Chain-of-Models Pre-Training: Rethinking Training Acceleration of Vision Foundation Models

Supplementary Materials

A. Datasets

Pre-training Datasets. CC3M [25] and CC12M [3] are two public image-text datasets, which are widely used in efficient CLIP [10, 18, 28]. Originally, CC3M and CC12M datasets consist of around 3.3M and 12.4M image-text pairs, respectively. However, since some URLs have become invalid, we use all valid samples from each dataset, 83.0% and 81.9% of the original CC3M and CC12M, respectively. The details are shown in Table A. We incorporate long captions generated by MLLM from [28] to enhance the diversity of captions. Specifically, for each image, we select additional 15 sub-captions split from the long caption to augment the original raw data. In this way, we can expand the 2.7M image-text pairs in CC3M to approximately 44.1M image-text pairs, and the 10.2M in CC12M to approximately 162.8M image-text pairs. We then combine the augmented CC3M and augmented CC12M to the final Merged-15M, which consists of 206.8M image-text pairs. Under such scales, the models pre-trained by the baseline can significantly move towards the results pre-trained on extremely large datasets, such as LAION-400M [23]. This approach allows us to achieve plausible acceleration ratios.

Table A. **CC3M, CC12M and our Merged-15M datasets used for pre-training in this paper.** Note that we only report the number of samples in the training set.

Dataset	# Total Pairs	# Collected Pairs	# Augmented Pairs
CC3M	3,318,333	2,753,427	44,054,832
CC12M	12,423,374	10,173,631	162,778,096
Merged-15M	-	12,927,058	206,832,928

Zero-shot Classification and Retrieval Datasets. As we mentioned in the main paper, we follow the standard evaluation protocols [6] for all datasets: ImageNet-1K [7] for zero-shot classification, COCO [15] for zero-shot image-text retrieval, and VTAB+ [24] for testing zero-shot transfer capabilities. Specifically, VTAB+ is one of the largest zero-shot benchmarks so far, which consists of 35 datasets, as shown in Table B. The evaluation is conducted on the official CLIP benchmark [5].

Datasets for Open-vocabulary Semantic Segmentation Tasks. We follow the training and evaluation protocol of SAN. i) We train the model on the training set of COCO-Stuff [2], which contains 118K images with 171 annotated classes. ii) We test the model on ADE-847 [30], ADE-150 [30], PC-459 [21], PC-59 [21], and VOC-20 [9]. Specifically, ADE-150 and ADE-847 have the same 2K validation images, but have different numbers of categories, 150 and 847 annotated classes, respectively. Similarly, PC-59 and PC-459 have 5K validation images, but have 59 and 459 classes, respectively. And VOC-20 contains 1449 images with 20 classes.

Table B. **Datasets in VTAB+ with abbreviations and test sizes.**

Dataset	Abbr.	Test size
ImageNet-1K	-	50,000
ImageNet-v2	-	10,000
ImageNet-R	-	30,000
ImageNet Sketch	-	50,889
ObjectNet	-	18,574
ImageNet-A	-	7,500
CIFAR-10	-	10,000
CIFAR-100	-	10,000
MNIST	-	10,000
Oxford Flowers 102	Flowers	6,149
Stanford Cars	Cars	8,041
SVHN	-	26,032
Facial Emotion Recognition 2013	FER2013	7,178
RenderedSST2	RSST2	1,821
Oxford-IIIT Pets	Pets	3,669
Caltech-101	-	6,085
Pascal VOC 2007 Classification	VOC2007	14,976
SUN397	-	108,754
FGVC Aircraft	Aircraft	3,333
Country211	-	21,100
Describable Textures	DTD	1,880
GTSRB	-	12,630
STL10	-	8,000
Diabetic Retinopathy	Retino	42,670
EuroSAT	-	5,400
RESISC45	-	6,300
PatchCamelyon	PCAM	32,768
CLEVR Counts	-	15,000
CLEVR Object Distance	CLEVR Dist	15,000
DSPRITES Orientation	DSPRITES Orient	73,728
DSPRITES Position	DSPRITES pos	73,728
SmallNORB Elevation	SmallNORB Elv	12,150
SmallNORB Azimuth	SmallNORB Azim	12,150
DMLAB	-	22,735
KITTI closest vehicle distance	KITTI Dist	711

Datasets for Vision-Language Tasks. The training process of LLaVA-1.5 [17] consists of two main stages: quick feature alignment training and visual instruction fine-tuning. For feature alignment training, we utilize the LLaVA-Pretrain LCS-558K dataset [17], comprising 558K image-text pairs filtered from the original LAION [24], CC12M [3], and SBU [22] datasets, with BLIP-generated captions [13]. Subsequently, visual instruction tuning is performed using the

Table C. **Detailed settings of baseline pre-training and chain-of-models pre-training on CC3M and Merged-15M datasets.**

(a) Baseline training strategy.

Configuration	Setting
Dataset	CC3M | Merged-15M
Batch Size	1024
Optimizer	AdamW
Optimizer Hyper-Parameters	β_1, β_2=(0.9, 0.98), ϵ=1e-8
Learning Rate Schedule	Cosine Decay
Initial Learning Rate	1e-3
Weight Decay	0.1
Training Epochs	128|64
Warmup Iterations	8000
Precision	AMP

(b) CoM-PT training strategy.

Configuration	Setting
Dataset	CC3M | Merged-15M
Batch Size	1024
Optimizer	AdamW
Optimizer Hyper-Parameters	β_1, β_2=(0.9, 0.98), ϵ=1e-8
Learning Rate Schedule	Cosine Decay
Initial Learning Rate	1e-3
Weight Decay	0.1
Training Epochs	*In Table D*
Warmup Iterations	1000
Precision	AMP

Table D. **Training epochs for different models in main experiments.**

Dataset	ViT Family					Swin Family			
	ViT-T/16	ViT-S/16	ViT-M/16	ViT-B/16	ViT-L/16	Swin-T	Swin-S	Swin-B	Swin-L
CC3M	128	24	–	18	15	128	24	16	12
Merged-15M	–	64	21	17	15	–	–	–	–

LLaVA-Instruct-150K dataset [17], which contains 150K GPT-4-generated multimodal instruction-following samples [1]. We assess the model's performance across several downstream tasks: TextVQA [26], which evaluates the model's ability to read and reason about text within images; ScienceQA [20], a benchmark with multimodal multiple-choice science questions testing the model's scientific reasoning capabilities; POPE [14], designed to evaluate object hallucination in vision-language models; and VQAv2 [11], a dataset containing open-ended questions about images, requiring models to integrate visual understanding, language processing, and commonsense knowledge.

B. Experimental Setups

B.1. Contrastive Language Image Pre-training

Compute Infrastructure. All experiments are conducted on a single node equipped with 8× NVIDIA A100 (80GB) GPUs.

Implementation Codebase. Our chain-of-models pre-training is implemented using the OpenCLIP framework [6].

Hyper-parameter Settings. The training loss for our experiments is defined in Equation 1 and Equation 4 of the main paper, which only has one hyper-parameter α. Based on the results in the Section D.2, we choose $\alpha = 500$ (corresponding to $r = 0.1$ in Figure D) for all experiments.

Training Strategy. Following the standard training protocol of OpenCLIP [6], automatic mixed precision (AMP) is applied as default. Besides that, to ensure fair acceleration ratios, all experiments on CC3M and Merged-15M datasets are typically conducted with the same setting, including data processing pipeline, optimizer, initial learning rate, batch size, *etc*. Details are shown in Table C.

Training Arrangement of CoM-PT. Derived from the ablation study regarding training arrangement in Section 5.3.2 of the main paper, we have summarized: *"training epochs allocation along the model chain decreases linearly as model size increases exponentially."* Following this principle, we arrange the training epochs for each model as shown in Table D. Notably, compared to the interpolated minimum epochs on CC3M (23.45, 18.28, and 12.34 for ViT-S/16, ViT-B/16, and ViT-L/16, respectively), we allocate additional epochs to the largest model in practice to obtain superior performance in the main experiments.

B.2. Side Fine-tuning on SAN

Compute Infrastructure. The experiments are conducted on a single node equipped with 8× NVIDIA A100 (80GB) GPUs.

Implementation Codebase. We conduct experiments using the official repository of SAN [27].

Selection of Segmentation Frameworks and Backbones. We choose SAN as the segmentation framework based on the pre-trained ViT-B/16 and ViT-L/16 backbones.

Training Strategy. We directly use the official fine-tuning strategy of SAN. Details are shown in Table E.

B.3. Visual Instruction Fine-tuning on LLaVA

Compute Infrastructure. The experiments are conducted on a single node equipped with 8× NVIDIA A100 (80GB) GPUs.

Implementation Codebase. We conduct the experiments using the official repository of LLaVA [16, 17].

Selection of MLLM and Backbones. We select LLaVA-1.5-7B [17] as the base MLLM, equipped with either ViT-B/16 or ViT-L/16 as the image encoder.

Training Strategy. The training consists of two stages. i) In the first stage, we align the features of the image encoder

Table E. **Detailed settings of fine-tuning on COCO-Stuff for open-vocabulary semantic segmentation tasks.**

(a) SAN with ViT-B/16 backbone.

Configuration	Setting
Segmentation Framework	SAN
Backbone	ViT-B/16
Feature Fusion Blocks	1-9
Mask Recognition Blocks	10-12
Input Resolution	640
Batch Size	32
Optimizer	AdamW
Optimizer Hyper-Parameters	β_1, β_2=(0.9, 0.98), ϵ=1e-8
Learning Rate Schedule	Poly Decay
Initial Learning Rate	1e-3
Learning Rate Decay	0.9
Weight Decay	1e-4
Training Iterations	60K
Random Resize Crop	✓
Precision	AMP

(b) SAN with ViT-L/16 backbone.

Configuration	Setting
Segmentation Framework	SAN
Backbone	ViT-L/16
Feature Fusion Blocks	1-18
Mask Recognition Blocks	19-24
Input Resolution	640
Batch Size	32
Optimizer	AdamW
Optimizer Hyper-Parameters	β_1, β_2=(0.9, 0.98), ϵ=1e-8
Learning Rate Schedule	Poly Decay
Initial Learning Rate	1e-3
Learning Rate Decay	0.9
Weight Decay	1e-4
Training Iterations	60K
Random Resize Crop	✓
Precision	AMP

Table F. **Detailed settings of feature alignment training and LoRA-based visual instruction fine-tuning.**

(a) Feature alignment training.

Configuration	Setting
Multi-Modality Large Language Model	LLaVA-1.5-7B
Vision Encoder	ViT-B/16 | ViT-L/16
Feature Alignment Projector	mlp2x_gelu
Model Max Length	2048
LoRA Rank	–
LoRA Alpha	–
Per Device Batch Size	48
Gradient Accumulation Steps	1
Optimizer	AdamW
Feature Alignment Projector LR	2e-3
LoRA LR	–
Learning Rate Schedule	Cosine Decay
Warmup Ratio	0.03
Training Epochs	1
BF16	✓

(b) LoRA-based visual instruction fine-tuning.

Configuration	Setting
Multi-Modality Large Language Model	LLaVA-1.5-7B
Vision Encoder	ViT-B/16 | ViT-L/16
Feature Alignment Projector	mlp2x_gelu
Model Max Length	2048
LoRA Rank	128
LoRA Alpha	256
Per Device Batch Size	16
Gradient Accumulation Steps	1
Optimizer	AdamW
Feature Alignment Projector LR	2e-5
Lora LR	2e-4
Learning Rate Schedule	Cosine Decay
Warmup Ratio	0.03
Training Epochs	1
BF16	✓

and the LLM by fully fine-tuning a lightweight two-layer MLP vision-language connector. The experimental setup is detailed in Table F(a). ii) In the second stage, we perform efficient visual instruction fine-tuning on the Vicuna-7B-v1.5 [29] using a LoRA-based [12] approach to enable its capability in handling vision-related tasks. The corresponding configuration is provided in Table F(b).

C. Architectural Specifications and Training Complexity of VFM Families

C.1. Model Architectures

Recall that we chose the standard ViT family and the Swin family in the main experiments. Detailed information about these two model families is provided in Table G. Note that the forward MACs are calculated based on 224×224 resolution images and 77 text tokens. Yet, with the inclusion of additional sub-captions, the forward MACs from the text encoder become $4\times$ than only using the original caption.

C.2. Calculation of Training Complexity

As we stated in the main paper, the training complexity C_t comprises the forward complexity C_f, the backward complexity C_b, and the parameter update complexity C_u concerning both models and optimizers. The training complexity C_t is formulated as:

$$C_t = C_b + C_f + C_u. \tag{1}$$

Considering multiply–accumulate operations (MACs), the backward complexity is approximately $2\times$ larger than the forward complexity for each layer, except the first layer. For a network with p layers, we can derive the following

Table G. **Architecture specifications for VFM families.** *Forward MACs* and *Training MACs* represent the computational complexity of the model for a sample (an image and its corresponding 4 text descriptions) in our experiment.

Model Family	Model	Image Encoder			Text Encoder			Overall		
		Width	Depth	Param (M)	Width	Depth	Param (M)	Param (M)	Forward MACs (G)	Training MACs (G)
ViT Family	ViT-T/16	192	12	5.62	256	12	9.48	15.10	5.82	17.57
	ViT-C/16	256	12	9.86	320	12	14.80	24.66	9.23	27.87
	ViT-S/16	384	12	21.81	384	12	21.29	43.10	14.44	43.65
	ViT-M/16	512	12	38.59	448	12	28.97	67.56	21.88	66.17
	ViT-B/16	768	12	86.19	512	12	37.83	124.02	35.09	106.27
	ViT-XB/16	1024	12	171.36	512	12	37.83	209.19	48.76	148.05
	ViT-L/16	1024	24	304.09	768	12	85.05	389.14	100.92	306.11
Swin Family	Swin-T	768	12	27.91	256	12	9.48	37.39	8.76	26.60
	Swin-S	768	24	49.23	384	12	21.29	70.52	18.40	55.82
	Swin-B	1024	24	87.27	512	12	37.83	125.10	32.71	98.84
	Swin-L	1536	24	195.78	768	12	85.05	280.83	73.48	222.94

equation:

$$C_b = \sum_{i=1}^{p} C_b^i = 2\sum_{i=1}^{p} C_f^i - C_f^1 = 2C_f - C_f^1, \quad (2)$$

where C_f^i and C_b^i denote the forward complexity and the backward complexity of layer i, respectively. Since we select AdamW [19] as the optimizer, the MACs of the parameter update C_u are $3\times$ the number of parameters. In this way, we can calculate the accurate C_t for each model.

For CoM-PT, the total training complexity is computed as a summation of student's full training complexity and teacher's forward complexity for inverse knowledge transfer.

D. More Ablation Studies

D.1. Inverse Weight Initialization

In this part, we address two key questions regarding our implementation of inverse weight initialization.

i) What kind of simple approach yields the best performance?

ii) Why is applying sophisticated implementations less important in our inverse knowledge transfer?

D.1.1. Exploration of Effective Simple Approaches

We explore several simple approaches for inverse weight initialization. As illustrated in Figure A, weight initialization for width is shown from (a) to (c), and weight initialization for depth is shown from (d) to (f). Then, we conduct experiments to compare the effectiveness of these approaches.

Table H. **Our simple approaches *vs.* previous sophisticated designs.** Experiments are conducted using the ViT-T/16→ViT-S/16 sub-chain, trained on the CC3M dataset.

Method	**ZS Performance**	
	ImageNet-1K	COCO
Baseline	26.48	28.34
+IFD	29.29	32.10
+IFD+IWI	30.24	**34.15**
+IFD+Net2Net [4]	30.07	33.63
+IFD+NetExpand [8]	**30.36**	34.13

To systematically evaluate these approaches, we conduct a two-stage analysis in Figure B. First, we examine width initialization methods within the ViT-T/16→ViT-S/16 sub-chain. Our findings reveal: i) weight duplication initially accelerates convergence but ultimately harms final performance; ii) simple insertion and linear interpolation improve convergence speed and final performance, with linear interpolation excelling in convergence acceleration while insertion achieves better final results. Subsequently, we study depth initialization based on simple insertion in width. We find that, compared to constant and interval insertion in depth, duplicate insertion delivers the best performance.

Based on this systematic evaluation, we adopt simple insertion for width differences and duplicate insertion for depth differences as our default implementation strategy.

D.1.2. The Efficacy of More Sophisticated Designs

We then compare our best simple approaches with previous sophisticated designs. Specifically, we implement Net2Net [4] and NetExpand [8], two representative methods for vision model expansion.

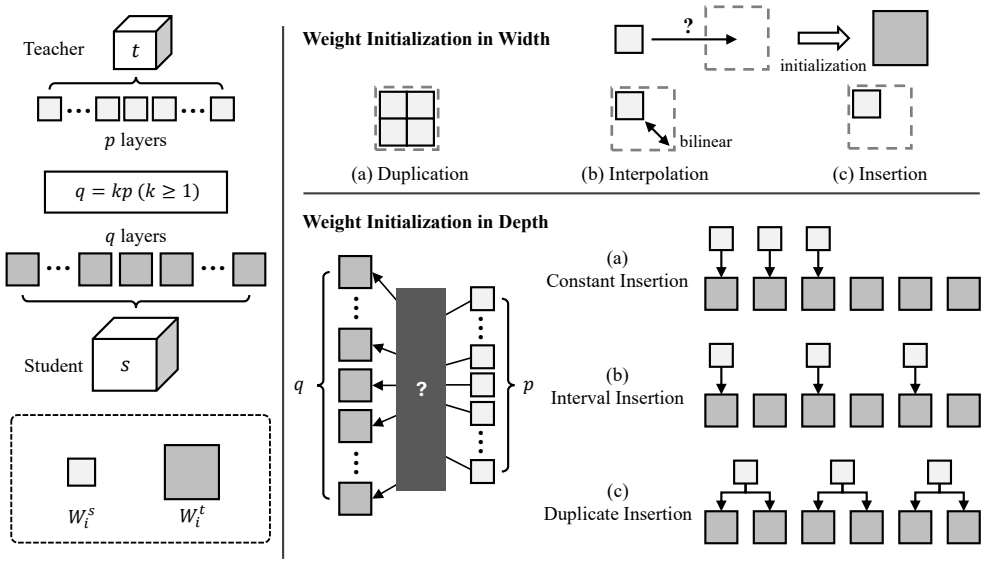


Figure A. **Detailed illustrations of weight initialization in width and depth.** First, (a)-(c) illustrates different weight initialization approaches for width differences. Duplication (a) involves copying the teacher's weights multiple times and directly assigning these copies to the student. Interpolation (b) conducts bilinear operations to expand the teacher's weights to fit the shape of the student. Insertion (c) involves directly assigning the teacher's weights to the student, with all remaining weights initialized randomly. Second, (d)-(f) show the different weight initialization methods for depth differences. (d) only initializes the first p layers across the total q layers, while the interval insertion (e) skips several layers by calculating q/p. As a combination, the duplicate insertion (f) initializes the skipped layers by duplicating the weights of the preceding layer.

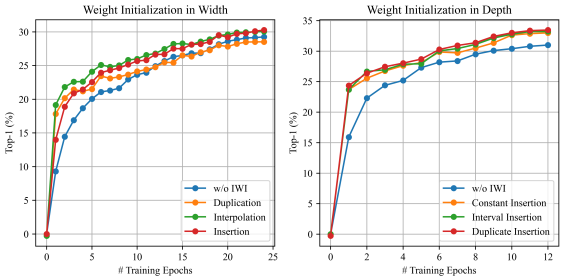


Figure B. **Ablation study on various simple weight initialization approaches.** For width initialization, experiments are conducted on the ViT-T/16→ViT-S/16 sub-chain. Progressively, with the best simple width initialization method, we explore depth initialization on the ViT-B/16→ViT-L/16 sub-chain. *Inverse feature distillation is applied as default. Details for each method are in Figure A.*

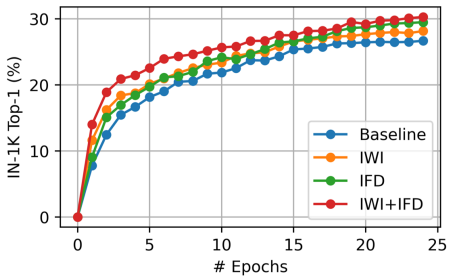


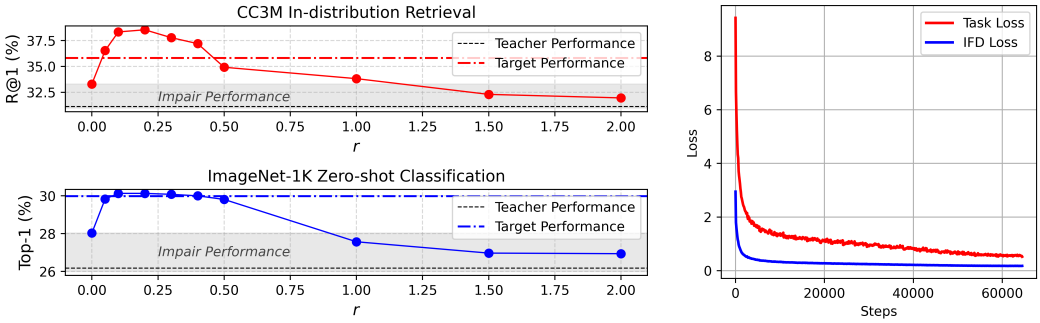
Figure C. **Validation curves of different choices in Table 4(b)** of the main paper.

Table H presents the comparative results. Our simple implementation achieves performance comparable to these sophisticated designs, delivering 3.76%|5.81% gains to the baseline on ImageNet-1K|COCO, respectively. In comparison, the sophisticated method, NetExpand, only provides a marginal 0.12% improvement to our approaches on ImageNet-1K.

These results demonstrate that our simple approach is already good enough, which strikes a promising balance between simplicity and performance.

D.2. Inverse Feature Distillation

In this part, we aim to suggest why we typically ensure $\mathcal{L}_{IFD} < \mathcal{L}_{task}$ by setting an appropriate α. Specifically, we train a series of models using CoM-PT with varying coefficients of feature distillation, and observe their performance trend on the validation set of CC3M and ImageNet-1K. As shown in Figure D, feature distillation can enhance the performance of the ViT-S/16 on both datasets when the ratio r between distillation loss and task loss is small, whereas a larger r tends to impair its performance. In the optimal setting, $\mathcal{L}_{IFD}$ is maintained at approximately 10% of $\mathcal{L}_{task}$, acting strictly as an **auxiliary loss**. This differs significantly from traditional FD, where the optimal feature distillation loss magnitude is typically larger than the task loss.



(a) Performance of ViT-S/16 under different r.

(b) Loss curves $r = 0.1$.

Figure D. **Ablation study on inverse feature distillation magnitude.** For better illustration, we define r as the ratio between the magnitude of L_{IFD} and L_{task}. The experiment is conducted on ViT-T/16→ViT-S/16 on the CC3M dataset. We show how IFD enables ViT-S/16 to achieve target performance (128-epoch individual pre-training) with only 24 epochs. The teacher model is ViT-T/16 trained individually for 128 epochs. *Inverse weight initialization is applied as default.*

Table I. **Zero-shot Top-1 accuracy (%) across the 35 datasets of the VTAB+ benchmark.** Detailed results corresponding to the VTAB+ averages presented in Table 2 of the main paper.

PT Dataset	CC3M														Merged-15M						
Models	ViT-T/16	ViT-S/16		ViT-B/16		ViT-L/16		Swin-T	Swin-S		Swin-B		Swin-L		ViT-S/16	ViT-M/16		ViT-B/16		ViT-L/16	
PT Method	baseline	baseline	CoM-PT	baseline	CoM-PT	baseline	CoM-PT	baseline	baseline	CoM-PT	baseline	CoM-PT	baseline	CoM-PT	baseline	baseline	CoM-PT	baseline	CoM-PT	baseline	CoM-PT
ImageNet-1K	26.16	30.16	30.24	31.80	31.83	33.77	34.27	33.84	35.88	36.19	36.13	36.71	36.77	37.06	43.97	45.90	45.85	47.15	47.39	48.80	49.47
ImageNet-v2	22.85	26.47	25.91	27.42	28.00	29.39	29.94	29.00	30.99	31.31	31.54	31.59	31.83	32.30	38.24	39.52	39.85	41.35	40.79	43.06	42.84
ImageNet-R	37.15	42.36	40.77	44.09	43.76	47.60	47.94	47.35	50.70	50.72	52.16	52.04	51.85	52.96	63.28	66.06	65.61	69.16	67.70	71.65	71.02
ImageNet Sketch	16.93	21.24	20.53	22.81	21.32	24.54	24.76	24.92	26.25	26.80	27.66	27.12	27.42	27.52	35.09	36.74	36.96	39.27	38.69	41.06	41.10
ObjectNet	16.05	18.02	16.95	19.41	19.46	19.73	21.45	22.67	24.71	24.07	25.46	24.43	25.31	25.50	30.56	31.70	32.99	34.94	33.29	36.92	36.24
Imagenet-A	7.48	9.79	8.13	10.11	9.51	10.73	11.29	12.09	14.68	13.76	15.24	14.75	14.99	15.15	18.28	21.57	19.75	24.89	21.63	24.55	25.85
CIFAR-10	66.53	74.42	78.38	73.50	77.74	77.81	83.06	74.02	77.39	74.59	71.21	75.95	72.85	77.59	89.08	89.43	90.69	90.36	91.28	91.66	93.52
CIFAR-100	35.38	41.87	42.19	42.27	44.60	46.03	51.75	40.86	43.47	46.21	45.58	47.84	47.51	48.72	58.88	62.07	62.53	63.87	65.69	65.21	70.17
MNIST	34.20	19.66	37.03	22.45	37.93	21.01	20.61	25.43	33.94	30.39	28.76	42.60	13.44	51.63	46.03	18.91	28.09	25.08	14.90	27.45	20.06
Flowers	15.92	19.45	19.53	23.73	20.36	19.97	22.44	19.61	19.79	20.77	25.48	20.49	24.22	21.74	22.52	29.87	25.97	32.01	29.05	32.56	29.16
Cars	2.74	4.04	3.92	5.01	4.35	5.81	4.89	4.70	5.20	4.99	5.45	4.58	4.73	4.60	16.75	19.13	18.83	22.56	21.89	23.53	23.41
SVHN	9.22	19.81	24.48	17.03	10.98	19.46	22.12	15.41	11.66	26.28	20.69	23.97	23.42	19.50	15.21	24.83	16.59	16.61	25.96	30.71	22.76
FER2013	24.28	26.68	30.80	28.75	34.30	30.61	40.62	18.84	32.56	31.72	41.36	34.79	34.10	28.25	24.30	34.15	25.56	38.53	35.69	30.55	41.08
RSST2	49.97	50.08	47.94	49.91	50.08	50.08	50.08	50.08	50.08	49.91	49.91	48.59	49.91	49.91	50.13	50.08	50.08	50.08	50.08	50.08	50.13
Pets	30.72	33.09	29.90	34.01	35.43	30.99	39.41	33.66	33.96	35.16	33.36	38.87	37.99	38.16	54.29	53.26	60.18	57.07	62.01	56.55	62.28
Caltech-101	66.24	70.96	71.64	70.94	71.57	73.74	73.03	70.60	71.65	71.83	71.39	72.88	72.69	72.74	77.47	78.39	77.47	78.23	78.85	79.75	79.85
VOC2007	43.32	40.00	46.01	46.11	52.11	51.76	53.41	40.89	44.82	48.54	47.99	46.27	50.81	51.30	58.49	64.31	71.79	55.98	67.13	65.30	70.01
SUN397	43.02	41.19	47.44	41.16	49.05	39.96	52.10	41.89	43.72	49.38	44.42	48.70	50.85	50.42	53.18	53.73	54.85	57.29	56.21	55.28	58.02
Aircraft	1.74	1.23	1.38	1.83	1.68	1.77	2.01	2.01	1.11	1.38	2.01	1.32	1.38	1.56	1.95	3.45	2.07	2.70	1.83	3.81	2.55
Country211	49.97	50.08	47.94	49.92	50.08	50.02	50.10	50.08	50.12	49.92	49.92	48.60	49.92	47.01	49.07	4.62	5.80	50.00	50.03	50.07	50.14
DTD	16.44	17.07	20.16	20.16	20.05	19.63	22.71	19.61	21.82	24.47	24.04	26.01	23.51	25.90	27.61	27.02	32.39	30.77	31.60	28.57	33.30
GTSRB	10.26	10.15	9.25	12.72	12.04	18.31	12.91	9.95	12.19	14.51	12.91	13.26	12.68	11.80	14.66	11.68	14.06	14.50	12.76	17.14	15.64
STL10	90.55	92.04	91.63	91.15	92.91	92.00	94.65	87.11	88.70	92.24	90.75	93.00	94.03	93.23	96.86	95.56	96.50	96.68	96.80	95.69	97.90
Retino	6.33	30.37	3.19	30.89	2.49	12.74	2.57	4.17	8.03	7.54	5.02	30.23	24.20	44.95	2.50	6.83	3.63	9.06	5.00	59.02	5.00
EuroSAT	29.75	31.30	33.15	30.90	33.45	32.65	36.00	36.20	37.20	38.15	34.75	39.40	38.35	39.45	40.70	40.05	42.80	37.95	42.30	37.40	41.60
RESISC45	23.08	27.57	23.84	29.08	29.84	30.27	35.98	34.27	32.48	33.44	33.63	33.17	32.51	33.38	39.94	41.81	41.25	43.90	45.24	45.44	47.71
PCAM	40.67	49.33	55.64	57.44	49.51	54.24	48.41	53.11	54.30	55.80	54.30	55.21	65.11	49.66	57.50	54.25	51.99	54.97	50.81	48.88	55.60
CLEVR Counts	12.94	14.31	14.75	17.70	18.41	18.53	15.21	19.03	19.93	20.03	21.17	12.34	24.27	23.03	19.55	26.79	16.64	26.13	25.28	22.66	14.77
CLEVR Dist	15.87	24.43	21.47	22.61	24.14	24.81	11.43	23.10	24.05	15.75	15.93	15.83	20.15	15.80	14.93	22.67	25.38	9.42	20.77	9.98	25.79
DSPRITE Orient	1.84	2.97	2.52	3.32	2.42	2.76	3.20	2.51	2.83	2.47	1.96	2.89	1.90	3.03	1.13	2.48	2.40	2.59	2.46	2.65	2.74
DSPRITE Position	3.21	3.26	3.20	3.20	3.21	3.25	3.03	3.03	3.04	3.07	3.01	3.49	3.10	3.17	3.20	3.14	3.17	3.14	3.06	3.08	3.07
SmallNORB Elv	11.06	9.32	10.36	10.83	13.99	10.12	11.44	11.25	12.41	10.48	11.84	11.10	10.27	10.76	11.14	13.78	11.2	10.74	11.05	10.77	10.09
SmallNORB AZim	4.15	6.13	5.17	5.24	5.46	5.1	4.97	5.15	5.19	6.19	6.10	5.13	6.29	5.21	5.44	5.88	5.43	5.39	4.83	6.35	5.08
DMLAB	18.06	17.04	18.01	11.96	14.40	16.67	14.49	14.25	15.75	16.64	14.61	19.07	17.44	17.73	18.21	19.43	17.20	18.29	16.16	14.15	17.69
KITTI Dist	12.24	33.47	18.00	43.18	29.54	37.69	25.04	24.47	28.55	32.49	35.16	36.43	17.16	13.08	36.99	24.75	36.99	33.61	23.07	31.08	45.43
Avg-35	25.61	28.84	28.61	30.08	29.89	30.39	30.78	28.72	30.55	31.35	31.17	32.53	31.80	32.68	35.35	34.97	35.22	36.98	36.89	38.90	38.89

Table J. **Zero-shot Top-1 accuracy (%) on ImageNet-1K for model chains with different expansion ratios.**

Model Chain	Models						
	ViT-T/16	ViT-C/16	ViT-S/16	ViT-M/16	ViT-B/16	ViT-XB/16	ViT-L/16
baseline	26.16	28.54	30.16	31.08	31.80	32.44	33.77
2× Expansion	26.16	-	-	31.22	-	-	33.82
4× Expansion	26.16	-	30.24	-	31.83	-	34.27
8× Expansion	26.16	29.05	30.32	31.46	32.14	32.76	34.33

Table K. **Comparison of Top-1 accuracy (%) on ImageNet-1K**: baseline vs. CoM-PT under varying training epochs.

# Baseline Epochs	Method	Models			
		ViT-T/16	ViT-S/16	ViT-B/16	ViT-L/16
32	*baseline*	24.40	27.53	28.77	29.84
	CoM-PT	24.40	28.23	29.00	29.99
64	*baseline*	25.41	28.83	30.57	32.02
	CoM-PT	25.41	29.49	30.75	32.03
128	*baseline*	26.16	29.98	31.80	33.77
	CoM-PT	26.16	30.24	31.83	34.27

E. More Detailed Results

Model Performance on VTAB+. In Table I, we detail the performance of models on each dataset of VTAB+ in the main experiments. From the results, we can see that: i) CoM-PT achieves performance comparable to baseline across most datasets; ii) the average performance across all 35 datasets confirms that CoM-PT maintains lossless acceleration relative to baseline pre-training. Some datasets exhibit inherent instability (e.g., Retino) due to the large domain gaps between the pre-training data and these specialized datasets. However, this instability affects both methods similarly, and the averaged results across the full benchmark provide a robust evaluation that mitigates domain-specific variations.

Model Chains with Different Expansion Ratios. To validate our claim in Section 5.3 of the main paper that all models achieve lossless performance, Table J details ImageNet-1K results for model chains with varying expansion ratios. The results clearly demonstrate that every model in the chains maintains performance-lossless acceleration compared to the baseline.

Model Performance under Various Training Epochs. In Table K, we illustrate the performance of models pre-trained by baseline and our CoM-PT under varying training epochs. We can observe that our CoM-PT outperforms the baseline on each individual model. This confirms that the acceleration ratios reported in Figure 9(a) of the main paper are achieved in a performance-lossless manner.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, 2018. 1

[3] 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. 1

[4] Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. *arXiv preprint arXiv:1511.05641*, 2015. 4

[5] Mehdi Cherti and Romain Beaumont. Clip benchmark, 2025. 1

[6] 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. In *CVPR*, 2023. 1, 2

[7] 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. 1

[8] Ning Ding, Yehui Tang, Kai Han, Chao Xu, and Yunhe Wang. Network expansion for practical training acceleration. In *CVPR*, 2023. 4

[9] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *PASMCL*, 2011. 1

[10] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. 2023. 1

[11] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 2

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 3

[13] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 1

[14] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 2

[15] 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 *ECCV*, 2014. 1

[16] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2

[17] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 1, 2

[18] Yanqing Liu, Kai Wang, Wenqi Shao, Ping Luo, Yu Qiao, Mike Zheng Shou, Kaipeng Zhang, and Yang You. Mllms-augmented visual-language representation learning. *arXiv preprint arXiv:2311.18765*, 2023. 1

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[20] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022. 2

[21] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 1

[22] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 1

[23] Christoph Schuhmann, Robert Kaczmarczyk, Aran Komatsuzaki, Aarush Katta, Richard Vencu, Romain Beaumont, Jenia Jitsev, Theo Coombes, and Clayton Mullis. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *NeurIPSW*, 2021. 1

[24] 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. In *NeurIPS*, 2022. 1

[25] 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. 1

[26] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019. 2

[27] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. San: side adapter network for open-vocabulary semantic segmentation. *TPAMI*, 2023. 2

[28] Kecheng Zheng, Yifei Zhang, Wei Wu, Fan Lu, Shuailei Ma, Xin Jin, Wei Chen, and Yujun Shen. Dreamlip: Language-image pre-training with long captions. In *ECCV*, 2024. 1

[29] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS*, 2023. 3

[30] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 1