EVA-02: A Visual Representation for Neon Genesis

Yuxin Fang[2,1] Quan Sun[1] Xinggang Wang[2] Tiejun Huang[1] Xinlong Wang[1] Yue Cao[1]

[1]Beijing Academy of Artificial Intelligence [2]Huazhong University of Science and Technology

Fight together with Asuka at `baaivision/EVA/02`

Abstract

*We launch **EVA-02**, a next-generation Transformer-based visual representation pre-trained to reconstruct strong and robust language-aligned vision features via masked image modeling. With an updated plain Transformer architecture as well as extensive pre-training from an open & accessible giant CLIP vision encoder, **EVA-02** demonstrates superior performance compared to prior state-of-the-art approaches across various representative vision tasks, while utilizing significantly fewer parameters and compute budgets. Notably, using exclusively publicly accessible training data, **EVA-02** with only **304M** parameters achieves a phenomenal **90.0** fine-tuning top-1 accuracy on ImageNet-1K val set. Additionally, our **EVA-02**-CLIP can reach up to **80.4** zero-shot top-1 on ImageNet-1K, outperforming the previous largest & best open-sourced CLIP with only ~1/6 parameters and ~1/6 image-text training data. We offer four **EVA-02** variants in various model sizes, ranging from 6M to 304M parameters, all with impressive performance. To facilitate open access and open research, we release the complete suite of **EVA-02** to the community.*

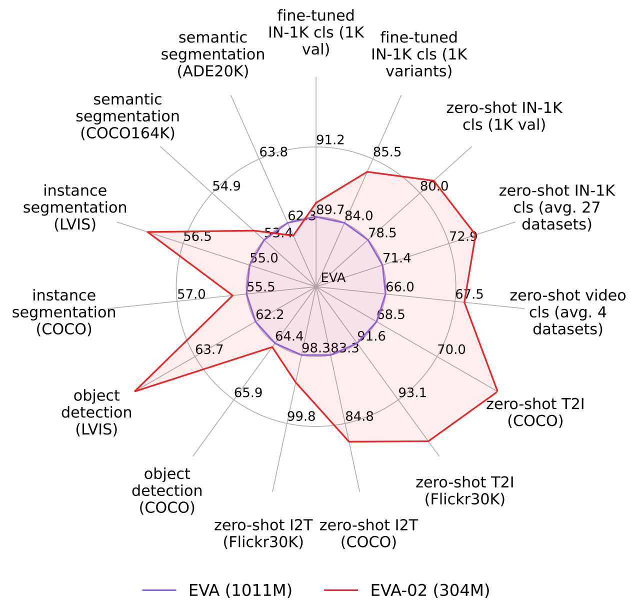


Figure 1: Qualitative comparisons between **EVA-02** (#params: 304M) and **EVA** (#params: 1011M) [45] pre-trained representations. **EVA-02** with only 304M pre-trained representations pulls off a "giant-killing" act against the previous state-of-the-art **EVA**.
∗ Notice that the scale of each axis in the radar chart is normalized by the performance of **EVA**, and the stride of each axis are the same

1. Introduction

Recent research advancements have led to a surge of interest in scaling up vision [81, 45, 124, 18] as well as vision-language [140, 123, 31, 139] representations. These efforts are driven by the belief that increasing the number of parameters, data, and compute budgets will ultimately result in improved performance [63, 142, 134, 93].

However, there is an increasing gap in computer vision between large-scale models that achieve state-of-the-art performance and models that are affordable for the wider research community. Training, tuning, and evaluating very large vision models requires significant computational resources, which can be prohibitively expensive and time-consuming. This usually leads to large-scale visual representations being trained in a few-shot or even single-shot manner, limiting the ability to fully optimize the entire process. In addition, the study of state-of-the-art representations is frequently conducted using huge amounts of infrastructure and web-scale private training data [142, 3, 27, 39], which makes it difficult to evaluate the effects of modeling advancements in a feasible and transparent way, and restricts access to a broad range of researchers and practitioners. These challenges highlight a pressing need for a more efficient and accessible approach of training and evaluating state-of-the-art vision as well as vision-language representations.

In this work, we present **EVA-02**, a series of robustly optimized plain Vision Transformers (ViTs) [118, 42] with moderate model sizes that are equipped with transferable bidirectional visual representations [41, 80] learned from a strong CLIP [95, 45] vision encoder via masked image modeling (MIM) pre-training [5]. Compared with current

		zero-shot evaluation with EVA-CLIP			**transfer learning**							
		image cls		video cls	e2e ft image cls		object det		instance seg		semantic seg	
	enc.	IN-1K	27 avg.	4 avg.	IN-1K	variants avg.	COCO	LVIS	COCO	LVIS	COCO164K	ADE20K
method	#params	(Table 10)	(Table 9)	(Table 11)	(Table 7)	(Table 6)	(Table 14)	(Table 14)	(Table 14)	(Table 14)	(Table 16)	(Table 16)
EVA [45]	1011M	78.5	71.4	66.0	89.7	84.0	64.4	62.2	55.5	55.0	53.4	**62.3**
EVA-02-L	304M	**80.4**	**73.5**	**67.7**	**90.0**	**85.2**	**64.5**	**65.2**	**55.8**	**57.3**	**53.7**	62.0
△	-707M	+1.9	+2.1	+1.7	+0.3	+1.2	+0.1	+3.0	+0.3	+2.3	+0.3	-0.3

Table 1: **Quantitative summary of EVA-02-L's performance on various mainstream vision benchmarks.**

leading vision models with billions of parameters [81, 45, 124, 18], these **EVA-02** variants require far fewer compute budgets and resources to investigate, allowing for an in-depth exploration of often-overlooked aspects.

Our empirical investigation indicates that the smaller-sized plain ViTs are highly capable, and their potential has been significantly underestimated. By leveraging the latest plain Transformer architecture design [38, 110, 113, 122] borrowed from language models, as well as thorough MIM pre-training from a publicly available giant **EVA**-CLIP [45] vision encoder, **EVA-02** is able to achieve superior performance compared to prior state-of-the-art approaches with much larger model sizes on various visual tasks.

Remarkably, using exclusively 38 million publicly accessible data, the small-sized variant of **EVA-02** with only 22M parameters achieves 85.8 fine-tuning top-1 accuracy on ImageNet-1K (IN-1K) val set [105], while the large model with only **304M** parameters achieves an outstanding **90.0** fine-tuning top-1 accuracy. Moreover, we show that initializing the image encoder of a CLIP via MIM pre-trained **EVA-02** representations can reach up to **80.4** zero-shot top-1 on IN-1K val, outperforming the previous largest & best open-sourced CLIP-Giant [1] with only ~1/6 parameters and ~1/6 image-text training data. **EVA-02** also achieves state-of-the-art performances on other representative vision tasks such as object detection and instance segmentation on LVIS [50] (65.2 AP^{box} & 57.3 AP^{mask} on `val`) and COCO [78] (64.5 AP^{box} & 55.8 AP^{mask} on `test-dev`), as well as semantic segmentation on COCO-stuff-164K [17] (53.7 mIoU^{ss}) and ADE20K [147] (61.7 mIoU^{ss} and 62.0 mIoU^{ms}). For a quantitative summary of **EVA-02**'s performance, please refer to Table 1.

arch.	norm	init.	FFN	pos. embed.	IN-1K ft top-1 acc.
	base-sized model (86M), IN-1K ft number of tokens = 196				
	pre-LN	BEiT	MLP	abs. PE	84.0 (∗)
	pre-LN	JAX	MLP	abs. PE	84.0
	pre-LN	BEiT	SwiGLU	abs. PE	83.9
	pre-LN	JAX	SwiGLU	abs. PE	85.0
	sub-LN	JAX	SwiGLU	abs. PE	85.2
TrV	sub-LN	JAX	SwiGLU	2D RoPE	**85.6** (†)
	sub-LN	JAX	SwiGLU	2D rel. PE	✗
	post-LN	JAX	SwiGLU	RoPE	✗

Table 2: **From ViT to TrV.** All experiments are conducted with the base-sized plain ViT (macro architecture: depth=12, width=768, #heads=12) with 300-epoch MIM pre-training on IN-1K. The MIM objective is to reconstruct the masked-out **EVA**-CLIP vision features conditioned on visible image patches. "✗": unstable or diverged pre-training.

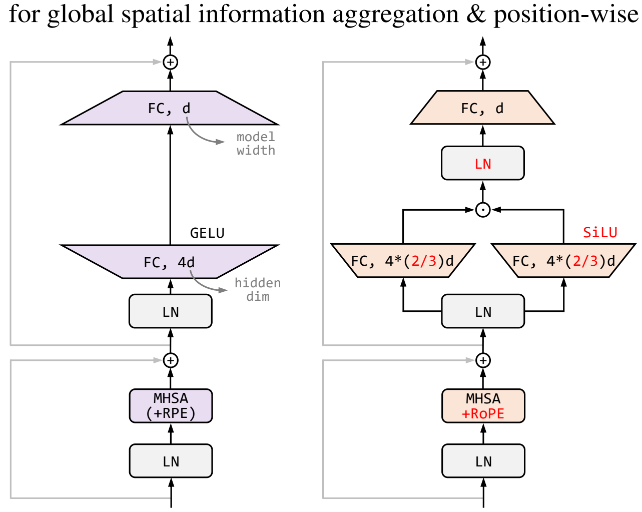
The proposed **EVA-02** series offers a diverse range of model sizes, ranging from 6M to 304M parameters, each demonstrating exceptional performance. The aim of this work is not necessarily to propose a novel method, but strive to identify a robust and effective recipe for making state-of-the-art models more affordable in practice. By providing a more accessible and performant option, **EVA-02** democratizes access to state-of-the-art vision models, allowing researchers as well as practitioners to conduct high-quality research without the need for extensive infrastructure or resources. We hope our efforts enable a broader range of the research community to advance the field in a more efficient and equitable manner.

2. Approach

The aim of **EVA-02** is to introduce a next-generation Transformer-based visual representation that achieves strong performances with moderate model sizes. To achieve this goal, our representation instrumentality project consists of two parts: architectural improvements made to the plain ViT in §2.1, as well as our MIM pre-training strategy in §2.2.

2.1. Architecture

At a high level, *plain* ViT along with its variants comes with interleaved multi-head self-attention (MHSA) layers for global spatial information aggregation & position-wise



(a) Vision Transformer (ViT) (b) Transform Vision (TrV)

Figure 2: **An illustration of ViT and TrV blocks. TrV** builds upon the original plain ViT architecture [42] and includes several enhancements: SwinGLU FFN, sub-LN, 2D RoPE, and JAX weight initialization. To keep the parameter & FLOPs consistent with the baseline, the FFN hidden dim of SwiGLU is 2/3× of the typical MLP counterpart.

arch.	MIM teacher	pt dataset	pt epochs	IN-21K intermed. ft	IN-1K ft top-1 acc.
(a) base-sized model (86M), IN-1K ft number of tokens = 196					
ViT-B	VQKD-B [92]	IN-1K	300 (0.2M-step)	✗	85.0
ViT-B	CLIP-B [95]	IN-1K	300 (0.2M-step)	✗	85.0
ViT-B	EVA-CLIP [45]	IN-1K	300 (0.2M-step)	✗	84.0 (∗)
TrV-B	EVA-CLIP [45]	IN-1K	300 (0.2M-step)	✗	85.6 (†)
(b) base-sized model, **longer** pre-training					
ViT-B	VQKD-B [92]	IN-1K	1600 (1M-step)	✗	85.5
TrV-B	EVA-CLIP [45]	IN-1K	1600 (1M-step)	✗	86.8
(c) base-sized model, **longer** pre-training & **larger** dataset					
ViT-B	VQKD-B [92]	IN-1K	1600 (1M-step)	90 epochs, 224^2	86.5
TrV-B	EVA-CLIP [45]	IN-21K	150 (1M-step)	✗	87.0

Table 3: **MIM target representations.** When pre-trained with sufficient compute budgets and data, learning from a giant EVA-CLIP can bring about considerable performance improvement compared with smaller CLIP teachers.

feedforward networks (FFNs) for feature transformation, *without* downsampling layers and multi-stage design [118, 42, 115]. This makes it an ideal testbed for representation learning due to its minimal visual structure prior and biases, as well as its natural compatibility with masked modeling, which is proven to be a simple, strong, and scalable pre-training approach [5, 92, 123, 45]. Pre-trained plain ViT can also be successfully adapted to challenging vision tasks that require high-resolution inputs & multi-scale representations with feasible costs [75, 46].

Although the inner-block micro architecture of plain ViT has continuously evolved since its inception in the year 2020 [109, 117], we notice that some significant architectural advances in language models have not yet been explored in the context of visual representation learning. These include gated linear unit [38, 110] with sigmoid linear unit (SiLU) [56] / swich activation [99] (**SwiGLU**) as the feedforward network, **sub-LN** [4, 122] as the normalization layer, and 2D rotary position embedding (**RoPE**) [113] for positional information injection.

In Table 2 we conduct a series of pilot experiments studying these architectural modifications[1]. The pretext task is to regress the masked-out EVA-CLIP vision features conditioned on visible image patches using IN-1K training images for 300 epochs, and the evaluation is done by fine-tuning the pre-trained base-sized models on IN-1K. Starting with the baseline ViT configurations used in the original BEiT series pre-training [5, 92, 123] (∗ in Table 2), we progressively refine the model design and make the following observations: (i) The performance of SwiGLU FFN is mediocre with the random weight initialization method used in BEiT, but works quite well with JAX weight initialization [14, 51] (+1.1). (ii) sub-LN slightly improves the performance compared with pre-LN (+0.2). (iii) 2D RoPE can improve the performance (+0.4), while the standard relative position embedding [109, 5, 92] suffers from unstable pre-training with

[1]More technical details can be found in the Appendix. All these modifications do not bring additional parameters as well as FLOPs.

method	IN-21K intermed. ft?	IN-1K ft img size	IN-1K ft top-1 acc.	IN-V2 ft top-1 acc.
EVA-02-B	✗	196^2	87.0	77.6
	✗	448^2	88.3	79.5
	40 epochs, 448^2	448^2	**88.6**	**79.8**
EVA-02-L	✗	196^2	88.9	80.7
	✗	448^2	89.6	82.3
	30 epochs, 448^2	448^2	**90.0**	**82.4**

Table 4: **More scaling can further boost the performance.** Pre-training and architectural configurations are detailed in Table 5. "IN-V2" refers to ImageNet-V2 [103].

other configurations unchanged.

The final model configuration († in Table 2), called **Transform Vision** (**TrV**, Fig. 2b), aligns with the model architecture in current leading language models [32], and achieves a favorable accuracy with an overall improvement of 1.6 points compared to the original configurations (*i.e.*, from 84.0 to 85.6), with one *caveat* that will be described next.

2.2. Pre-training Strategy

In the previous section, we choose to use features from a giant CLIP vision encoder with one billion parameters as the target representation for our MIM pretext task. However, we have not yet explained the rationale behind this choice. Although similar pre-training strategies have been well-studied in recent literature [126, 59, 45, 79, 145] and shown to be effective, they typically use vision features from much smaller CLIP models. Choosing the 1B-parameter EVA-CLIP is based on our assumption that larger CLIP will provide more robust and transferable target representations for MIM, and will ultimately lead to better pre-trained models. In Table 3, we study the impact of target representations produced by different-sized CLIPs.

A caveat from a crash course. At first glance, compared with the smaller VQKD-B [92] and CLIP-B [95] as MIM teachers, the accuracy *degenerate* (*i.e.*, from 85.0 to 84.0) with EVA-CLIP target when the students are base-sized plain ViT in [42, 5] with 300 epochs pre-training on IN-1K (∗ in Table 2 and Table 3). The architectural modifications from TrV compensate for this to some extent, resulting in a modest total improvement of 0.6-point († in Table 2 and Table 3).

We conjecture that as the teacher becomes stronger, it becomes harder for the students to learn robust and transferable representations in a crash course. Consequently, more extensive pre-training is required for the students to fully master the teacher's knowledge. As we extend the pre-training schedule to 1600 epochs (∼1M steps), TrV with EVA-CLIP as the MIM teacher yields a 1.3-point non-trivial improvement over BEiTv2 [92]. Furthermore, with 150 epochs (∼1M steps) pure MIM pre-training on ImageNet-21K (IN-21K, 14.2M images) [40], our base-sized TrV achieves 87.0 top-1 accuracy, even outperform BEiTv2 with 1600 epochs (∼1M steps) MIM pre-training on IN-1K *plus an additional 90 epochs intermediate fine-tuning on IN-21K with labels*.

Ulteriorly, in Table 4 we show that scaling model size,

model	MIM pre-training settings			macro arch configs (refer to Table 2 & Fig. 2b for micro arch)						enc.	FLOPs
	teacher	pt data	pt epochs	patch size	depth	width	attn heads	FFN type	FFN hidden dim	#params	(#tokens = 196)
EVA-02-Ti	EVA-CLIP	IN-21K (14M)	240	14×14	12	192	3	SwiGLU	512	6M	1.3G
EVA-02-S	EVA-CLIP	IN-21K (14M)	240	14×14	12	384	6	SwiGLU	1024	22M	4.6G
EVA-02-B	EVA-CLIP	IN-21K (14M)	150	14×14	12	768	12	SwiGLU	2048	86M	18G
EVA-02-L	EVA-CLIP	Merged-38M	56	14×14	24	1024	16	SwiGLU	2730	304M	62G

Table 5: **Summary of MIM pre-training settings and architecture configurations.**

resolution as well as injecting labels via intermediate fine-tuning can further boost the performance, reaching up to 90.0 top-1 accuracy on IN-1K with only a 304M-parameter **EVA-02**. Notably, our *pure* MIM pre-trained representations can achieve very competitive performance *without* additional intermediate fine-tuning.

From now on, we denote **TrV** with sufficient MIM pre-training from **EVA**-CLIP as **EVA-02**. In the rest of this section, we present some technical details of MIM pre-training before go into the performance evaluation in §3.

Model variants and architectures. We provide four variants, *i.e.*, **EVA-02**-Ti (6M), -S (22M), -B (86M) and -L (304M), as detailed in Table 5. The marco architecture (*e.g.*, model depth, width, #head) of **EVA-02** variants follows the canonical plain ViT configurations in [115, 42]. The inner-block modifications are detailed in §2.1.

Pre-training objective is similar to **EVA** [45], which is to regress the masked-out image-text aligned vision features conditioned on visible image patches only. We corrupt the input patches with [MASK] tokens, and we use block-wise masking with a masking ratio of 40% following [5, 45]. The target representation for MIM pre-training is from the publicly accessible **EVA**-CLIP [45] vision tower with one billion parameters. The output feature of **EVA-02** is first normalized [4] and then projected to the same dimension as the **EVA**-CLIP's vision feature via a linear layer. We use negative cosine similarity as the loss function.

Pre-training data. For **EVA-02**-Ti, -S and -B, we use images from IN-21K [40] for pre-training. For **EVA-02**-L, we use a merged dataset consisting of IN-21K, CC12M [23], CC3M [108], COCO [78], ADE20K [147], Object365 [107] and OpenImages [67]. For CC12M and CC3M, we only use the image data without captions. For COCO and ADE20K, we only use the training set images. The merged dataset for pre-training **EVA-02**-L has 38 million images in total (denoted as Merged-38M). All these datasets are publicly accessible.

Hyper-parameters generally follow the BEiT series [5, 92, 123]. The optimizer is Adam [64] with decoupled weight decay [84] / β_2 of 0.05 / 0.98 [80]. The peak learning rate / batch size is 3e-3 / 4k for tiny- and small-sized models, and 1.5e-3 / 2k for base- and large-sized models. We train tiny- and small-sized models for ~0.8M steps, and base- and large-sized models for ~1M steps.

Implementation. The pre-training code is based on the open-sourced **EVA** implementation [91, 45, 44]. We adopt `DeepSpeed` [102] with ZeRO stage-0 / -1 optimizer and `fp16` precision with dynamic loss scaling [98]. All MHSA operations are accelerated by `xFormers` [72]. Although our MIM teacher comes with one billion parameters, the wall-clock pre-training time is ~10% shorter than the official BEiT series implementations [5, 92].

3. Experiments and Evaluation

In this section, we present a comprehensive evaluation of our approach on representative vision tasks and benchmarks, including image classification in §3.1, contrastive image-text pre-training (CLIP) with zero-shot evaluation in §3.2, object detection & instance segmentation in §3.3.1, and semantic segmentation in §3.3.2. We conduct experiments mainly using base-sized (86M) and large-sized (304M) pre-trained representations. Our results demonstrate that **EVA-02** is capable of outperforming larger counterparts and achieving state-of-the-art performance without or with only minimal additional intermediate fine-tuning. Additional details and results can be found in the Appendix.

3.1. Image Classification

Datasets. For image classification, we mainly evaluate **EVA-02** on IN-1K [105]. We also evaluate the robustness & generalization capability of **EVA-02** along with our training settings & hyper-parameters using some IN-1K validation set variants, including ImageNet-V2 matched frequency (IN-V2) [104], ImageNet-ReaL (IN-ReaL) [8], ImageNet-Adversarial (IN-Adv.) [57], ImageNet-Rendition (IN-Ren.) [55], ImageNet-Sketch (IN-Ske.) [121], as well as ObjectNet (ObjNet) [6], following the settings in [51, 45].

Training settings. To fully unleash the potential of **EVA-02**, we optionally perform intermediate fine-tuning following [5, 92] for base- / large-sized model on IN-21K [40] for 40 / 30 epochs in Table 7. The final IN-1K fine-tuning for all-sized models (including **EVA-02**-Ti and -S) can be done without using strong regularization such as cutmix [141], mixup [143] and random erasing [146]. In the Appendix, we show that our pre-trained representations are robust enough that can be fine-tuned using various numerical precisions (*e.g.*, `fp16` and `bf16`) and optimizers (*e.g.*, Lion [26], AdamW [64, 84], and SGD [87]). Remarkably, the fine-tuning can be done even using the SGD optimizer with only 0.1-point performance drop.

method	#params	extra labeled data	crop size	IN-1K top-1
(a) comparisons with SOTA **base**-sized models (86M)				
LAION-ViT-CLIP-B† [68]	86M	LAION-2B & IN-21K	384^2	87.2
BEiTv2-B [92]	86M	IN-21K (14M)	384^2	87.5
ViT-B ⚗ ViT-22B-JFT-4B [39]	86M	JFT-4B	384^2	88.6
EVA-02-B	86M	IN-21K (14M)	448^2	**88.6**
(b) comparisons with **larger** SOTA models				
LAION-ViT-CLIP-L† [70]	304M	LAION-2B & IN-21K	336^2	88.2
FD-CLIP-L [127]	304M	IN-21K (14M)	336^2	89.0
BEiTv2-L [92]	304M	IN-21K (14M)	384^2	89.2
ViT-L ⚗ ViT-22B-JFT-4B [39]	304M	JFT-4B	384^2	89.6
EVA-02-L	304M	IN-21K (14M)	448^2	**90.0**
InternImage-H [124]	~1080M	427M img-txt & IN-21K	640^2	89.2
EVA-CLIP† [45]	1011M	IN-21K (14M)	336^2	89.5
BEiT-3 [123]	~1900M	400M img-txt & IN-21K	336^2	89.6
EVA [45]	1011M	IN-21K (14M)	560^2	89.7
RevCol-H [18]	2158M	168M (semi sup.)	640^2	90.0

Table 7: **EVA-02-B and EVA-02-L image classification performance on IN-1K val set.** Using only publicly accessible data, EVA-02 creates phenomenal results with affordable model size.
"†": fine-tuned CLIP vision encoder. "⚗": model distillation [58, 9].

IN-1K results (EVA-02-B & -L). Table 7 compares EVA-02 with some state-of-the-art models on IN-1K val set. Our base-sized model, trained with ImageNet data only, outperforms several strong competitors and achieves the same performance with a ViT-B distilled from a 4B-parameter teacher using large-scale in-house training data [39]. Furthermore, EVA-02-L with only 304M-parameter can achieve a phenomenal 90.0 fine-tuning top-1 accuracy, outperforms several state-of-the-art larger models trained with more (often publicly inaccessible) data, *including its fine-tuned EVA-CLIP MIM teacher*, which distinguishes MIM from knowledge distillation [58].

IN-1K results (EVA-02-Ti & -S). It is commonly believed that plain ViTs perform mediocrely due to the lack of inductive biases in light-weight settings. However, compared with specialized light-weight networks with strong visual

method	#params	IN-1K ft img size	FLOPs	IN-21K label?	IN-1K top-1
(a) model size: 5M~10M					
MobileViTv3-1.0 [120]	5.1M	384^2	4.2G	✗	79.7
MobileViTv2-1.5 [86]	10.6M	256^2	4.0G	✗	80.4
EVA-02-Ti	5.7M	336^2	4.8G	✗	**80.7**
(b) model size: 20M~30M					
DeiT-III-S [116]	22M	384^2	16G	✓	84.8
ConvNeXt V2-T [129]	29M	384^2	13G	✓	85.1
MOAT-0 [135]	28M	384^2	18G	✓	85.7
EVA-02-S	22M	336^2	16G	✗	**85.8**
BEiTv2-B [92]	86M	224^2	18G	✗	85.5

Table 8: **EVA-02-Ti and EVA-02-S image classification performance on IN-1K val set.** EVA-02 with fewer inductive biases but sufficient MIM pre-training is performant in light-weight settings.

structure prior in Table 8, EVA-02 as a plain ViT variant equipped with extensive MIM pre-training can trump inductive biases, and achieve favorable performance with tiny and small models.

Robustness evaluation. We evaluate the robustness and generalization capability of EVA-02 on several IN-1K val set variants. Following the evaluation procedure in [51, 45], all these models are first fine-tuned on the original IN-1K training set, and then directly evaluated on different val sets using the *same* fine-tuned model *without further hyper-parameter selection and specialized fine-tuning*.

In Table 6, we compare EVA-02 with some top open-sourced models. EVA-02 is the most competitive one in terms of top-1 accuracies. Besides the absolute performance, we also care about whether a model along with its training settings biases towards the original validation set and generalizes well on others. From this perspective, EVA-02 not only achieves the highest averaged accuracy, but also has the smallest performance gap (as measured by the difference between the averaged accuracy of val set variants and the original IN-1K val set accuracy), which reflects the excellent robustness and generalization ability of EVA-02.

method	#params	data	IN-1K [105]	IN-V2 [104]	IN-ReaL [8]	IN-Adv. [57]	IN-Ren. [55]	IN-Ske. [121]	ObjNet [6]	avg.	Δ↓
(a) comparisons with SOTA **base**-sized models (86M)											
LAION-ViT-CLIP-B† [68]	86M	LAION-2B & IN-21K	87.2	77.8	90.2	59.2	66.2	53.5	-	72.4	14.8
DeiT-III-H [116]	632M	IN-21K	87.2	79.2	90.2	70.2	70.8	55.8	-	75.6	11.6
EVA-02-B	**86M**	IN-21K	**88.6**	**79.8**	**90.8**	**78.1**	**76.8**	**57.7**	55.3	**78.6**	**10.0**
(b) comparisons with **larger** SOTA models											
LAION-ViT-CLIP-H† [69]	632M	LAION-2B & IN-21K	88.6	79.5	90.5	74.2	83.1	65.3	-	80.2	8.4
EVA [45] (prev. best)	1011M	Merged-30M	89.6	81.6	90.8	86.2	88.3	67.7	60.9	84.0	5.6
EVA-02-L	**304M**	Merged-38M	**90.0**	**82.4**	**91.1**	**87.7**	**89.9**	**70.1**	62.8	**85.2**	**4.8**

Table 6: **Robustness & generalization capability evaluation on IN-1K variants.** All these models are first fine-tuned on the original IN-1K training set and then evaluated on different val sets using the *same* fine-tuned model *without any specialized fine-tuning*. "avg.": the averaged top-1 accuracy on different IN-1K val set variants (*i.e.*, IN-{1K, V2, ReaL, Adv., Ren., Ske.}, excluding ObjNet). "Δ↓": The gap between the averaged top-1 accuracy of val set variants and the original IN-1K validation set top-1 accuracy (the lower the better).
"†": fine-tuned CLIP vision encoder

method	#params (img+text)	precision	dataset & samples	img size	IN-1K zs top-1
(a) comparisons with CLIP-**Base** baselines					
OpenAI CLIP-B/16	86M+63M	`fp16`	WIT-400M & 13B	224^2	68.3
OpenCLIP-B/16	86M+63M	`bf16`	LAION-2B & 34B	224^2	70.2
EVA-02-CLIP-B/16	86M+63M	`fp16`	Merged-2B & **8B**	224^2	**74.7**
(b) comparisons with CLIP-**Large** baselines					
OpenAI CLIP-L/14	0.3B+124M	`fp16`	WIT-400M & 13B	224^2	75.5
OpenCLIP-L/14	0.3B+124M	`bf16`	LAION-2B & 32B	224^2	75.3
EVA-02-CLIP-L/14	0.3B+124M	`fp16`	Merged-2B & **4B**	224^2	**79.8**
(c) comparisons with **larger** CLIPs trained with **more samples**					
OpenAI CLIP-L/14+	0.3B+124M	`fp16`	WIT-400M & 13B	336^2	76.6
OpenCLIP-H/14	0.6B+354M	`bf16`	LAION-2B & 32B	224^2	78.0
FLIP-H/14	0.6B+354M	`fp32`	LAION-2B & 26B	224^2	78.1
EVA-CLIP-g/14	1.0B+124M	`fp16`	LAION-0.4B & 11B	224^2	78.5
⚗ OpenCLIP-G/14	1.8B+695M	`bf16`	LAION-2B & 39B	224^2	80.1
EVA-02-CLIP-L/14+	**0.3B+124M**	`fp16`	Merged-2B & **6B**	336^2	**80.4**

Table 10: **CLIP configurations & IN-1K zero-shot performance.** EVA-02-CLIP achieves better performance with affordable size and fewer image-text samples.
"+": initialized from CLIP checkpoint trained with 224^2 following [95]
"⚗": model soups [130]

method	#params (img+text)	UCF-101	K-400	K-600	K-700	**avg. acc.**
(a) comparisons with CLIP-**Base** baselines						
OpenAI CLIP-B/16	86M+63M	67.1	**57.6**	56.5	49.3	57.6
EVA-02-CLIP-B/16	86M+63M	**68.6**	57.4	**57.0**	**50.0**	**58.3**
(b) comparisons with **larger**-sized CLIP models						
OpenAI CLIP-L/14+	0.3B+124M	78.1	64.9	65.0	58.5	66.6
OpenCLIP-H/14	0.6B+354M	78.2	63.1	63.6	56.1	65.3
EVA-02-CLIP-L/14+	0.3B+124M	**78.6**	**65.9**	**66.1**	**60.2**	**67.7**

Table 11: **Zero-shot video classification performance.** Following [95], we report top-1 accuracy for UCF-101 [111], and the mean of top-1 and top-5 accuracies for K-400 [22], K-600 [20] and K-700 [21] datasets.

3.2. Contrastive Language-Image Pre-training and Zero-shot Evaluation

Contrastive Language-Image Pre-trained (CLIP) model is a kind of foundation model that aligns vision and natural language through contrastive image-text pre-training [95]. Its impact on the field of representation learning has been significant, making it a powerful engine for both recognition and generation tasks, as well as uni-modal and multi-modal applications [100, 45, 73, 106].

In this section, we thoroughly demonstrate the efficacy of initializing **EVA-02** as the CLIP vision encoder following the settings in [45]. The resulting model, referred to as **EVA-02**-CLIP, significantly improves zero-shot performance, sample efficiency, and training speed.

CLIP configurations & zero-shot classification. We present CLIP model configurations and IN-1K zero-shot accuracies in Table 10. To train **EVA-02**-CLIP, we merge the data from the publicly accessible LAION-2B [106] and COYO-700M [16], which results in a dataset of 2 billion image-text pairs (we only have ∼1.6B / ∼400M valid samples from LAION-2B / COYO-700M datasets). Leveraging MIM pre-trained **EVA-02** representations, our CLIP model significantly outperforms previous approaches in IN-1K zero-shot classification, achieving an outstanding 74.7 / 80.4 top-1 accuracy with base- / large-sized models.

In Table 9, we further demonstrate the efficacy and robustness of our approach on 26 additional zero-shot classification benchmarks. Notably, our **EVA-02**-CLIP-L model, which only has ∼1/2 of the model size and ∼1/5 image-text pairs, achieves a 1.2-point non-trival averaged improvement over OpenCLIP-H.

Finally, in Table 11 we show that **EVA-02**-CLIP is also quite effective in zero-shot video recognition benchmarks.

Zero-shot retrieval performance. Table 12 comprehensively reports the zero-shot image and text retrieval results on Flickr30K [138] and COCO [78]. **EVA-02**-CLIP outperforms all the competitors with the same model size. While the zero-shot retrieval performance of **EVA-02**-CLIP is not as significant as classification compared to OpenCLIP-H,

	ImageNet-1K [105]	ImageNet-V2 [104]	ImageNet-Adv. [57]	ImageNet-Ren. [55]	ImageNet-Ske. [121]	ObjectNet [6]	CIFAR-10 [66]	CIFAR-100 [66]	MNIST [71]	Caltech-101 [47]	SUN397 [131]	FGVC Aircraft [85]	Country-211 [95]	Stanford Cars [65]	Birdsnap [7]	DTD [33]	Eurosat [54]	FER2013 [49]	Flowers-102 [88]	Food-101 [13]	GTSRB [112]	PCam [119]	Pets [90]	Rendered SST2 [95]	Resisc45 [30]	STL10 [35]	VOC2017 [43]	**avg. top-1 acc.**
(a) comparisons with CLIP-**Base** baselines																												
OpenAI CLIP-B/16	68.3	61.9	50.0	77.7	48.2	55.3	90.8	67.0	**51.6**	84.7	64.4	24.4	**22.8**	64.8	34.5	44.7	55.0	46.2	71.3	88.8	43.5	50.7	89.1	**60.8**	59.1	98.3	78.3	61.2
EVA-02-CLIP-B/16	**74.7**	**67.0**	**54.1**	**82.5**	**57.7**	**62.3**	**98.4**	**87.7**	47.9	**86.3**	**70.7**	**24.8**	21.4	**78.6**	**37.7**	**53.1**	**67.0**	**51.2**	**75.9**	**89.4**	**46.3**	**50.9**	**92.2**	54.1	**60.7**	**99.5**	**80.2**	**65.6**
(b) comparisons with **larger** CLIP models																												
OpenAI CLIP-L/14+	76.6	70.9	77.5	89.0	61.0	72.0	94.9	74.4	**79.0**	87.2	68.7	33.4	**34.5**	79.3	41.0	56.0	61.5	49.1	78.6	93.9	52.4	**60.8**	93.8	**70.7**	65.4	99.4	78.1	70.3
OpenCLIP-H/14	78.0	70.8	59.2	89.3	66.6	69.7	97.4	84.7	72.9	85.0	**75.2**	**42.8**	30.0	**93.5**	**52.9**	**67.8**	**72.7**	**52.0**	**80.1**	92.7	**58.4**	54.2	**94.5**	64.3	**70.5**	98.5	77.7	72.3
EVA-02-CLIP-L/14+	**80.4**	**73.8**	**82.9**	**93.2**	**68.9**	**78.4**	**98.9**	**89.8**	64.3	**89.5**	74.8	37.5	33.6	91.6	45.8	64.5	71.4	51.0	77.2	**94.2**	57.6	54.9	94.2	64.6	69.8	**99.7**	**82.7**	**73.5**

Table 9: **Summary of EVA-02-CLIP zero-shot image classification performance on 27 datasets.**

method	#params (img + text)	dataset	img-text samples	zero-shot **text** retrieval Flickr30K R@1	R@5	R@10	COCO R@1	R@5	R@10	zero-shot **image** retrieval Flickr30K R@1	R@5	R@10	COCO R@1	R@5	R@10
(a) comparisons with CLIP-**Base** baselines															
OpenAI CLIP-B/16	86M + 63M	WIT-400M	13B	81.9	96.2	98.8	52.4	76.8	84.7	62.1	85.6	91.8	33.1	58.4	69.0
EVA-02-CLIP-B/16	86M + 63M	Merged-2B	8B	**85.7**	**96.7**	**98.9**	**58.7**	**80.7**	**88.2**	**71.2**	**91.0**	**94.7**	**42.2**	**66.9**	**76.3**
OpenAI CLIP-L/14	304M + 124M	WIT-400M	13B	85.2	97.3	99.0	56.3	79.3	86.7	65.2	87.3	92.0	36.5	61.0	71.1
(b) comparisons with **larger** CLIP models															
OpenAI CLIP-L/14	0.3B + 124M	WIT-400M	13B	85.2	97.3	99.0	56.3	79.3	86.7	65.2	87.3	92.0	36.5	61.0	71.1
OpenCLIP-L/14	0.3B + 124M	LAION-2B	32B	88.7	98.4	99.2	62.1	83.4	90.3	75.0	92.5	95.6	46.1	70.7	79.4
EVA-02-CLIP-L/14	0.3B + 124M	Merged-2B	4B	**89.7**	98.6	99.2	63.7	84.3	90.4	77.3	93.6	**96.8**	47.5	71.2	79.7
OpenAI CLIP-L/14+	0.3B + 124M	WIT-400M	13B	87.4	98.3	99.3	57.9	81.2	87.9	67.3	89.0	93.3	37.1	61.6	71.5
EVA-02-CLIP-L/14+	0.3B + 124M	Merged-2B	6B	89.2	**98.9**	**99.6**	**64.1**	**85.2**	**90.8**	**77.9**	**94.2**	**96.8**	**47.9**	**71.7**	**80.0**
OpenCLIP-H/14	0.6B + 354M	LAION-2B	32B	90.8	99.3	99.7	66.0	86.1	91.9	77.8	94.1	96.6	49.5	73.4	81.5

Table 12: **EVA-02-CLIP zero-shot retrieval performance.**

method	enc. #params	COCO val AP^{box}	COCO val AP^{mask}	LVIS val AP^{box}	LVIS val AP^{mask}
ViTDet-B [75]	86M	54.0	46.7	43.0	38.9
EVA-02-B	86M	**55.5**	**47.1**	**47.1**	**41.4**

(a) **Head-to-head comparisons** with the open-sourced ViTDet config.

method	enc. #params	COCO val AP^{box}	COCO val AP^{mask}
ViTDet-B [75]	86M	56.0	48.0
MViTv2-L [76]	218M	56.9	48.6
MViTv2-H [76]	667M	57.1	48.8
EVA-02-B	**86M**	**58.9**	**50.7**

(b) **System comparisons** *without* additional detection training data.

Table 13: **Object detection and instance segmentation results of EVA-02-B.**

the results are still competitive. We speculate that the main reason for this difference is that retrieval tasks depend more on the capacity and capability of the language encoder compared to classification tasks.

3.3. Object Detection and Segmentation

In this section, we evaluate the transfer learning performance of **EVA-02** to mainstream object-level and pixel-level recognition benchmarks, namely, object detection and instance segmentation on COCO [78] and LVIS [50] in §3.3.1, as well as semantic segmentation on COCO-Stuff-164K [17] and ADE20K [147] in §3.3.2.

3.3.1 Object Detection and Instance Segmentation

For thoroughly evaluating the performance of **EVA-02** on object detection and instance segmentation tasks, we adopt the canonical Cascade Mask R-CNN [52, 19] as the task layer. This choice is motivated by its versatility in simultaneously performing both tasks, as well as its robustness and accuracy. To ensure fair comparisons with existing state-of-the-art methods, we essentially follow the training settings and architecture configurations of ViTDet [75], which includes large-scale jittering (LSJ) data augmentation [48] and interleaved windowed and global attention mechanisms.

The model architecture as well as the hyper-parameters for COCO and LVIS are almost the same, except we use federated loss [149] and repeat factor sampling [50] following ViTDet on LVIS. For LVIS, we use the IN-21K MIM pre-trained checkpoints of **EVA-02** for all experiments, as the COCO training images in the Merged-38M dataset include 10k images in LVIS `val` set[2].

In the rest of this section, we evaluate **EVA-02** under three different transfer learning settings in Table 13 and Table 14, including (i) a sanity check, (ii) a system-level comparison *without* using additional detection data, and (iii) a system-level comparison *with* additional intermediate detection fine-tuning.

(i) A sanity check. We first use the same open-sourced architectural configurations as ViTDet (LSJ with 1024^2 crops, 4×global attention blocks) to perform a head-to-head comparison. In general, both **EVA-02**-B in Table 13a and **EVA-02**-L in Table 14a can outperform the same- / larger-sized ViTDet *w/* Cascade Mask R-CNN counterparts by a large margin, especially on LVIS.

(ii) System comparisons *w/o* additional detection data. In Table 13b and Table 14b, we explore the limits of *pure* MIM pre-trained **EVA-02**-B and -L representations in object detection and instance segmentation tasks. To fully unleash the potential of **EVA-02**, we use an improved ViTDet configuration (LSJ with 1536^2 crops, windowed attention with a size of 32, and 6× / 8×global attention blocks for base- / large-sized models). Soft-NMS [12] is also applied. For instance segmentation task, the classification score is calibrated [62] via maskness [125]. The baselines we compared also adopt improved settings such as larger input resolution, Soft-NMS, *etc*., and RevCol

[2] In the Appendix, we show including unlabeled images from development / test set for MIM pre-training **does not** improve the final performance.

method	enc. #params	COCO val AP^{box}	COCO val AP^{mask}	LVIS val AP^{box}	LVIS val AP^{mask}
ViTDet-L [75]	304M	57.6	50.0	49.2	44.5
ViTDet-H [75]	632M	58.7	**51.0**	51.5	46.6
EVA-02-L	**304M**	**59.2**	50.8	**55.3**	**48.6**

(a) **Head-to-head comparisons** with the open-sourced ViTDet config.

method	enc. #params	COCO val AP^{box}	COCO val AP^{mask}	LVIS val AP^{box}	LVIS val AP^{mask}
ViTDet-L [75]	304M	59.6	51.1	51.2	46.0
ViTDet-H [75]	632M	60.4	52.0	53.4	48.1
RevCol-H [18]	2158M	61.1	53.0	-	-
EVA-02-L	**304M**	**62.3**	**53.8**	**60.1**	**53.5**

(b) **System comparisons** *without* additional detection training data.

method	enc. #params	COCO val AP^{box}	COCO val AP^{mask}	COCO test-dev AP^{box}	COCO test-dev AP^{mask}
BEiT-3 [123]	1011M	-	-	63.7^{tta}	54.8^{tta}
FocalNet-H [136]	689M	63.8	-	63.9	-
FD-SwinV2-G [127]	~3000M	-	-	64.2^{tta}	55.4^{tta}
InternImg-XL†† [124]	~600M	64.2^{tta}	-	64.3^{tta}	-
GDETRv2 [25]	632M	-	-	$\mathbf{64.5}^{tta}$	-
EVA [45]	1011M	**64.2**	55.0	64.4	55.5
EVA-02-L	**304M**	64.1	**55.4**	**64.5**	**55.8**
InternImg-H†† [124]	~2000M	65.0^{tta}	-	65.4^{tta}	-

(c) **System comparisons** on COCO *with* additional training on O365.

method	enc. #params	LVIS val AP^{box}	LVIS val AP^{mask}
EVA [45]	1011M	62.2	55.0
InternImg-H†† [124]	~2000M	63.2^{tta}	-
EVA-02-L	**304M**	**65.2**	**57.3**

(d) **System comparisons** on LVIS *with* additional training on O365.

Table 14: **Object detection and instance segmentation results of EVA-02-L.**

"††": encoder parameters doubled using model composite technique [77]

initializes HTC++ [24, 82] as the task layer, which is an improved version of Cascade Mask R-CNN we used.

Our experiments demonstrate that **EVA-02** significantly outperforms the same- and larger-sized counterparts, particularly on LVIS. These findings are consistent with our previous results in Table 13a and Table 14a. We also encourage future work in representation learning to conduct more in-depth investigations on the *original* pre-trained representations before adding more intermediate processes to chase the absolute performance.

(iii) System comparisons *w/* additional O365 training. For the state-of-the-art detection system comparisons in Table 14c and Table 14d, all methods use Object365 (O365) [107] detection annotations for further performance improvements. We additionally use EMA [94] to update model weights. All results of **EVA-02** use single-scale evaluation while methods leveraging test-time augmentations are marked with "tta" superscript. Methods that sacrifice instance segmentation ability in Table 14c use the better-established DINO [144] as the detector. Compared with other state-of-the-art approaches with much larger model sizes, our **EVA-02** is still quite competitive, especially on LVIS.

method	enc. #params	crop size	extra labeled data	ADE20K mIoU
(a) comparisons with **based**-sized encoders				
BEiTv2-B [92]	86M	512^2	✗	53.1
BEiTv2-B [92]	86M	512^2	IN-21K	53.5
EVA-02-B	**86M**	512^2	✗	**55.3**
DeiT-III-L [116]	304M	512^2	IN-21K	54.6
InternImage-XL [124]	335M	640^2	IN-21K	55.0
ConvNeXt V2-H [129]	707M	512^2	✗	55.0
(b) comparisons with **larger**-sized encoders				
BEiTv2-L [92]	304M	512^2	✗	56.7
BEiTv2-L [92]	304M	512^2	IN-21K	57.5
EVA-02-L	**304M**	512^2	✗	**59.8**
EVA-02-L+	**304M**	640^2	✗	**60.1**
ConvNeXt V2-H [129]	707M	640^2	IN-21K	57.0
RevCol-H [18]	2158M	640^2	168M	57.8
SwinV2-G [81]	~3000M	896^2	70M	59.3
InternImage-H [124]	1080M	896^2	IN-21K	59.9

Table 15: **Semantic segmentation with UperNet.** All methods use single-scale evaluation.

"+": using larger input resolution and segmentation head dimension

method	enc. #params	crop size	COCO164K $mIoU^{ss}$	ADE20K $mIoU^{ss}$	ADE20K $mIoU^{ms}$
RevCol-H	2158M	640^2	-	60.4	61.0
BEiTv2-L w/ ViT-Ada.	304M	896^2	52.3	61.2	61.5
EVA w/ ViT-Ada.	1011M	896^2	53.4	61.5	**62.3**
EVA-02-L	**304M**	640^2	**53.7**	**61.7**	62.0

Table 16: **Semantic segmentation with Mask2Former.** "$mIoU^{ss}$/ $mIoU^{ms}$": mIoU using single-scale / multi-scale evaluation.

3.3.2 Semantic Segmentation

We comprehensively evaluate the semantic segmentation performance of **EVA-02**-B and -L models using two different task layers: UperNet [132] and Mask2Former [29] on two widely adopted benchmarks: ADE20K [147] and COCO-Stuff-164K [17]. Notably, unlike previous mainstream approaches that involve additional fine-tuning such as using IN-21K intermediate fine-tuned models for semantic segmentation, we primarily evaluate *pure* MIM pre-trained representations of **EVA-02**.

UperNet results. As shown in Table 15, both pure MIM pre-trained **EVA-02**-B and -L models with UperNet segmenter significantly outperform the same-sized BEiTv2 models without or *with* the additional 90-epoch IN-21K intermediate fine-tuning. Furthermore, our representation can outperform larger pre-trained counterparts such as ConvNeXt V2, InternImage, *etc*., and achieves up to 60.1 mIoU with single-scale evaluation.

Mask2Former results. Table 16 shows the state-of-the-art model comparisons on COCO-Stuff-164K and ADE20K benchmarks. Models for ADE20K segmentation are initialized from COCO-Stuff-164K pre-trained representations as long as the COCO-Stuff-164K results are reported. BEiTv2-L and EVA also utilize ViT-Adapter (ViT-Ada. in Table 16) [28] for architectural improvements.

Compared with larger models using the Mask2Former task layer, our approach is still quite performant, and creates new state-of-the-art results with large-sized models on both COCO-Stuff-164K and ADE20K semantic segmentation benchmarks.

3.4. Summary of All Evaluations

In §3, we demonstrate the excellent transfer learning ability of pre-trained **EVA-02** representations on a large diversity of downstream tasks. Although all tasks / benchmarks we evaluated are at the core of computer vision, here we would like to (re-)emphasize the importance of the ones related to **EVA-02**-CLIP: not only for the promising zero-shot transferability, but also because the vision features from **EVA-02**-CLIP are *well aligned* with natural language that comes with much broader supervision than pure vision signals / features as well as fixed set of pre-determined label sets. Therefore, we hope **EVA-02**-CLIP can serve as a basic building blocks and provide more robust vision features for future multi-modal systems.

4. Related Work

Some previous advancements in representation learning do not necessarily come with entirely new ideas or novel approaches. The GPT series [96, 97, 15, 89] achieve quantitative changes that transform the landscape of scientific research by continuously scaling the simplest language modeling. RoBERTa [80] present a detailed replication study of BERT pre-training [41] that carefully measures the impact of many key hyper-parameters, training data and objectives, which results in greatly improved bidirectional language representations. DeiT [115] and RSB [128] closely evaluate the training recipe for smaller-sized plain ViTs [42] and ResNets [53] respectively, while ConvNeXt [83] collectively examines previous architectural advancements for the next-gen ConvNets model design. [9] empirically shows that a robust and effective recipe of knowledge distillation makes state-of-the-art large-scale image classification models affordable in practice.

Inspired by the spirits of these works, this paper provides a thorough evaluation of MIM visual representation learning [5, 148, 133, 51] that significantly bridge the gap between large-scale visual representations that achieve state-of-the-art performance and models that are affordable and accessible for the wider research community.

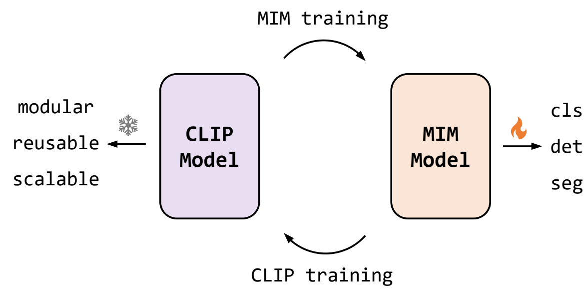


Figure 3: **Alternate learning of MIM and CLIP representations.** Starting with a off-the-shelf CLIP(*e.g.*, OpenAI CLIP [95]), alternate training of the pure MIM visual representations as well as vision-language CLIP representations can improve both MIM and CLIP performances in a bootstrapped manner. The MIM representations can be used to fine-tune various downstream tasks while the (frozen) CLIP representations enable modular, reusable and scalable next-gen model design.

5. Discussion and Conclusion

In this work, we aim to contribute to the ongoing research on visual and vision-language representation learning. Instead of proposing an entirely new architecture or method, we present an in-depth evaluation of the existing MIM pre-training with CLIP vision features as the pretext task's targets. Our experiments demonstrated that if robustly optimized, this approach is capable of producing highly performant, affordable, and transferable representations that outperform larger state-of-the-art specialized models.

Our analysis has revealed that base- & large-sized **EVA-02** models can be effectively leveraged to obtain compact and expressive CLIP representations, which have the potential to facilitate modular, reusable, and scalable model design in the future [100, 3, 27, 73]. Our findings on moderate-sized models can also serve as a valuable reference for future research on model and representation scaling.

Furthermore, in combination with **EVA** [45], we demonstrate that alternate training of the pure MIM visual representations as well as vision-language CLIP representations can improve both MIM and CLIP performances in a bootstrapped manner (Fig. 3). This suggests a promising and scalable approach for pre-training both vision and vision-language representations of various sizes, which warrants further exploration in future research.

Acknowledgement

We would like to thank Hanxiao Qu, Yan Tian, Yemin Shi and Xigang Cao for their help on GPU resources. Zhao Xue, Quanyue Ma and Bowen Zhang for their help on datasets and benchmarks, and other colleagues at Beijing Academy of Artificial Intelligence for support throughout this project. We also thank Wen Wang for constructive discussions and assistance on object detection & instance segmentation tasks.

method	MIM teacher	IN-1K ft img size	FLOPs	IN-21K label?	IN-1K top-1
(a) ViT-Base model (86M), IN-1K ft number of tokens = 196					
BEiTv2-B [92]	VQKD-B	224^2	18G	✗	85.5
dBOT-B [79]	CLIP-B	224^2	18G	✗	85.7
BEiTv2-B [92]	VQKD-B	224^2	18G	✓	86.5
EVA-02-B	EVA-CLIP	196^2	18G	✗	**87.0**
(b) ViT-Large model (304M), IN-1K ft number of tokens = 196					
BEiTv2-L [92]	VQKD-B	224^2	62G	✗	87.3
dBOT-L [79]	CLIP-L	224^2	62G	✗	87.8
BEiTv2-L [92]	VQKD-B	224^2	62G	✓	88.4
EVA-02-L	EVA-CLIP	196^2	62G	✗	**88.9**

Table 17: **Head-to-head comparisons of based- and large-sized models on IN-1K val set classification.** The fine-tuning settings are relatively moderate with the same compute budget for each model.

method	optimizer	IN-1K top-1 fp16	IN-1K top-1 bf16	IN-V2 top-1 fp16	IN-V2 top-1 bf16
EVA-02-B	SGD	88.40	88.37	79.73	79.67
	AdamW	88.57	**88.58**	79.78	79.74
	Lion	88.52	88.50	**79.97**	79.96
EVA-02-L	SGD	89.87	89.84	82.15	82.17
	AdamW	89.98	89.95	82.43	**82.61**
	Lion	89.97	**90.00**	82.19	82.37

Table 18: **Study of different numerical precisions and optimizers on IN-1K classification fine-tuning.** To explore the limit of EVA-02 representation, all pre-trained models are fine-tuned at a resolution of 448^2 with IN-21K intermediate fine-tuning following the most performant settings in Table 4.

method	role	enc. #params	best IN-1K top-1 *w/o* IN-21K ft	best IN-1K top-1 *w/* IN-21K ft
EVA-CLIP†	teacher	1011M	89.4	89.5
EVA-02-L	student	304M	89.6	90.0

Table 19: **Indigo blue comes from indigo.** With sufficient pre-training, EVA-02-L with 304M-parameter is able to surpass its teacher with 1011M-parameter in IN-1K image classification.
"†": fine-tuned CLIP vision encoder

A. Appendix

A.1. Architecture

SwiGLU FFN. The position-wise feedforward network (FFN) in the original ViT design [42] is a multi-layer perceptron (MLP) contains two layers (represented by the weight matrices W_1 and W_2, biases are omitted) with a GELU [56] activation function, denoted as $\mathrm{FFN}_{\mathrm{MLP}}$. Formally,

$$\mathrm{FFN}_{\mathrm{MLP}}\left(x, W_1, W_2\right) = \mathrm{GELU}\left(xW_1\right) W_2. \tag{1}$$

SwiGLU FFN [110] replace the first transformation of the original ViT's FFN with a variant of the Gated Linear Unit (GLU) [38] with a SiLU ($\mathrm{SiLU} = x * \mathrm{sigmoid}(x)$) activation function [56, 99], Formally,

$$\mathrm{FFN}_{\mathrm{SwiGLU}}\left(x, U, V, W\right) = \left(\mathrm{SiLU}(xU) \odot xV\right) W, \tag{2}$$

where $\odot$ is the element-wise product.

To keep the number of parameters and the amount of computation constant, we reduce the hidden units (the output dimension of U and V and the input dimension of W) of $\mathrm{FFN}_{\mathrm{SwiGLU}}$ by a factor of 2/3 when comparing these layers to the original $\mathrm{FFN}_{\mathrm{MLP}}$.

Normalization. We use sub-LN [122] (We find the inner attention LN unnecessary so we drop it) as the default normalization scheme for EVA-02-B and -L blocks. For the tiny- and small-sized model, we find using the default pre-LN configuration following [42, 5] is sufficient.

RoPE is a type of position embedding that unifies absolute as well as relative potential representations, and is widely adopted in state-of-the-art language models [11, 32, 27]. For a detailed description of RoPE, please refer to [113, 10]. Our implementation is based on the open-sourced [2].

In brief, RoPE twists / rotates the input embedding (without changing the norm) such that the attention of a token at position m to a token at position n is linearly dependent on $m - n$. Notably, unlike the conventional relative position representations that inject the positional information into the attention matrix, RoPE only manipulates q, k vectors. So RoPE is naturally compatible with off-the-shelf fused high-performance MHSA operators such as [37, 72].

A.2. Additional Results for Image Classification

EVA-02-B and -L. In Table 17, we show that sufficiently pre-trained pure MIM EVA-02 representations (*w/o* IN-21K intermediate fine-tuning) outperform some previous leading approaches (even *w/* intermediate fine-tuning).

Precisions and optimizers. In Table 18, we show that sufficiently pre-trained EVA-02 representations are robust enough that can be fine-tuned using various numerical precisions (*e.g.*, `fp16` and `bf16`) and optimizers (*e.g.*, Lion [26], AdamW [64, 84], and SGD [87]). Remarkably, the fine-tuning can be done using the SGD optimizer with only little performance drop.

The student is the master. Table 19 distinguishes MIM from conventional knowledge distillation [58] in the context of "pre-training & fine-tuning" paradigm.

A.3. Data Contamination in MIM Pre-training: A Case Study

We provide a case study about the impact of data contamination in MIM pre-training when transferred to object detection and instance segmentation tasks. In short, we find the impact is minor.

MIM pt data	MIM to O365 AP^box	MIM to LVIS (Table 14a) AP^box	AP^mask	MIM to O365 to LVIS (Table 14d) AP^box	AP^mask
Merged-38M	50.57	55.34	48.74	65.42	57.42
IN-21K	50.47	55.28	48.59	65.22	57.32

Table 20: **The impact of data contamination in MIM pre-training** when transferred to object detection & instance segmentation tasks. The setting in pink is the default setting we used in Table 14 for LVIS `val` set evaluation.

	based-sized model (86M) ViT	TrV	larged-sized model (304M) ViT	TrV
throughput (img / s)	1600	2226	554	636

Table 21: **Inference throughput comparisons using one A100 GPU.** The batch size is 1024. The number of patch tokens is 196. The architecture of ViT follows BEiT series [5, 92] (with rel. PE [109] and LayerScale [117]).

We pre-train two **EVA-02**-L models, one uses the Merged-38M unlabeled images for MIM pre-training, and the other uses the images from IN-21K as the pre-training data. Both models are pre-trained with 1M steps with a batch size of 2k. Other settings & configurations are the same. Notice that the Merged-38M unlabeled images contain all Object365 (O365) [107] `test` set images, as well as 15k out of 20k LVIS [50] `val` set images (the Merged-38M images contain all the COCO training images, and LVISv1.0 `val` split also contains 15k images from the COCO training set).

We study the transfer learning performance in three different settings:

(i) Directly transfer pure MIM pre-trained **EVA-02** representations to O365 (MIM to O365), The performance is evaluated using the O365 `test` set[3](the `test` set is a very large and challenging benchmark with ~200k images and ~2.5M instances in 365 different categories).

(ii) Directly transfer pure MIM pre-trained **EVA-02** representations to LVIS (MIM to LVIS, Table 14a). The performance is evaluated using LVIS `val` set (the `val` set is a long-tail, large-vocabulary challenging benchmark with ~20k images and ~0.25M federated annotated instances in more than 1.2k different categories).

(iii) Transfer the **EVA-02** representations with additional O365 intermediate fine-tuning to LVIS (MIM to O365 to LVIS, Table 14d). The performance is evaluated using LVIS `val` set.

The results are summarized in Table 20. Overall, we find including unlabeled images from the development / test set for MIM pre-training has little impact on the final performance.

These experiments are motivated by our initial use of the Merged-38M pre-trained representation for LVIS `val` set evaluation, which resulted in unintended use of unlabeled images from the development / test set for MIM pre-training, similar to the issue raised in [60]. [31] also reports a small percentage of images from IN-1K along with its variants, Flickr30K and COCO were detected in the LAION-400M dataset. This data contamination issue raises concerns about the validity of downstream benchmarks when a large number of unlabeled images are used for pre-training. While it is possible to identify and remove all duplicates for existing benchmarks, it may be infeasible to do so on already pre-trained models for future benchmarks or in real-world applications. Nonetheless, we believe that this issue should not hinder progress in data scaling for future representation learning studies.

[3]The `test` set images are publicly available. We have permission to access the annotations.

A.4. Implementation Details

In this section, we summarize the training / evaluation settings, configurations, and hyper-parameters.

A.4.1 MIM pre-training

EVA-02 MIM pre-training setting. See Table 22.

A.4.2 Image Classification

TrV throughput. See Table 21.

Intermediate fine-tuning setting for IN-21K. See Table 23.

Fine-tuning setting for IN-1K (*w/* IN-21K intermediate fine-tuning). See Table 24.

Fine-tuning setting for IN-1K (*w/o* IN-21K intermediate fine-tuning). See Table 25.

A.4.3 Contrastive Language-Image Pre-training

EVA-02 enhanced CLIP training setting. See Table 26.

A.4.4 Object Detection and Instance Segmentation

O365 intermediate fine-tuning. See Table 27.

COCO head-to-head comparisons. See Table 28.

LVIS head-to-head comparisons. See Table 29.

COCO system-level comparisons (*w/o* O365 intermediate fine-tuning). See Table 30.

LVIS system-level comparisons (*w/o* O365 intermediate fine-tuning). See Table 31.

COCO system-level comparisons (*w/* O365 intermediate fine-tuning). See Table 32.

LVIS system-level comparisons (*w/* O365 intermediate fine-tuning). See Table 33.

A.4.5 Semantic Segmentation

Using UperNet on ADE20K. See Table 34.

Using Mask2Former on COCO-Stuff-164K. See Table 35.

Using Mask2Former on ADE20K. See Table 36.

config	EVA-02-Ti / -S / -B / -L
enc. weight initialization	JAX random initialization [14, 51]
MIM teacher	EVA-CLIP vision encoder [45]
image data source	IN-21K / IN-21K / IN-21K / Merged-38M
peak learning rate	3e-3 / 3e-3 / 1.5e-3 / 1.5e-3
learning rate schedule	cosine decay
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.98, 1e-6
weight decay	0.05
input resolution	224^2
patch size	14^2
masking ratio	40%
batch size	4k / 4k / 2k / 2k
training steps	0.85M / 0.85M / 1M / 1M
training epochs	240 / 240 / 150 / 56
warmup epochs	1
drop path [61]	0.0 / 0.0 / 0.0 / 0.1
random resized crop	(0.2, 1)
numerical precision	`DeepSpeed fp16` [102]
ZeRO optimizer [101]	stage 0 or 1

Table 22: MIM pre-training setting.

config	EVA-02-B / -L
enc. weight initialization	MIM pre-trained EVA-02 (Table 22)
peak learning rate	3e-4
layer-wise lr decay [34, 5]	0.70 / 0.75
learning rate schedule	cosine decay
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.05
input resolution	448^2
patch size	14^2
batch size	2048
training epochs	40 / 30
warmup epochs	1
drop path [61]	0.10 / 0.15
label smoothing [114]	0.1
augmentation	RandAug (9, 0.5) [36]
random resized crop	(0.2, 1)
numerical precision	`DeepSpeed fp16` [102]
ZeRO optimizer [101]	stage 0 or 1
ema [94]	✗
cutmix [141]	✗
mixup [143]	✗
random erasing [146]	✗

Table 23: Intermediate fine-tuning setting for **IN-21K**.

config	EVA-02-B / -L
enc. weight initialization	IN-21K fine-tuned EVA-02 (Table 23)
peak learning rate	5e-5 / 2e-5
layer-wise lr decay [34, 5]	0.80 / 0.85
learning rate schedule	cosine decay
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.05
input resolution	448^2
patch size	14^2
batch size	512
training epochs	15 / 20
warmup epochs	2
drop path [61]	0.15
label smoothing [114]	0.2
augmentation	RandAug (9, 0.5) [36]
random resized crop	(0.08, 1)
test crop ratio	1.0
numerical precision	`DeepSpeed fp16` [102]
ZeRO optimizer [101]	stage 0 or 1
ema [94]	0.9999
cutmix [141]	✗
mixup [143]	✗
random erasing [146]	✗

Table 24: Fine-tuning setting for **IN-1K** (***w/*** **IN-21K** intermediate fine-tuning).

config	EVA-02-Ti / -S / -B / -L
enc. weight initialization	MIM pre-trained EVA-02 (Table 22)
peak learning rate	2e-4 / 1e-4 / 1e-4 / 7e-5
layer-wise lr decay [34, 5]	0.90 / 0.80 / 0.70 / 0.80
learning rate schedule	cosine decay
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.05
input resolution	336^2 / 336^2 / 448^2 / 448^2
patch size	14^2
batch size	1024
training epochs	100 / 100 / 30 / 30
warmup epochs	5 / 5 / 3 / 3
drop path [61]	0.10 / 0.10 / 0.10 / 0.15
label smoothing [114]	0.1 / 0.1 / 0.1 / 0.2
augmentation	RandAug (9, 0.5) [36]
random resized crop	(0.08, 1)
test crop ratio	1.0
numerical precision	`DeepSpeed fp16` [102]
ZeRO optimizer [101]	stage 0 or 1
ema [94]	0.9999
cutmix [141]	✗
mixup [143]	✗
random erasing [146]	✗

Table 25: Fine-tuning setting for **IN-1K** (***w/o*** **IN-21K** intermediate fine-tuning).

config	EVA-02-B / -L / -L+
image enc. weight init.	EVA-02-B / -L / EVA-02-CLIP-L
text enc. weight init.	OpenAI CLIP-B / -L / EVA-02-CLIP-L
image-text data	LAION-1.6B [106] + COYO-0.4B [16]
image enc. peak learning rate	2e-4 / 4e-4 / 4e-4
image enc. layer-wise lr decay [34, 5]	0.75 / 0.85 / 0.75
text enc. peak learning rate	2e-5 / 4e-5 / 4e-5
text enc. layer-wise lr decay [34, 5]	0.75 / 0.75 / 0.65
learning rate schedule	cosine decay
optimizer	LAMB [137]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.98, 1e-6
weight decay	0.05
input resolution	224^2 / 224^2 / 336^2
patch size	16^2 / 14^2 / 14^2
batch size	131k / 131k / 61k
samples seen	8B / 4B / 2B
random resized crop	(0.9, 1)
numerical precision	`DeepSpeed fp16` [102]
ZeRO optimizer [101]	stage 1
drop path [61]	✗
FLIP training [74]	✗
ema [94]	✗
image augmentation	✗
image cutmix [141]	✗
image mixup [143]	✗
image random erasing [146]	✗

Table 26: **EVA-02** enhanced Contrastive Language-Image Pre-training (CLIP) setting.

config	EVA-02-L
enc. weight initialization	MIM pre-trained EVA-02 (Table 22)
learning rate	6e-5
layer-wise lr decay	0.8
batch size	160
training steps	400k
learning rate schedule	lr step at [320k, 360k]
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.1
LSJ [48] crop size	1536^2
patch size	16^2
attention window size	16^2
#global attention blocks	evenly 8 blocks
drop path	0.4
numerical precision	`PyTorch amp fp16` [91]
ema [94]	✗

Table 27: **O365** object detection and instance segmentation **intermediate fine-tuning** setting based on ViTDet [75].

config	EVA-02-B / -L
enc. weight initialization	MIM pre-trained EVA-02 (Table 22)
learning rate	5e-5 / 6e-5
layer-wise lr decay	0.7 / 0.8
batch size	128 / 144
training steps	60k
learning rate schedule	lr step at [48k, 54k]
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.1
LSJ [48] crop size	1024^2
patch size	16^2
attention window size	16^2
#global attention blocks	evenly 4 blocks
drop path	0.1 / 0.4
test score threshold	0.05
max numbers of detection	100
numerical precision	`PyTorch amp fp16` [91]
softnms [12]	✗
maskness scoring [62, 125]	✗
ema [94]	✗

Table 28: **COCO** object detection and instance segmentation, **head-to-head** comparisons setting based on ViTDet [75].

config	EVA-02-B / -L
enc. weight initialization	MIM pre-trained EVA-02 (Table 22)
learning rate	1e-4
layer-wise lr decay	0.7 / 0.8
batch size	128
training steps	50k / 40k
learning rate schedule	lr step at [40k, 45k] / [32k, 36k]
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.1
LSJ [48] crop size	1024^2
patch size	16^2
attention window size	16^2
#global attention blocks	evenly 4 blocks
drop path	0.1 / 0.4
test score threshold	0.02
numerical precision	`PyTorch amp fp16` [91]
softnms [12]	✗
maskness scoring [62, 125]	✗
ema [94]	✗

Table 29: **LVIS** object detection and instance segmentation, **head-to-head** comparisons setting based on ViTDet [75].

config	EVA-02-B / -L
enc. weight initialization	MIM pre-trained EVA-02 (Table 22)
learning rate	5e-5
layer-wise lr decay	0.7 / 0.8
batch size	128
training steps	60k
learning rate schedule	lr step at [48k, 54k]
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.1
LSJ [48] crop size	1536^2
patch size	16^2
attention window size	32^2
#global attention blocks	evenly 6 / 8 blocks
drop path	0.1 / 0.4
test score threshold	0.00
max numbers of detection	100
softnms [12]	IoU threshold = 0.6
maskness scoring [62, 125]	maskness threshold = 0.5 (instance seg only)
ema [94]	✗
numerical precision	`PyTorch amp fp16` [91]

Table 30: **COCO** object detection and instance segmentation, **system-level** comparisons setting based on ViTDet [75] (***w/o*** O365 intermediate fine-tuning).

config	EVA-02-L
enc. weight initialization	O365 fine-tuned EVA-02 (Table 27)
learning rate	4e-5
layer-wise lr decay	0.8
batch size	64
training steps	40k
learning rate schedule	constant
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.1
LSJ [48] crop size	1536^2
patch size	16^2
attention window size	16^2
#global attention blocks	evenly 8 blocks
drop path	0.3
test score threshold	0.00
max numbers of detection	100
softnms [12]	IoU threshold = 0.6
maskness scoring [62, 125]	maskness threshold = 0.5 (instance seg only)
numerical precision	`PyTorch amp fp16` [91]
ema [94]	0.9999

Table 32: **COCO** object detection and instance segmentation, **system-level** comparisons setting based on ViTDet [75] (***w/*** O365 intermediate fine-tuning).

config	EVA-02-L
enc. weight initialization	MIM pre-trained EVA-02 (Table 22)
learning rate	1e-4
layer-wise lr decay	0.8
batch size	128
training steps	40k
learning rate schedule	lr step at [32k, 36k]
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.1
LSJ [48] crop size	1536^2
patch size	16^2
attention window size	32^2
#global attention blocks	evenly 8 blocks
drop path	0.4
test score threshold	0.02
max numbers of detection	300
softnms [12]	IoU threshold = 0.6
maskness scoring [62, 125]	maskness threshold = 0.5
numerical precision	`PyTorch amp fp16` [91]
ema [94]	✗

Table 31: **LVIS** object detection and instance segmentation, **system-level** comparisons setting based on ViTDet [75] (***w/o*** O365 intermediate fine-tuning).

config	EVA-02-L
enc. weight initialization	O365 fine-tuned EVA-02 (Table 27)
learning rate	4e-5
layer-wise lr decay	0.8
batch size	64
training steps	70k
learning rate schedule	constant
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.1
LSJ [48] crop size	1536^2
patch size	16^2
attention window size	16^2
#global attention blocks	evenly 8 blocks
drop path	0.3
test score threshold	0.02
max numbers of detection	1000
softnms [12]	IoU threshold = 0.6
maskness scoring [62, 125]	maskness threshold = 0.5
numerical precision	`PyTorch amp fp16` [91]
ema [94]	0.9999

Table 33: **LVIS** object detection and instance segmentation, **system-level** comparisons setting based on ViTDet [75] (***w/*** O365 intermediate fine-tuning).

config	EVA-02-B / -L / -L+
enc. weight initialization	MIM pre-trained EVA-02 (Table 22)
learning rate	6e-5 / 4e-5 / 4e-5
layer-wise lr decay	0.85 / 0.90 / 0.90
batch size	32 / 16 / 16
training steps	60k / 80k / 80k
learning rate schedule	linear decay
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.05
crop size	512^2 / 512^2 / 640^2
patch size	16^2
drop path	0.15 / 0.20 / 0.20
seg head dim	768 / 1024 / 1536
numerical precision	`PyTorch amp fp16` [91]
ViT-Adapter [28]	✗

Table 34: Semantic segmentation on **ADE20K** using **UperNet** [132].

config	EVA-02-L
enc. weight initialization	MIM pre-trained EVA-02 (Table 22)
learning rate	2e-5
layer-wise lr decay	0.9
batch size	16
training steps	120k
learning rate schedule	linear decay
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.05
crop size	640^2
patch size	16^2
drop path	0.2
seg head dim	1024
seg head #enc. & #dec.	6 & 9
numerical precision	`PyTorch amp fp16` [91]
ViT-Adapter [28]	✗

Table 35: Semantic segmentation on **COCO-Stuff-164K** using **Mask2Former** [29].

config	EVA-02-L
enc. weight initialization	COCO-Stuff fine-tuned EVA-02 (Table 35)
learning rate	2e-5
layer-wise lr decay	0.9
batch size	64
training steps	20k
learning rate schedule	linear decay
optimizer	AdamW [64, 84]
optimizer hyper-parameters	$\beta_1, \beta_2, \epsilon$ = 0.9, 0.999, 1e-8
weight decay	0.05
crop size	640^2
patch size	16^2
drop path	0.2
seg head dim	1024
seg head #enc. & #dec.	6 & 9
numerical precision	`PyTorch amp fp16` [91]
ViT-Adapter [28]	✗

Table 36: Semantic segmentation on **COCO-Stuff-164K** using **Mask2Former** [29].

References

[1] Reaching 80 zero-shot accuracy with openclip: Vit-g/14 trained on laion-2b. `https://laion.ai/blog/giant-openclip/`. 2

[2] A standalone library for adding rotary embeddings to transformers in pytorch. `https://github.com/lucidrains/rotary-embedding-torch`. 10

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. 1, 9

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3, 4

[5] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 1, 3, 4, 9, 10, 11, 12, 13

[6] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019. 4, 5, 6

[7] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *CVPR*, 2014. 6

[8] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020. 4, 5

[9] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *CVPR*, 2022. 5, 9

[10] Stella Biderman, Sid Black, Charles Foster, Leo Gao, Eric Hallahan, Horace He, Ben Wang, and Phil Wang. Rotary embeddings: A relative revolution, 2021. 10

[11] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022. 10

[12] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, 2017. 7, 13, 14

[13] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 6

[14] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. 3, 12

[15] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 9

[16] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. `https://github.com/kakaobrain/coyo-dataset`, 2022. 6, 13

[17] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018. 2, 7, 8

[18] Yuxuan Cai, Yizhuang Zhou, Qi Han, Jianjian Sun, Xiangwen Kong, Jun Li, and Xiangyu Zhang. Reversible column networks. In *ICLR*, 2023. 1, 2, 5, 8

[19] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *TPAMI*, 2019. 7

[20] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 6

[21] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 6

[22] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 6

[23] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 4

[24] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, 2019. 8

[25] Qiang Chen, Jian Wang, Chuchu Han, Shan Zhang, Zexian Li, Xiaokang Chen, Jiahui Chen, Xiaodi Wang, Shuming Han, Gang Zhang, et al. Group detr v2: Strong object detector with encoder-decoder pretraining. *arXiv preprint arXiv:2211.03594*, 2022. 8

[26] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, et al. Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*, 2023. 4, 10

[27] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 1, 9, 10

[28] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 9, 15

[29] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv preprint arXiv:2112.01527*, 2021. 8, 15

[30] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017. 6

[31] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *arXiv preprint arXiv:2212.07143*, 2022. 1, 11

[32] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 3, 10

[33] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 6

[34] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020. 12, 13

[35] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTAT*, 2011. 6

[36] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 2020. 12

[37] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022. 10

[38] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *ICML*, 2017. 2, 3, 10

[39] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. *arXiv preprint arXiv:2302.05442*, 2023. 1, 5

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3, 4

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 9

[42] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 3, 4, 9, 10

[43] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 6

[44] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Code and models of eva: Exploring the limits of masked visual representation learning at scale. **https://github.com/baaivision/EVA**, 2022. 4

[45] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. *arXiv preprint arXiv:2211.07636*, 2022. 1, 2, 3, 4, 5, 6, 8, 9, 12

[46] Yuxin Fang, Shusheng Yang, Shijie Wang, Yixiao Ge, Ying Shan, and Xinggang Wang. Unleashing vanilla vision transformer with masked image modeling for object detection. *arXiv preprint arXiv:2204.02964*, 2022. 3

[47] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPRW*, 2004. 6

[48] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 7, 13, 14

[49] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *ICONIP*, 2013. 6

[50] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 2, 7, 11

[51] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. 3, 4, 5, 9, 12

[52] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 7

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 9

[54] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 2019. 6

[55] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *CVPR*, 2021. 4, 5, 6

[56] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3, 10

[57] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021. 4, 5, 6

[58] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5, 10

[59] Zejiang Hou, Fei Sun, Yen-Kuang Chen, Yuan Xie, and Sun-Yuan Kung. Milan: Masked image pretraining on language assisted representation. *arXiv preprint arXiv:2208.06049*, 2022. 3

[60] Ronghang Hu, Shoubhik Debnath, Saining Xie, and Xinlei Chen. Exploring long-sequence masked autoencoders. *arXiv preprint arXiv:2210.07224*, 2022. 11

[61] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 12, 13

[62] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019. 7, 13, 14

[63] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

[64] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 10, 12, 13, 14, 15

[65] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013. 6

[66] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[67] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 4

[68] The LAION and Timm Team. Pytorch image models: vit-base-patch16-clip-384px-laion2b-ft-in12k-in1k. **https://huggingface.co/timm/vit_base_patch16_clip_384.laion2b_ft_in12k_in1k**, 2022. 5

[69] The LAION and Timm Team. Pytorch image models: vit-huge-patch14-clip-336px-laion2b-ft-in12k-in1k. **https://huggingface.co/timm/vit_huge_patch14_clip_336.laion2b_ft_in12k_in1k**, 2022. 5

[70] The LAION and Timm Team. Pytorch image models: vit-large-patch14-clip-336px-laion2b-ft-in12k-in1k. https://huggingface.co/timm/vit_large_patch14_clip_336.laion2b_ft_in12k_in1k, 2022. 5

[71] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 6

[72] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022. 4, 10

[73] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 6, 9

[74] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. *arXiv preprint arXiv:2212.00794*, 2022. 13

[75] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022. 3, 7, 8, 13, 14

[76] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*, 2021. 7

[77] Tingting Liang, Xiaojie Chu, Yudong Liu, Yongtao Wang, Zhi Tang, Wei Chu, Jingdong Chen, and Haibin Ling. Cbnet: A composite backbone network architecture for object detection. *TIP*, 2022. 8

[78] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 4, 6, 7

[79] Xingbin Liu, Jinghao Zhou, Tao Kong, Xianming Lin, and Rongrong Ji. Exploring target representations for masked autoencoders. *arXiv preprint arXiv:2209.03917*, 2022. 3, 10

[80] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 1, 4, 9

[81] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022. 1, 2, 8

[82] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 8

[83] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 9

[84] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 4, 10, 12, 13, 14, 15

[85] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 6

[86] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *arXiv preprint arXiv:2206.02680*, 2022. 5

[87] Yurii Evgen'evich Nesterov. A method of solving a convex programming problem with convergence rate o\bigl(k^2\bigr). In *Doklady Akademii Nauk*, 1983. 4, 10

[88] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 6

[89] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022. 9

[90] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 6

[91] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 4, 13, 14, 15

[92] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 3, 4, 5, 8, 10, 11

[93] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Hanxiao Liu, Adams Wei Yu, Minh-Thang Luong, Mingxing Tan, and Quoc V Le. Combined scaling for zero-shot transfer learning. *arXiv preprint arXiv:2111.10050*, 2021. 1

[94] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 1992. 8, 12, 13, 14

[95] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 3, 6, 9

[96] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 9

[97] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 9

[98] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20*, 2020. 4

[99] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 3, 10

[100] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 6, 9

[101] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. 12, 13

[102] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD*, 2020. 4, 12, 13

[103] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet?, 2019. 3

[104] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019. 4, 5, 6

[105] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 2, 4, 5, 6

[106] 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. *arXiv preprint arXiv:2210.08402*, 2022. 6, 13

[107] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019. 4, 8, 11

[108] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 4

[109] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 3, 11

[110] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. 2, 3, 10

[111] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6

[112] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 2012. 6

[113] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 2, 3, 10

[114] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 12

[115] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3, 4, 9

[116] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *ECCV*, 2022. 5, 8

[117] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, 2021. 3, 11

[118] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 1, 3

[119] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *MICCAI*, 2018. 6

[120] Shakti N Wadekar and Abhishek Chaurasia. Mobilevitv3: Mobile-friendly vision transformer with simple and effective fusion of local, global and input features. *arXiv preprint arXiv:2209.15159*, 2022. 5

[121] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *NeurIPS*, 2019. 4, 5, 6

[122] Hongyu Wang, Shuming Ma, Shaohan Huang, Li Dong, Wenhui Wang, Zhiliang Peng, Yu Wu, Payal Bajaj, Saksham Singhal, Alon Benhaim, et al. Foundation transformers. *arXiv preprint arXiv:2210.06423*, 2022. 2, 3, 10

[123] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 1, 3, 4, 5, 8

[124] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. *arXiv preprint arXiv:2211.05778*, 2022. 1, 2, 5, 8

[125] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Solo: A simple framework for instance segmentation. *TPAMI*, 2021. 7, 13, 14

[126] Longhui Wei, Lingxi Xie, Wengang Zhou, Houqiang Li, and Qi Tian. Mvp: Multimodality-guided visual pre-training. *arXiv preprint arXiv:2203.05175*, 2022. 3

[127] Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Contrastive learning rivals masked image modeling in fine-tuning via feature distillation. *arXiv preprint arXiv:2205.14141*, 2022. 5, 8

[128] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021. 9

[129] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*, 2023. 5, 8

[130] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, 2022. 6

[131] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 6

[132] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 8, 15

[133] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*, 2021. 9

[134] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Yixuan Wei, Qi Dai, and Han Hu. On data scaling in masked image modeling. *arXiv preprint arXiv:2206.04664*, 2022. 1

[135] Chenglin Yang, Siyuan Qiao, Qihang Yu, Xiaoding Yuan, Yukun Zhu, Alan Yuille, Hartwig Adam, and Liang-Chieh Chen. Moat: Alternating mobile convolution and attention brings strong vision models. In *ICLR*, 2023. 5

[136] Jianwei Yang, Chunyuan Li, and Jianfeng Gao. Focal modulation networks. *arXiv preprint arXiv:2203.11926*, 2022. 8

[137] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019. 13

[138] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity

metrics for semantic inference over event descriptions. *TACL*, 2014. 6

[139] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 1

[140] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 1

[141] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 4, 12, 13

[142] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022. 1

[143] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 4, 12, 13

[144] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 8

[145] Xinyu Zhang, Jiahui Chen, Junkun Yuan, Qiang Chen, Jian Wang, Xiaodi Wang, Shumin Han, Xiaokang Chen, Jimin Pi, Kun Yao, et al. Cae v2: Context autoencoder with clip target. *arXiv preprint arXiv:2211.09799*, 2022. 3

[146] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020. 4, 12, 13

[147] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2018. 2, 4, 7, 8

[148] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 9

[149] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021. 7