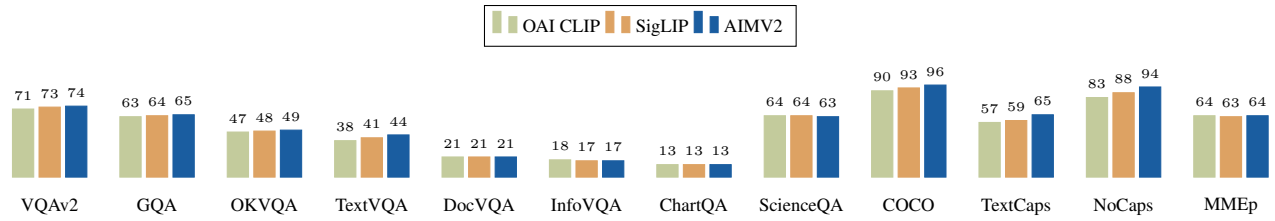
Table E2. Ablation across window sizes. We report the performance on mean average precision (AP) for COCO and LVIS. For RefCOCO* we report Precision @1 on the respective validation splits.

be small. There we observe an improvement of +1.3 AP on the *frequent* subset against DFN-CLIP.

Window Size Ablation. Due to varying input resolutions and feature map sizes used during pre-training, we ablate the effect of window size [68] for AIMv2 and DINOv2 in Table E2. For AIMv2 the input image resolution is scaled during pre-training such that the feature map size matches the window size during finetuning, while for DINOv2 the window size is fixed to match AIMv2. For comparison we also add DINOv2 trained with a window size of 37, which matches its pre-training feature map size. Across the window sizes, AIMv2 outperforms DINOv2 across all OVD and for two out of three referring comprehension benchmarks. When comparing our best performing AIMv2 with the best performing DINOv2 across all benchmarks, we observe that AIMv2 strongly outperforms on LVIS Val while outperforming on all except one benchmark against DINOv2.

E.2. Detection and Segmentation via ViTDet Mask-RCNN

To compare vision only capabilities of the encoders we incorporate them into a Mask-RCNN[47] detection model as backbones by utilizing a ViTDet formulation to accommodate for high resolution (1024) detector training / testing input size. We ensure that ViTDet [68] backbone forward pass



(a) DCLM 1B + Llava SFT mixture

Figure D2. Instruction with a small decoder (DCLM). Performance comparison of OAI CLIP, SigLIP, and AIMv2 across 12 multimodal benchmarks using the Llava SFT mixture paired with a DCLM 1B decoder. AIMv2 exhibits superior performance across most benchmarks, even with the constrained capacity of a small decoder.

	detection mAP50:95				mask mAP50:95			
Model	AP_{all}	AP_s	AP_m	AP_l	AP_{all}	AP_s	AP_m	AP_l
OAI CLIP	53.6	37.2	58.5	69.2	46.7	26.6	50.9	66.2
DFN-CLIP	53.4	37.1	58.3	69.3	46.2	26.4	50.8	66.4
SigLIP	53.3	37.2	57.6	69.7	46.6	27.1	50.5	66.3
DINOv2	**55.5**	**39.5**	**59.9**	**70.6**	**48.3**	**29.4**	**52.3**	**67.4**
AIMv2	54.0	37.4	58.8	70.0	46.7	26.7	51.1	66.5

Table E3. COCO17 detection and segmentation benchmarks. We report overall detection and segmentation scores along with the *small*, *medium*, and *large* subset breakdowns.

outputs match the respective ViT-L implementations before the training. We utilize the same set of hyperparameters for training all compared detectors: consistent windowed attention size (16) ensuring comparable compute, AdamW optimizer, cosine decay learning rate schedule, layer-wise learning rate, and weight decay. All detectors are fine-tuned on coco17 train split for 100 epochs with a global batch size of 64 following the default recipe from MMDetection [18]. We report results from the coco17-val split in Table E3. AIMv2 consistently outperforms encoders pre-trained on contrastive objectives, falling slightly behind DINOv2 which provides the strongest performance.