A. Supplementary Overview

In what follows we first provide more exhaustive details on our training and evaluation datasets in Section B, before then detailing our pre-training configuration in C. Section D shows the effect of masking CLIP input in the low-data regime. Sections E, F and G show more results that were used to compare MAE-CLIP and CLIP. In Section H we explain our VQA evaluation setup, and in Section I we cover zero-shot segmentation. Finally, Section J gives background on our choice of "similarity masking" (as described in the main paper) over random masking, and Section K provides updated results for our large-scale M3AE [31] run, as at the time of submission we were unable to provide results for a fully trained baseline.

B. Datasets

In this section we describe our pre-training dataset, and also provide background as to why there are two missing tasks in our VTAB [106] evaluation.

B.1. Pre-training Data

We made use of two internal datasets as well as several public datasets to build our "large-scale" pre-training dataset.

High Quality Image Text Pairs dataset: The High Quality Image Text Pairs (HQITP-134M) dataset consists of approximately 134M diverse and high quality images paired with descriptive captions and titles. Images range in spatial resolution from 320 to 2048 pixels on the short side. All images are JPEG format and most are RGB. Each example image is associated with a title, and a list of several captions. A small fraction ($\ll 1\%$) of the examples are missing both captions and title. We favor the associated captions, and find that these tokenize to an average length of 20.1 tokens, although the shortest caption is only one token and the longest is over $1,000$. This dataset was licensed to our industrial research lab by a third party for commercial use.

English Web Image Text dataset: The English-Web-Image-Text-2.2B (EWIT-2.2B) dataset consists of approximately 2.2B images paired with one or more related pieces of text. The data is the result of filtering English-language web-sourced data, using a combination of the filtering rules described in ALIGN [44] and CLIP [68]. Images range in spatial resolution from 200 to 5000 pixels on the short side, and 200 to 8650 on the long side, with a maximum aspect ratio of 3 and a mean of 1.385. Each image has an average of 1.341 pieces of text associated with it, although some have as many as 179. We find that the average associated text produces 15.5 tokens when tokenized.

Public datasets: As well as our internal datasets, we include Conceptual 12M (CC12M) [7], CC3M [76], and LAION-400M [74].

Overall pre-training dataset: Our overall training dataset is the result of combining HQITP-134M, CC12M, CC3M, and LAION-400M, before applying global image-byte-level de-duplication to drop image text pairs where either the image occurs more than once or the image occurs in one of our test sets. This results in a final training dataset of just over 1.4B image-text pairs.

B.2. VTAB Evaluation Data

In Sections 4 and 5 in the main paper we evaluate the quality of the learned visual representations by training a linear classifier on the predicted visual features of the VTAB datasets [105]. However, we do not include the Diabetic Retinopathy [46] dataset due to licensing concerns (the original dataset was provided solely for use in a Kaggle competition), and Sun397 [92], due to a missing image

at the time of preparing the datasets for the VTAB benchmark. The issue with Sun397 has since then been resolved, and it could be included in a future iteration of this work.

Component	Parameter	Value
Image Encoder	Depth	12
	Width	768
	MLP Heads	12
Text Encoder	Depth	12
	Width	512
	MLP Heads	8
Decoder	Depth	8
	Width	512
	MLP Heads	8
Model	Weight decay	0.1
	Base LR	5e-04
	LR Schedule	Cosine decay [59]
	LR Warmup steps	200
	Local contrastive steps [33]	500
	Batch size	256
	Optimizer	AdamW [60]
	Optimizer momentum	$\beta 1$, $\beta 2 = 0.9, 0.98$
	Augmentation	RandomResizedCrop

Table 8. Parameters used for pre-training

C. Training configuration

Table 8 shows the used hyperparameter setup for pre-training of MAE-CLIP and all our baselines. Following [68], we use an AdamW optimizer [48, 60], a linear learning rate warmup over 200 steps before then decaying to 0 with a cosine schedule [59] over the remainder of training. Using warmup steps, (where only a local contrastive loss is used instead of a global contrastive loss), helps the model to converge faster at the beginning of training. To train MAE-CLIP, we simply sum the contrastive and generative losses for the local contrastive phase of training, but multiply the generative image loss by 0.05 and the generative text loss by 0.1 when computing the global contrastive loss. This allows us to accommodate for the dramatically reduced gradient norm of a global contrastive loss with a large batch size, and was arrived at through hyperparameter search. For the image encoder, initial empirical experimentation showed that using a trainable or fixed position encoding does not influence results and we therefore use a fixed 2D position encoding. For both image and text encoders, we use pre-layer-norm [94] and the initialization scheme from [70].

We use the same training configuration for the *web-crawled* dataset, with a few changes due to the larger number of total steps. We use $10,000$ local contrastive loss steps, and $1,000$ warmup steps for the cosine learning rate scheduler.

Models	Zero-shot	Linear Probing
masked CLIP_{MAP}	23.0 *(29.7)*	48.0 *(52.6)*
masked CLIP_{GAP}	23.6 *(29.3)*	51.7 *(59.8)*
MAE-CLIP	**33.8**	**58.9**

Table 9. ImageNet classification with zero-shot transfer or linear probing after pretraining on the *CC* dataset. CLIP is trained on masked input, showing unmasked performance between brackets.

Model	COCO		FLICKR		COCOA	
	I→T	**T→I**	**I→T**	**T→I**	T1	T5
CLIP_{GAP}	51.9	36.6	78.8	62.3	24.2	46.9
CLIP_{MAX}	**55.3**	**39.0**	80.5	**65.3**	22.7	**51.6**
$\text{MAE-CLIP}_{\text{GAP}}$	53.0	37.0	77.3	62.0	20.7	39.5
$\text{MAE-CLIP}_{\text{MAX}}$	54.4	37.7	**81.2**	64.2	**24.6**	41.4

Table 10. Zero-shot (retrieval) accuracy (%). All trained on our *web-crawled* dataset (1.4B images). I→T: Image to Text, T→I: Text to Image.

D. Masking influence

In MAE-CLIP, we never mask the input for the contrastive task. However, one can argue that in the low-data regime, masking is a heavy data augmentation that improves performance (similar to how e.g. DINO [6] has great performance in the low-data regime, likely also because of its heavy data augmentations). We therefore also run an ablation where the input to a contrastive-only (normal CLIP) model is masked, similar to how input is masked for the generative task in MAE-CLIP.

E. Retrieval results

Table 10 shows zero-shot retrieval accuracy for three different datasets. We compare COCO [58], FLICKR [67] and COCOAmodal [83]. We chose COCOAmodal as a third retrieval evaluation set, as adding the masked auto-encoder might have provided MAE-CLIP a benefit over CLIP on occluded objects or non-object-centric datasets. We show that even on non-object-centric datasets, CLIP outperforms MAE-CLIP at scale.

F. VTAB Fine-tuning

As MAE is often used in a full-finetuning setting, we also show that fully-finetuning follows the linear-probing results (see Table 11). Adding masked autoencoding still does not outperform a contrastive-only baseline in a large-scale training. When training on CCxM only, we see MAE-CLIP outperforming CLIP on average.

	Caltech101 ●	CIFAR-100 ●	DTD ●	Flowers102 ●	Pets ●	SVHN ●	EuroSAT ●	Camelyon ●	Resisc45 ●	Clevr/Closest ●	Clevr/Count ●	DMLab ●	dSprites/Ori ●	dSprites/Loc ●	KITTI/Dist ●	sNORB/Azim ●	sNORB/Elev ●	Average
MAE-CLIP$_{\text{GAP}}$	92.6	84.2	74.4	95.1	85.7	80.2	97.0	88.0	92.9	87.1	84.3	71.0	96.4	100	46.8	99.0	95.0	86.5
MAE-CLIP$_{\text{MAX}}$	95.9	86.3	80.3	97.5	90.4	97.3	98.9	88.8	95.6	87.1	84.1	72.7	96.1	100	53.4	99.3	96.6	89.4
CLIP$_{\text{GAP}}$	96.0	87.7	81.8	98.1	90.7	97.4	98.8	85.7	95.9	87.8	80.0	72.5	96.5	100	51.0	98.7	93.6	88.9
CLIP$_{\text{MAX}}$	95.5	86.3	81.1	98.0	90.6	97.6	98.9	89.6	96.2	87.6	80.1	73.6	96.2	100	52.9	99.4	95.9	89.4
MAE-CLIP$_{\text{GAP}}$ *CCxM*	91.9	80.9	70.5	91.2	82.7	96.3	98.5	88.0	95.5	79.6	67.7	66.0	96.3	100	49.6	99.7	85.9	84.7
MAE-CLIP$_{\text{MAX}}$ *CCxM*	90.8	81.6	69.0	90.3	81.0	96.8	98.4	89.5	95.3	69.0	74.3	67.4	956	100	48.0	98.7	83.6	84.1
CLIP$_{\text{GAP}}$ *CCxM*	91.1	81.3	69.7	91.8	81.1	96.2	98.1	85.9	95.3	75.1	65.1	64.6	96.4	100	48.6	99.6	78.0	83.4
CLIP$_{\text{MAX}}$ *CCxM*	89.6	80.6	68.4	90.3	79.8	96.5	98.4	87.1	95.3	78.3	72.2	64.0	96.0	100	47.3	99.8	76.3	83.5

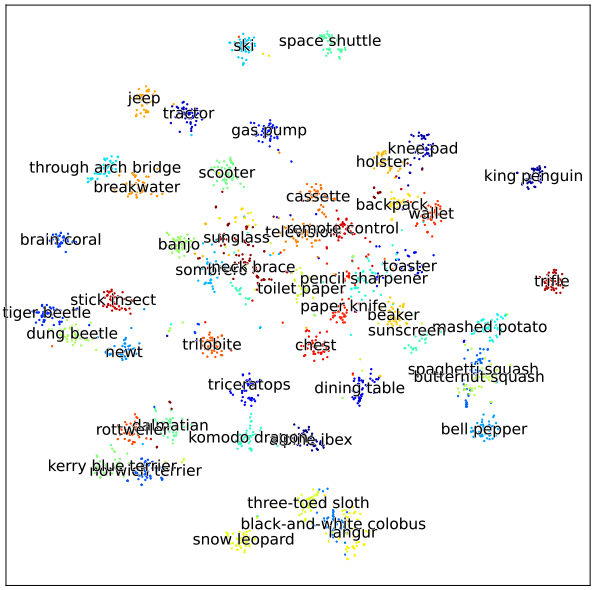
Table 11. Full-finetuning accuracy (%) on classification tasks. Models are all trained on our *web-crawled* dataset (1.4B images) or CCxM when specified.
● VTAB/natural, ● VTAB/specialized and ● VTAB/structured.

G. T-SNE analysis

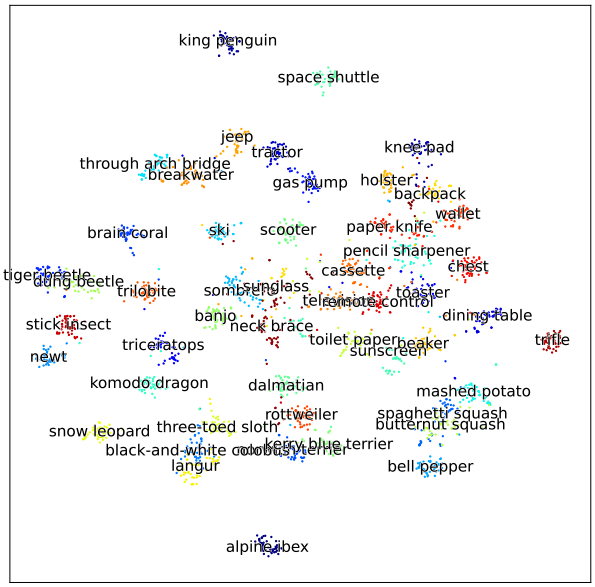
We also visually inspect the generated embeddings using t-SNE [85], see Figure 3. There is no clear difference between the embeddings of CLIP and MAE-CLIP, at scale (see Figure 3a and Figure 3b). For CLIP, MAE-CLIP and MAE trained on CCMxM, we can see that the MAE embeddings (see Figure 3e) look more cluttered than the CLIP and MAE-CLIP embeddings.

H. VQA Finetuning

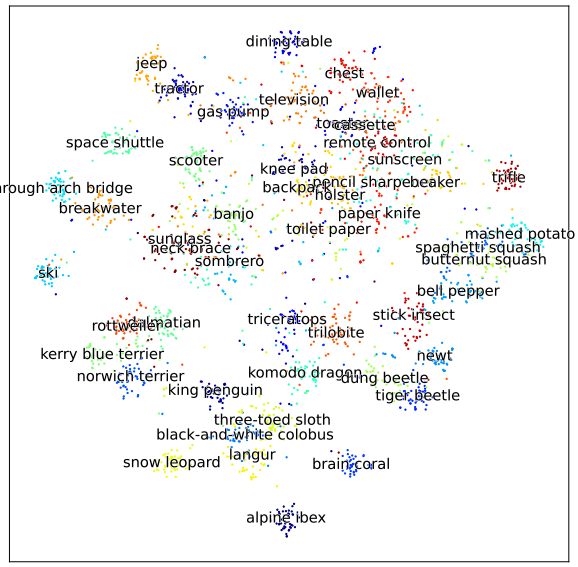
We evaluate on three VQA benchmark datasets CLEVR [45], VQAv2 [34] and GQA [42]. As mentioned in the main paper, we finetune our models for VQA, by freezing the image and texts encoders and adding a new decoder. We treat the problem as a classification problem by calculating the set of possible answers and treating each as a separate class. Following our pre-training setup, we concatenate the image and text embeddings, add positional encoding and a modality specific token before using it as input in the decoder. The `BOS` token is used as an output token and linearly projected to the possible classes. Figure 4 depicts our finetuning pipeline.



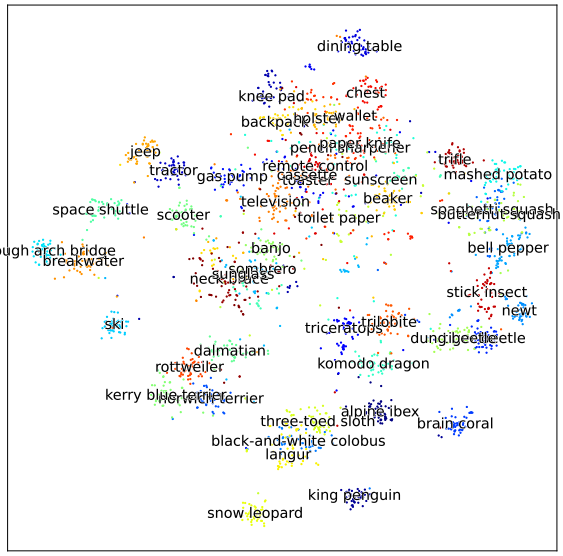
(a) CLIP



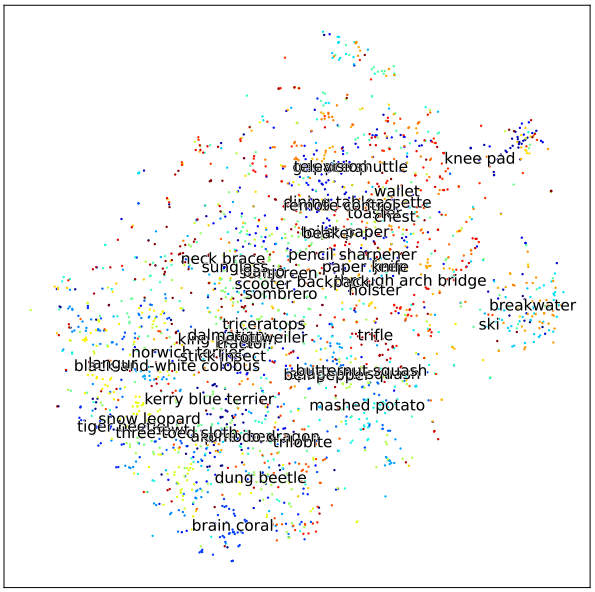
(b) MAE-CLIP



(c) CLIP *CCxM*



(d) MAE-CLIP *CCxM*



(e) MAE *CCxM*

Figure 3. T-sne visualizations for CLIP and MAE-CLIP, both large-scale training as CCXM.

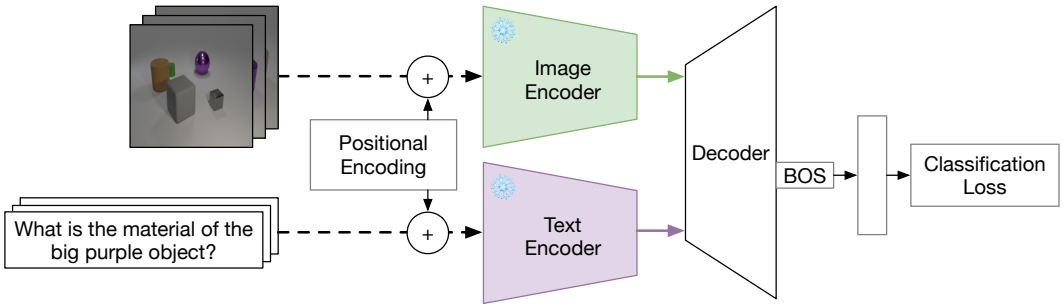


Figure 4. VQA Fine-tuning: decoder classification. Encoders are frozen, the BOS token of the decoder output is used as the classification token.

I. Zero-shot Segmentation

In Section 6 in the main paper, we present results on zero-shot semantic segmentation in order to evaluate the effects of self-supervision on visual grounding. Here, we describe our zero-shot semantic segmentation methodology and present a quantitative evaluation on three datasets. In particular, we use Pascal VOC [27], ADE20K [107] and COCO [58] with 20, 150 and 133 labels respectively. To compute segmentation masks, we first extract a feature per input pixel using bilinear interpolation from the per-patch features. Subsequently, we classify each pixel by computing the similarity of the feature to the embedding of the prompt for each class.

Model	Pooling	COCO	ADE20K	Pascal VOC
CLIP	MAP	8.4	4.6	19.1
MAE-CLIP	MAP	9.1	5.7	20.6
CLIP	GAP	7.6	4.1	19.4
MAE-CLIP	GAP	8.5	5.5	20.4
CLIP	MAX	16.5	9.4	36.6
MAE-CLIP	MAX	17.8	11.1	36.9

Table 12. Zero-shot semantic segmentation results for CLIP and MAE-CLIP after training on the CC dataset (11.3M images). MAE-CLIP consistently improves upon CLIP for semantic segmentation regardless of the pooling strategy, as also seen qualitatively in Figure 2 in the main paper.

Table 12 evaluates the performance of CLIP and MAE-CLIP trained on the CC dataset with respect to mean intersection over union for all three datasets. We observe that self-supervision consistently improves the performance of CLIP for zero-shot semantic segmentation. However, as mentioned in Section 6 in the main paper, the choice of pooling operator has a much larger effect.

I.1. Prompts

In this section, we provide the prompts used for our zero-shot semantic segmentation experiments. Each prompt is made into a sentence by prepending "a photo of a" or "an" depending on whether the label starts with a vowel. For COCO [58] we simply use the 133 label names as they are provided by https://github.com/cocodataset/panopticapi. For Pascal VOC [27] we use

```
aeroplane, bicycle, bird, boat,
bottle, bus, car, cat, chair,
cow, table, dog, horse, motorbike,
person, potted plant, sheep, sofa,
train, television monitor
```

for classes 1 to 20 and

```
background, bag, bed, bench, book,
building, cabinet, ceiling, cloth,
computer, cup, door, fence, floor,
flower, food, grass, ground,
keyboard, light, mountain, mouse,
curtain, platform, sign, plate,
road, rock, shelves, sidewalk,
sky, snow, bedclothes, track, tree,
truck, wall, water, window, wood
```

for the 0-th class. Namely, if a pixel is classified as any of the latter categories it is considered to be a background pixel. Finally, for ADE20K [107] we associate several prompots for each of the 150 categories. Subsequently, we compute the similarity of each pixel with each of the prompts and select the maximum similarity per category from all the associated prompts. The per-category prompts are as follows:

1. wall, walls, brick wall, stone wall, interior wall
2. building, buildings, edifice, edifices
3. sky, clouds
4. floor, flooring
5. tree, trees
6. ceiling
7. road, route, street, roads, streets, routes
8. bed, beds
9. windowpane, window, windows
10. grass, grass field
11. cabinet, cabinets, wall mounted cabine
12. sidewalk, pavement
13. person, child, girl, boy, woman, man, people, children, girls, boys, women, men
14. earth, ground
15. door, double door, doors
16. table, tables, tablecloth
17. mountain, mount, mountains
18. plant, flora, plant life, plants, bushes
19. curtain, drape, drapery, mantle, pall
20. chair, chairs
21. car, automobile, cars
22. water
23. painting, picture, paintings, pictures, wallart, framed canvas
24. sofa, couch, sofas, couches
25. shelf, shelves
26. house exterior
27. sea, ocean
28. mirror, mirrors
29. rug, carpet, carpeting
30. field
31. armchair, armchairs
32. seat, seats
33. fence, fencing
34. desk, desks
35. rock, stone, rocks, stones
36. wardrobe, closet, press, wardrobes, closets
37. lamp, lamps
38. bathtub, bathing tub, bath, tub
39. railing, rail
40. cushion, cushions
41. pedestal
42. box, boxes
43. column, pillar
44. signboard, sign, signboards, signs
45. chest of drawers, chest, bureau, dresser
46. counter
47. sand
48. sink
49. skyscraper, skyscrapers
50. fireplace, hearth, open fireplace
51. refrigerator, icebox
52. grandstand, covered stand
53. path
54. stairs, steps
55. runway
56. case, display case, showcase, vitrine
57. pool table, billiard table, snooker table
58. pillow, pillows
59. screen door, shower door
60. stairway, staircase
61. river
62. bridge, span
63. bookcase
64. window screen, door screen
65. coffee table, cocktail table
66. toilet, commode, crapper, potty
67. flower, flowers
68. book, books
69. hill
70. bench, benches
71. countertop, counter top, worktop
72. stove, kitchen stove, kitchen range, kitchen range, cooking stove
73. palm tree, palm trees
74. kitchen island
75. computer, computing machine, computing device, data processor, electronic computer, information processing system
76. swivel chair
77. boat
78. bar
79. arcade machine, arcade machines
80. hovel, hut, hutch, shack, shanty
81. bus, autobus, double-decker, jitney, motorbus, motorcoach, omnibus, passenger vehicle
82. towel
83. light bulb, lightbulb, bulb, incandescent lamp, electric light, electric-light bulb
84. truck, motortruck
85. tower, towers
86. chandelier, pendant, pendent
87. awning, sunshade, sunblind
88. streetlight, street lamp
89. booth, cubicle, stall, kiosk
90. television receiver, television, television set, tv, tv set
91. airplane, aeroplane, airplanes, aeroplanes
92. dirt track
93. apparel, wearing apparel, dress, clothes
94. pole
95. land, soil
96. bannister, banister, balustrade, balusters, handrail
97. escalator, moving staircase, moving stairway
98. ottoman, pouf, pouffe, puff, hassock
99. bottle, bottles, water bottle
100. buffet, sideboard
101. poster, posting, placard, notice, bill, card
102. stage
103. van
104. ship
105. fountain
106. conveyer belt, conveyor belt, conveyer, conveyor, transporter
107. canopy
108. washer, automatic washer, washing machine
109. plaything, toy, toys
110. swimming pool, swimming bath
111. stool, stools
112. barrel, cask, barrels, casks
113. basket, handbasket
114. waterfall, falls
115. tent, collapsible shelter
116. bag, bags, gift bag, paper bag
117. minibike, motorbike
118. cradle
119. oven
120. ball, balls
121. food, solid food
122. step, stair
123. tank, storage tank
124. trade name, brand name, brand, marque
125. microwave, microwave oven
126. plant pots, plant pot, flower pot, flowerpot, planter
127. animal, animate being, dog, cat, horse, cow, sheep, zebra, girraffe, bird
128. bicycle, bike
129. lake
130. dishwasher, dish washer, dishwashing machine
131. projection screen
132. blanket, cover
133. sculpture, sculptures
134. exhaust hood
135. sconce, sconce lamp, sconce light
136. vase, vases
137. traffic light, traffic signal, traffic lights
138. tray, trays
139. ashcan, trash can, garbage can, wastebin, ash bin, ash-bin, ashbin, dustbin, trash barrel, trash bin
140. ceiling fan, floor fan
141. pier, wharf, wharfage, dock
142. crt screen
143. plate, plates
144. monitor, monitoring device, monitors
145. bulletin board, notice board
146. shower
147. radiator
148. cup, cups, drinking glass, drinking glasses
149. clock
150. flag, flags

J. MAE-CLIP masking strategy

In Table 13 we compare random masking to similarity masking for MAE-CLIP$_{\text{MAX}}$, training on the CC dataset. Our experiments show that both strategies perform very similarly in all cases with random masking showing a small improvement for classification tasks while similarity masking an improvement on VQA and semantic segmentation, namely tasks that benefit from better visual grounding. All MAE-CLIP experiments in the main paper employ similarity masking. Further experiments are needed to properly evaluate the effect of the masking strategy across different scales and pooling methods.

K. M3AE results

This section provides an updated version of the "large-scale" tables of results from Section 5 in the main paper. They are presented in Tables 14, 15a and 15b. We include these because our M3AE [31] baseline had not fully converged at time of submission. All numbers except those associated with M3AE are identical to the ones presented in the main paper.

Firstly, we report the full linear probing evaluation on the VTAB tasks (Table 14). We observe that M3AE performs on par with MAE-CLIP$_{\text{GAP}}$. Moreover, we note that M3AE performs consistently worse for all VTAB natural tasks while outperforming MAE-CLIP on VTAB strucuted tasks. This trend also continues with the rest of the results, where M3AE performs measurably worse on ImageNet linear-classification (Table 15a) while siginficantly better on CLEVR VQA (Table 15b); the former being a very natural task and the latter being very structured.

Model	Masking	$\text{VQA}_{\text{Avg.}}$	$\text{SemSeg}_{\text{Avg.}}$	VTAB_{Avg}	IN1K_{LP}	IN_{ZS}
$\text{MAE-CLIP}_{\text{MAX}}$	similarity	**68.53**	**21.95**	69.14	63.16	35.2
$\text{MAE-CLIP}_{\text{MAX}}$	random	68.44	21.27	**69.92**	**63.46**	**35.4**

Table 13. Similarity vs random masking for $\text{MAE-CLIP}_{\text{MAX}}$ trained on the CC dataset. We show average VQA, zero-shot semantic segmentation (SemSeg), and VTAB results, as well as ImageNet1K (IN) Top-1 linear probe and zero-shot scores as measured on the validation set.

Table 14. Linear probing accuracy (%) on classification tasks. Models are all trained on our *web-crawled* dataset (1.4B images). (● VTAB/natural, ● VTAB/specialized and ● VTAB/structured.) In the large scale pretraining regime, the difference between MAE-CLIP and CLIP is reduced to $< 1\%$. This table provides the results for a fully trained M3AE compared to Table 5 in the main paper where M3AE had completed 50% of the training steps.

	Caltech101 ●	CIFAR-100 ●	DTD ●	Flowers102 ●	Pets ●	SVHN ●	EuroSAT ●	Camelyon ●	Resisc45 ●	Clevr/Closest ●	Clevr/Count ●	DMLab ●	dSprites/Ori ●	dSprites/Loc ●	KITTI/Dist ●	sNORB/Azim ●	sNORB/Elev ●	Average
M3AE	94.0	75.6	78.8	96.1	84.6	67.8	97.2	85.3	92.4	65.4	75.9	52.0	57.4	79.2	51.1	40.6	68.3	74.2
CLIP	94.9	78.4	80.0	97.3	86.9	59.0	94.1	82.3	92.7	45.6	62.1	46.0	46.1	53.3	50.9	20.3	35.8	66.2
CLIP_{MAX}	96.1	81.0	80.9	97.3	89.9	65.7	96.0	83.2	94.1	52.8	67.8	49.9	59.5	67.6	41.2	23.4	45.8	70.1
$\text{MAE-CLIP}_{\text{MAX}}$	95.8	79.2	81.5	96.8	88.2	62.1	95.8	81.8	93.0	52.0	66.9	49.6	53.7	72.5	53.0	32.3	45.4	70.6
CLIP_{GAP}	95.8	80.5	81.6	97.6	88.7	66.0	97.0	84.4	93.3	56.7	71.4	53.3	58.0	70.1	50.6	38.3	55.1	72.9
$\text{MAE-CLIP}_{\text{GAP}}$	95.4	79.3	82.2	97.4	88.6	72.8	96.6	84.5	93.5	57.5	73.6	52.7	57.5	71.2	51.6	45.6	55.2	73.8

Models	Zero-shot	Linear Probing
M3AE	–	71.5
CLIP_{GAP}	61.8	75.9
CLIP_{MAX}	**63.7**	**77.5**
$\text{MAE-CLIP}_{\text{GAP}}$	57.4	75.7
$\text{MAE-CLIP}_{\text{MAX}}$	60.9	76.6

(a) ImageNet classification

Model	CLEVR	VQAv2	GQA
M3AE	**97.2**	61.1	54.6
CLIP_{GAP}	87.8	61.8	55.0
CLIP_{MAX}	89.5	60.6	53.6
$\text{MAE-CLIP}_{\text{GAP}}$	92.8	**61.9**	**55.3**
$\text{MAE-CLIP}_{\text{MAX}}$	93.9	61.5	53.7

(b) VQA finetuning results

Table 15. ImageNet classification and VQA results after pretraining on *web-crawled* dataset (1.4B images). In the large scale regime, self-supervision does not complement natural language supervision and all methods perform similarly on both tasks. This table provides updated results for M3AE (after it was fully trained) compared to Tables 6 and 7 in the main paper.