RILS: Masked Visual Reconstruction in Language Semantic Space

A. Wall-clock Time Comparison

Method	Dataset	PT Epo.	ZS.	FT.	Rel.GHs.
MAE [5]	LAION-20M	25	-	82.1	1.0×
CLIP [10]			40.3	82.7	1.5×
SLIP [9]			41.6	82.6	3.7×
MAE+CLIP			42.3	82.9	1.8×
RILS	LAION-20M	25	**<u>45.0</u>**	**<u>83.3</u>**	1.8×

Table S1. **Wall-clock time comparison.** We report zero-shot (ZS.) and end-to-end fine-tuning (FT.) accuracy on ImageNet-1K for reference. Rel.GHs. denotes relative GPU hours. Compared to CLIP, our method only brings 20% extra training time costs. Compared to MAE+CLIP, our method exhibits better performances under the same training overhead.

B. Data Scaling

Method	Dataset	PT Epo.	ZS.	Lin.	FT.
RILS	LAION-10M	25	37.5	68.5	82.7
	YFCC-15Mv2 [7]		41.5	70.2	82.9
	LAION-20M		45.0	71.5	83.3
	LAION-50M		49.4	71.9	83.6
	LAION-100M		50.6	72.2	83.7

Table S2. **Scaling property of our RILS.** All models are pre-trained with ViT-B/16 [2] as vision encoder for 25 epochs, and report zero-shot (ZS.), linear probing (Lin.) and fine-tuning (FT.) classification accuracy on ImageNet-1K. We observe contiguous gains when our approach meets more image-text pairs. Besides, we notice that increasing data from 50M to 100M shows relatively minor improvements, we speculate this is due to only scaling dataset instead of jointly scale-up dataset with model size. We leave this exploration in the future.

C. Implementation Details

C.1. Model Architecture Details

	Configuration	Value
Vision Encoder	Patch Size	16 × 16
	Layers	12
	Width	768
	Heads	12
	MLP Ratio	4.0
	# Parameters	85.8M
Vision Decoder	Layers	1
	Width	768
	Heads	12
	MLP Ratio	4.0
	# Parameters	8.4M
Language Encoder	Layers	12
	Width	512
	Heads	8
	MLP Ratio	4.0
	# Parameters	37.8M

Table S3. **Model architecture details.**

C.2. Pre-training

Configuration	Value
Batch Size	4096
Vocabulary Size	49408
Training Epochs	25
Optimizer	AdamW [6]
Learning Rate	5e-4
Minimal Learning Rate	1e-5
Weight Decay	0.5
Adam β_1	0.9
Adam β_2	0.98
Warmup Epochs	1
Learning Rate Schedule	Cosine
Augmentation	RandomResizedCrop(0.5, 1.0)
Mask Ratio	0.75

Table S4. **Pre-training settings.**

C.3. ImageNet-1K Fine-tuning

Configuration	Value
Batch Size	1024
Training Epochs	100
Optimizer	AdamW [6]
Learning Rate	4e-4
Weight Decay	0.05
Adam β_1	0.9
Adam β_2	0.999
Warmup Epochs	5
Learning Rate Schedule	Cosine
Layer-wise LR Decay	0.65
Augmentation	RandAug(9, 0.5) [1]
Label Smoothing	0.1
Mixup	0.8
CutMix	1.0
Drop Path	0.1

Table S5. **ImageNet-1K fine-tuning settings.**

C.4. Semantic Segmentation Fine-tuning

Configuration	Value
Batch Size	16
Training Iters	160K
Optimizer	AdamW [6]
Learning Rate	1e-4
Weight Decay	0.05
Adam β_1	0.9
Adam β_2	0.999
Warmup Iters	1500
Learning Rate Schedule	Poly
Layer-wise LR Decay	0.65
Image Size	512×512

Table S6. **ADE20K fine-tuning settings.**

C.5. Detection Fine-tuning

Configuration	COCO [8]	LVIS [4]
Batch Size	16	
Training Epochs	25	
Optimizer	AdamW [6]	
Learning Rate	1e-4	2e-4
Weight Decay	0.1	
Adam β_1	0.9	
Adam β_2	0.999	
Warmup Iters	250	
Learning Rate Schedule	Cosine	
Layer-wise LR Decay	0.7	
Drop Path	0.1	
Image Size	1024×1024	
Augmentation	LSJ(0.1, 2.0) [3]	

Table S7. **COCO and LVIS fine-tuning settings.**

D. Pre-training Pseudo Code

Algorithm 1 RILS pre-training pseudo-code in PyTorch style.

```
# xi, xt: input images and texts
# v_enc, v_dec: vision encoder, vision decoder
# l_enc: language encoder
# v_proj, l_proj: vision and language projector
# sigma, tau1, tau2: temperatures
# lambda_1, lambda_2: loss coefficients
# B, N, D: batch size, patch numbers, feature dimension

def forward(xi, xt):

    #random mask input images
    masked_xi = random_mask(xi, mask_ratio=0.75)

    zi = v_enc(xi) #[B, N, D]
    masked_zi = v_enc(masked_xi)
    gi = v_dec(masked_zi) #[B, N, D]
    zt = l_enc(xt) #[B, D]

    return forward_loss(zi, gi, zt)

def forward_loss(zi, gi, zt):

    #vision language contrastive
    ei = norm(v_proj(zi.mean(dim=1))) #[B, D]
    et = norm(l_proj(zt)) #[B, D]

    label = range(B)
    logit = ei @ et.T / sigma #[B, B]

    i2t = cross_entropy(logit, label)
    t2i = cross_entropy(logit.T, label)
    l_contra = (i2t + t2i) / 2.

    #masked visual reconstruction
    zi = norm(v_proj(zi)) #[B, N, D]
    gi = norm(v_proj(gi)) #[B, N, D]

    logit_p = (gi @ zt.T / tau1).softmax(-1) #[B, N, B]
    logit_t = (zi @ zt.T / tau2).softmax(-1) #[B, N, B]

    #reconstruction in language semantic space
    l_recon = kl_divergence(logit_p, logit_t)

    return lambda_1 * l_contra + lambda_2 * l_recon
```

References

[1] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 2020. 2

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[3] 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. 2

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

[5] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1, 2

[7] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021. 1

[8] 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. 2

[9] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pretraining. In *European Conference on Computer Vision*, 2022. 1

[10] 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