Learning Visual Representations with Caption Annotations *Supplementary Material*

Mert Bulent Sariyildiz, Julien Perez, Diane Larlus

NAVER LABS Europe

In this document, we provide the following:

(i) An ablation study on label sets *vs.* target task performances (Sec. A).

(ii) An ablation study on possible design choices in ICMLM$_\star$ architectures *vs.* target task performances (Sec. B).

(iii) Evaluations of the models trained in Sec. 4.2 of the main paper, on zero-shot image classification tasks (Sec. C).

(iv) Qualitative results obtained by our ICMLM$_{\texttt{att-fc}}$ model with VGG16 backbone (Sec. D).

(v) The details of the transformer encoder we employ in our ICMLM$_{\texttt{tfm}}$ models (Sec. E).

(vi) Implementation details including both training and evaluation of the models we compare in this paper (Sec. F).

A Label sets *vs.* target task performances

As we mention in the main paper, for TP$_{\text{Postag}}$ and ICMLM$_\star$ models, we can construct multiple concept (or label) sets from captions, *e.g.* the most frequent K nouns, adjectives or verbs in captions can be used as tags for TP$_{\text{Postag}}$ and as maskable tokens for ICMLM$_\star$ models. In this section, we investigate the impact of learning from such label sets on target task performances. To do so, we compare learning visual representations using annotated labels of images *vs.* tags derived from captions, *i.e.* TP$_{\text{Label}}$ *vs.* TP$_{\text{Postag}}$ with various label sets.

For this analysis, we first train ResNet50 backbones, and then, once a model is trained, we extract image representations from the frozen backbones. To test generalization capabilities of the representations, we train linear SVMs on VOC [4] and linear logistic regression classifiers on IN-1K [14]. Additionally, to understand how effectively models can learn from the training set, we also train linear SVMs on COCO [11].

Results are presented in Tab. 1. All TP$_{\text{Postag}}$ models trained for the ablation improve over TP$_{\text{Label}}$, suggesting that a caption describing an image can provide more comprehensive supervision compared to labeling it with a small set of classes. It is surprising that gaps are more significant on IN-1K, indicating that a large vocabulary of tags allows backbones to encode more discriminative patterns. TP$_{\text{Postag}}$ obtained by using the most frequent 5K nouns, adjectives and verbs in captions improves TP$_{\text{Label}}$ by **2.4**%, **9.9**% and **2.0**% on VOC, IN-1K

Table 1. Label sets *vs.* target task performances of $TP_\star$ models trained on COCO using ResNet-50 backbones. We report mAP (and top-1) scores obtained with linear SVMs on VOC and COCO (and logistic regression classifiers on IN-1K). NN, ADJ, VB denote that nouns, adjectives and verbs are present in a label set. In parantheses are the number of concepts (*e.g.* classes) in the label sets. Blue numbers are not transfer tasks.

Label Set	VOC	IN-1K	COCO	Label Set	VOC
GT Labels (TP_{Label}, 80)	80.2	34.0	73.5	NN + ADJ + VB (1K)	81.4
NN (5K)	81.8	43.9	75.3	NN + ADJ + VB (2.5K)	82.1
NN + ADJ (5K)	82.3	**44.5**	**75.5**	NN + ADJ + VB (5K)	**82.6**
NN + ADJ + VB (5K)	**82.6**	43.9	**75.5**	NN + ADJ + VB (10K)	81.9

Table 2. ICMLM *vs.* target task performances. We train $ICMLM_\star$ models with different numbers of hidden layers (#L) and attention heads (#H) on COCO using ResNet-50 backbones and compare them on **proxy** and **target** tasks. While training $ICMLM_\star$ models we set $\lambda = 0$ in Eq. 8 of the main paper. For the **proxy** task, we report top-1 MTP scores on COCO; for the **target** tasks see the caption of Tab. 1. $BERT_{base}$ alone achieves 25.7% on the **proxy** task. Blue numbers are not transfer tasks.

		$ICMLM_{\texttt{tfm}}$				$ICMLM_{\texttt{att-fc}}$			
#L	#H	Proxy	VOC	IN-1K	COCO	Proxy	VOC	IN-1K	COCO
1	1	65.2	85.7	50.6	**77.6**	58.5	**86.8**	47.2	**78.9**
1	4	66.1	85.3	50.7	77.5	59.4	86.7	46.8	**78.9**
1	12	66.5	85.5	50.4	77.2	59.5	86.6	47.3	**78.9**
2	1	66.7	85.0	46.6	76.2	59.5	86.4	48.1	78.8
2	4	67.1	85.0	46.7	76.3	60.2	86.3	48.5	78.6
2	12	**67.5**	84.8	46.6	76.1	**60.4**	86.3	**48.7**	78.5

and COCO. In Sec. 4.3 of the main paper, we report results of TP_{Postag} and $ICMLM_\star$ models trained with this label set.

B ICMLM *vs.* target task performances

This section extends the analysis reported in Sec. 4.1 of the main paper, *i.e.* we study how the masked language modeling (MLM) performance (the proxy task) translates to target tasks. This time we use ResNet50 backbones instead of VGG16 (as in Sec. 4.1 of the main paper). To do so, we train $ICMLM_{\texttt{tfm}}$ (and $ICMLM_{\texttt{att-fc}}$) models with different numbers of hidden layers and attention heads in `tfm` (and, in `fc` and `att`, respectively) modules, and monitor both proxy and target task results. While training $ICMLM_\star$ models, we set $\lambda = 0$ in Eq. 8 of the main paper: for this ablation study the training solely depends on ℓ_{mlm} defined by Eq. 7 in the main paper. Similar to the previous analysis, we perform target tasks using pre-extracted image features on VOC, IN-1K and COCO. We also report top-1 masked token prediction (MTP) scores on COCO.

Results are reported in Tab. 2. We observe that having more hidden layers or attention heads improve the MLM performance at the expense of reduced target task results. We believe that as the complexity of `tfm`, `att` or `fc` modules increase, they can learn more interconnections between visual and textual cues, and this, in turn, lifts some of the burden of capturing the semantics of the caption off the visual model itself, and leads ϕ to learn weaker visual features. Moreover, similar to the observations we made in Secs. 4.1 and 4.2 of the main paper, ICMLM$_{\texttt{tfm}}$ significantly outperforms ICMLM$_{\texttt{att-fc}}$ on MLM and IN-1K, however, ICMLM$_{\texttt{att-fc}}$ is slightly better than ICMLM$_{\texttt{tfm}}$ on VOC and COCO. The fact that IN-1K performance of ICMLM$_{\texttt{att-fc}}$ increases when `fc` module has two hidden layers also supports the hypothesis that ICMLM$_{\texttt{att-fc}}$ tends to overfit to the concepts present in the training set (hence it performs better on VOC and COCO).

Comparing Tab. 1 and Tab. 2, we see overall that ICMLM$_{\star}$ (when #L and #H are 1) improves TP$_{\text{Postag}}$ by at least **3.1**%, **3.3**%, **2.1**% and TP$_{\text{Label}}$ by at least **5.5**%, **13.2**%, **4.1**% on VOC, IN-1K and COCO.

A note for ICMLM$_{\star}$ models with a VGG16 backbone. We tried these settings for VGG16 backbones: one attention head in ICMLM$_{\star}$ models and $\lambda = 0$. (Eq. 8 of the main paper) but this lead to inferior models. We believe that this is due to the absence of residual connections in the backbone architecture, which leads to overfitting to MLM tasks (a similar behavior is observed in [8] for self-supervised learning methods trained with VGG16 architecture).

Importance of λ in Eq. 8 of the main paper. We discuss in Sec. 3 of the main paper that global *vs.* localized semantics in images can and should be captured separately. To this end, in Eq. 8 of the main paper, we propose to optimize a combination of ℓ_{tp} and ℓ_{mlm} losses to effectively train backbones by providing supervision for both global and localized semantics. In our ICMLM$_{\star}$ experiments, we validated the coefficient λ combining these loss terms by monitoring the ℓ_{tp} loss on the validation sets of COCO or VG. We tried three values for $\lambda \in \{0.0, 0.1, 1.0\}$ and found that $\lambda = 0.1$ and $\lambda = 1.0$ minimize ℓ_{tp} loss on the validation sets with ResNet-50 and VGG16 backbones respectively, and moreover improve target task results. This finding supports our claim that ℓ_{tp} and ℓ_{mlm} loss terms are complementary.

C Zero-shot Object Classification

We also extend the analysis in Sec. 4.2 of the main paper on an additional target task, zero-shot image classification, on CUB-200-2011 (CUB) [18] and Attributes with Animals 2 (AWA2) datasets [19]. The CUB dataset contains roughly 12K images for 200 types of *fine-grained* bird species defined by 312 different semantic *attributes*. The AWA2 dataset has roughly 38K images for 50 *coarse-grained* animals defined by 85 different attributes. The classes in these datasets are split into two subsets called *seen* and *unseen* classes. The goal of these benchmarks is to train a classification model on seen classes in a way that the classification model can effectively be used for both seen and unseen classes.

Using the recently proposed splits [19], we have 150 (resp. 40) and 50 (resp. 10) classes in the seen and the unseen splits for CUB (and AWA) datasets. Image samples from seen classes are divided into training and test sets whereas image samples from unseen classes are solely used for testing purposes.

In this analysis, we take the VGG16 backbones trained by $\text{TP}_\star$ or $\text{ICMLM}_\star$ models on the MS-COCO [11] (COCO) or Visual Genome [10] (VG) datasets. Similar to what we report in Sec. 4.2 from the paper, using the activations from the last three convolutional layers, we train bilinear score functions [15] that measure the *compatibility* between the visual features $\mathbf{x} \in \mathbb{R}^m$ (pooled and flattened to have roughly 9K dimension) and class-level attribute vectors $\mathbf{a} \in \mathbb{R}^n$ (n is 312 for CUB and 85 for AWA). Concretely, we define the score function as

$$f(\mathbf{x}, \mathbf{a}) = \mathbf{a}^\top (\boldsymbol{\Sigma}\mathbf{x} + \mathbf{b}) \tag{1}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times m}$ and $\mathbf{b} \in \mathbb{R}^n$ are parameters of the score function to be learned. Using the score function, class predictions are simply made by:

$$\hat{y} = \underset{c \in \mathcal{C}}{\arg\max} \; f(\mathbf{x}, \mathcal{A}_c), \tag{2}$$

where $\mathcal{A}_c \in \mathbb{R}^n$ denotes the class-level attribute vector for class c and $\mathcal{C}$ is the set of all classes. We train the score function by minimizing the following:

$$\boldsymbol{\Sigma}^\star, \mathbf{b}^\star = \underset{\boldsymbol{\Sigma}, \mathbf{b}}{\arg\min} \; - \underset{(\mathbf{x}, y) \in \mathcal{D}}{\mathbb{E}} \left[\log \left(p \left(y | \mathbf{x}, \mathcal{A} \right) \right) \right], \tag{3}$$

where $\mathcal{D}$ is a dataset of feature-label pairs $(\mathbf{x}, y)$ s.t. $y \in \{1, \dots, \mathcal{C}\}$ and

$$p \left(y = c | \mathbf{x}, \mathcal{A} \right) = \frac{\exp(f(\mathbf{x}, \mathcal{A}_c))}{\sum_j \exp(f(\mathbf{x}, \mathcal{A}_j))}. \tag{4}$$

Results. Tab. 3 reports top-1 prediction accuracies among all classes for both datasets. We make the following observations.

(i) We see that $\text{ICMLM}_{\texttt{tfm}}$ model trained on VG significantly improves $\text{TP}_\star$ models on CUB, *i.e.* up to **1.4%**, **1.3%** and **2.2%** with C-11, C-12 and C-13 features. On the other hand, $\text{ICMLM}_{\texttt{tfm}}$ model trained on COCO improves $\text{TP}_\star$ models on AWA2 up to **1.1%**, **0.9%** and **1.0%** with C-11, C-12 and C-13 features. In fact, $\text{ICMLM}_\star$ models tend to perform slightly better on AWA2 (particularly with C-13 evaluations), when they are pretrained on COCO indicating that the concepts in COCO are semantically more similar to the concepts in AWA2.

(ii) When trained on VG, the C-13 features learned by $\text{ICMLM}_{\texttt{att-fc}}$ are inferior to $\text{TP}_{\text{Postag}}$, *i.e.* the scores drop up to 0.9%. This implies that the VGG16 backbone trained by $\text{ICMLM}_{\texttt{att-fc}}$ slightly overfits to MLM task. However, the opposite is true for the C-11 and C-12 features, suggesting that the network is able to extract richer semantics from the earlier layers.

Table 3. Zero-shot object classification with VGG16 backbones. We report top-1 accuracies over all classes (seen + unseen) on CUB and AWA2 datasets. Those are obtained by training a bilinear function between the visual features produced by each of the methods and the class-level attribute vectors. We report the mean of 5 runs with different seeds (std. ≤ 0.3 for all settings). #I: The number of images in the training set. [9] shows that transfer learning performance is correlated with the overlap of classes between IN-1K and target task datasets. The fact that IN-1K contains 59 bird-related classes and the majority of the classes in AWA2 dataset provides ImageNet pretrained models an unfair advantage. Therefore, we distinguish them with blue numbers.

	Proxy tasks			**CUB**			**AWA2**		
Method	**Dataset**	**Supervision**	**#I**	C-11	C-12	C-13	C-11	C-12	C-13
ImageNet	IN-1K	1K classes	1.3M	10.2	**19.4**	**24.4**	11.4	**37.1**	**38.9**
$\mathcal{S}$-ImageNet	IN-1K	1K classes	100K	11.6	16.1	18.3	12.7	33.2	34.9
$\mathcal{S}$-ImageNet	IN-1K	100 classes	100K	**12.5**	14.1	15.7	**13.1**	32.0	33.3
TP_{Label}	COCO	80 classes	118K	11.1	11.7	11.5	31.1	32.0	32.8
$\text{TP}_{\text{Cluster}}$ *(Ours)*	VG	1K clusters	103K	9.8	10.3	10.3	30.3	30.8	30.6
$\text{TP}_{\text{Cluster}}$ *(Ours)*	VG	10K clusters	103K	10.3	10.7	10.4	30.9	31.6	31.9
$\text{TP}_{\text{Postag}}$ *(Ours)*	VG	1K tokens	103K	10.6	11.1	11.5	30.8	31.7	32.3
$\text{TP}_{\text{Postag}}$ *(Ours)*	VG	10K tokens	103K	10.4	10.9	11.3	31.0	31.9	32.4
$\text{ICMLM}_{\texttt{tfm}}$ *(Ours)*	VG	sentences	103K	**12.5**	**13.0**	**13.7**	**32.2**	32.8	33.1
$\text{ICMLM}_{\texttt{att-fc}}$ *(Ours)*	VG	sentences	103K	12.1	12.0	10.9	31.5	32.1	31.5
$\text{ICMLM}_{\texttt{tfm}}$ *(Ours)*	COCO	sentences	118K	12.4	12.8	13.3	**32.2**	**32.9**	**33.8**
$\text{ICMLM}_{\texttt{att-fc}}$ *(Ours)*	COCO	sentences	118K	11.9	12.3	12.4	31.8	32.7	33.1

D Additional qualitative results

In Figs. 1 and 3 of the main paper, we show attention maps produced by our $\text{ICMLM}_{\texttt{tfm}}$ model (`tfm` module contains 1 hidden layer and 1 attention head) with ResNet-50 backbone trained on COCO. This section provides additional attention maps obtained by the `att` module in our $\text{ICMLM}_{\texttt{att-fc}}$ model (`fc` and `att` modules contain 1 hidden layer and 12 attention heads, respectively) with VGG16 backbone trained on COCO. These maps are shown in Figs 1 and 2.

First, we see from the figures that the `att` module can successfully localize object categories that have a clear visual appearance. This is the case for instance of the banana, the baby, the cats, or the sheep from Fig. 1. This is also the case even in cluttered scenes, such as the bed on the second row of Fig. 1.

Second, it is interesting to see that even visual concepts that are more abstract than object categories can also be localized, such as the *mirror* or *glass*. In the particular case of the *glass* category, the versatility of this concept is successfully captured by our model, covering the drinking glass and the material of the table and of the vase.

Third, the model goes beyond nouns and learns the visual appearance associated to colors or textures. For instance, the concepts *blue*, *striped* or *colorful* are illustrated in Fig. 2.

Finally, we show some failure cases. This is often the case for ambiguous concepts whose visual appearance is not properly defined, such as *middle* and

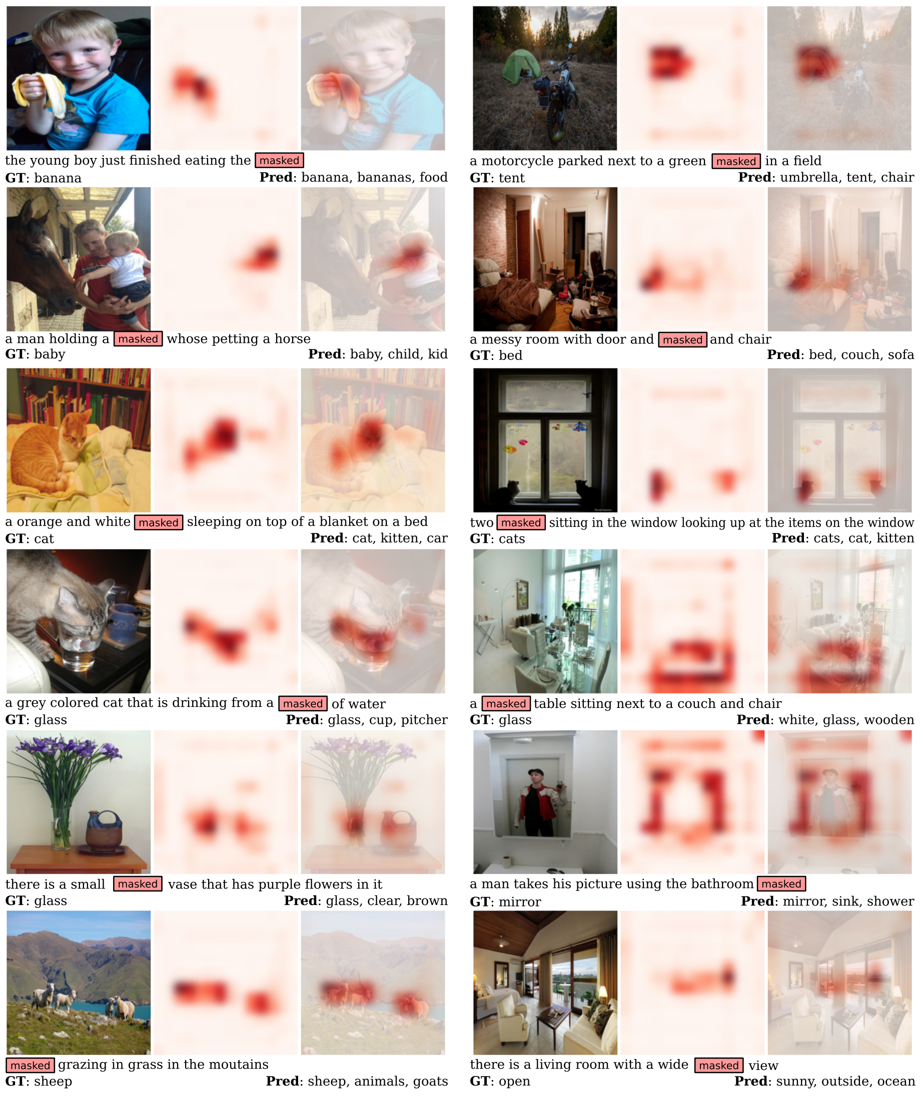
the young boy just finished eating the masked
GT: banana
Pred: banana, bananas, food
a motorcycle parked next to a green masked in a field
GT: tent
Pred: umbrella, tent, chair
a man holding a masked whose petting a horse
GT: baby
Pred: baby, child, kid
a messy room with door and masked and chair
GT: bed
Pred: bed, couch, sofa
a orange and white masked sleeping on top of a blanket on a bed
GT: cat
Pred: cat, kitten, car
two masked sitting in the window looking up at the items on the window
GT: cats
Pred: cats, cat, kitten
a grey colored cat that is drinking from a masked of water
GT: glass
Pred: glass, cup, pitcher
a masked table sitting next to a couch and chair
GT: glass
Pred: white, glass, wooden
there is a small masked vase that has purple flowers in it
GT: glass
Pred: glass, clear, brown
a man takes his picture using the bathroom masked
GT: mirror
Pred: mirror, sink, shower
masked grazing in grass in the moutains
GT: sheep
Pred: sheep, animals, goats
there is a living room with a wide masked view
GT: open
Pred: sunny, outside, ocean

Fig. 1. Qualitative results. For several image-caption pairs of the validation set of the COCO dataset and for a masked token, we show the ground-truth label (GT) together with the top 3 predictions (Pred) and the attention map generated by our ICMLM$_{\texttt{att-fc}}$ model with VGG16 backbone. The red parts correspond to higher attentions.

open which are respectively illustrated in the bottom right of Fig. 1 and Fig. 2. In some extreme cases, the attention maps are meaningless, and the masked word prediction relies on the rest of the caption instead. An other failure case is the bottom left of Fig. 2 which shows that grouping several concepts (like

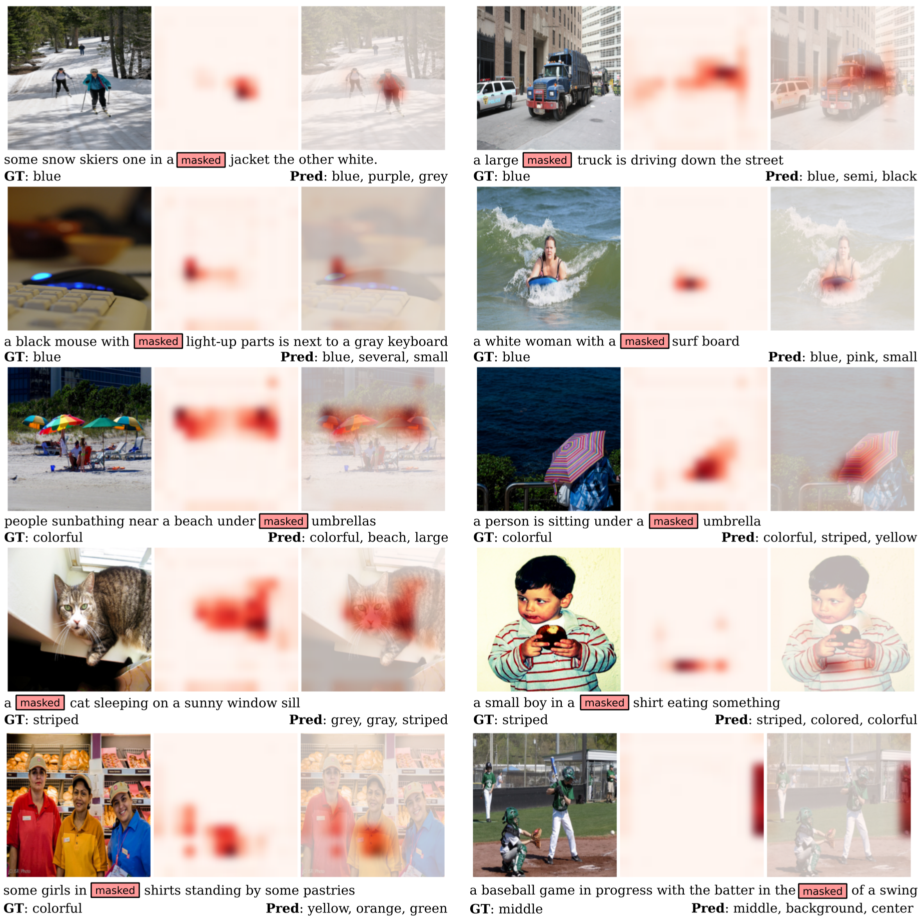
some snow skiers one in a masked jacket the other white.
GT: blue
Pred: blue, purple, grey
a large masked truck is driving down the street
GT: blue
Pred: blue, semi, black
a black mouse with masked light-up parts is next to a gray keyboard
GT: blue
Pred: blue, several, small
a white woman with a masked surf board
GT: blue
Pred: blue, pink, small
people sunbathing near a beach under masked umbrellas
GT: colorful
Pred: colorful, beach, large
a person is sitting under a masked umbrella
GT: colorful
Pred: colorful, striped, yellow
a masked cat sleeping on a sunny window sill
GT: striped
Pred: grey, gray, striped
a small boy in a masked shirt eating something
GT: striped
Pred: striped, colored, colorful
some girls in masked shirts standing by some pastries
GT: colorful
Pred: yellow, orange, green
a baseball game in progress with the batter in the masked of a swing
GT: middle
Pred: middle, background, center

Fig. 2. Qualitative results. For several image-caption pairs of the validation set of the COCO dataset and for a masked token, we show the ground-truth label (GT) together with the top 3 predictions (Pred) and the attention map generated by our ICMLM$_{\texttt{att-fc}}$ with VGG16 backbone. The red parts correspond to higher attentions.

the different colors of the three shirts) is still way beyond the capacity of the ICMLM model.

E Transformer network in ICMLM

This section extends Sec. 3.2 of the main paper and describes in detail the transformer encoder layer [17] in our ICMLM$_{\texttt{tfm}}$ model.

In ICMLM$_{\texttt{tfm}}$, we use the multi-headed attention network proposed in [17] in order to contextualize the token embeddings computed by BERT$_{\text{base}}$ model, *i.e.* $\mathbf{W}_i \in \mathbb{R}^{T \times d_{\text{w}}}$, among the visual features mapped to the token embedding space of BERT$_{\text{base}}$, *i.e.* $\bar{\mathbf{X}}_i \in \mathbb{R}^{H \times W \times d_{\mathrm{w}}}$, for the i-th data sample. To do so, in our model, we use 1-layer transformer encoder with 8 attention heads which are computed in parallel. The transformer encoder takes as input the concatenation of $\bar{\mathbf{X}}_i$ and $\mathbf{W}_i$, *i.e.* $Z_i = [\bar{\mathbf{X}}_i; \mathbf{W}_i] \in \mathbb{R}^{S \times d_{\mathrm{w}}}$, where $S = (H \times W + T)$ denotes the total number of (visual + textual) tokens.

Each attention head $O_h, h \in 1, \cdots, 8$ in the encoder performs the *scaled dot-product attention* [17] on top of Z_i as follows. First, 3 linear projections of Z_i are computed:

$$\begin{aligned} K_i^h &= Z_i \Sigma_K^h + b_K^h, \\ Q_i^h &= Z_i \Sigma_Q^h + b_Q^h, \\ V_i^h &= Z_i \Sigma_V^h + b_V^h, \end{aligned} \tag{5}$$

where K_i^h, Q_i^h and V_i^h are respectively the keys, queries and values $\in \mathbb{R}^{S \times d_{\mathrm{w}}}$ computed by the attention head h. In this formulation, Σ_K^h, Σ_Q^h and $\Sigma_V^h \in \mathbb{R}^{d_{\mathrm{w}} \times d_{\mathrm{w}}}$ are weight; b_K^h, b_Q^h and $b_V^h \in \mathbb{R}^{d_{\mathrm{w}}}$ are bias parameters of the projection layers in O^h. Then the output of each head $O^h(Z_i) \in \mathbb{R}^{S \times d_{\mathrm{w}}}$ is computed using the keys, queries and values defined above:

$$O^h(Z_i) = \operatorname{softmax}\left(\frac{K_i^h {Q_i^h}^\top}{\sqrt{D}} \right) V_i^h. \tag{6}$$

Finally all attention heads are merged simply by concatenating the individual head's outputs, and we compute:

$$O(Z_i) = \left[O^1(Z_i) \,|\, \cdots \,|\, O^8(Z_i) \right] \Sigma^O + b^O, \tag{7}$$

where $\Sigma_O \in \mathbb{R}^{8 \times d_{\mathrm{w}} \times d_{\mathrm{w}}}$ and $b^O \in \mathbb{R}^{d_{\mathrm{w}}}$ are learnable parameters, and [.|.] denotes concatenation. The output of the multi-headed attention layer is followed by residual connection [7], dropout [16], LayerNorm [2], ReLU and linear projection layers to obtain the final output of the transformer.

F Implementation details

This section provides technical details of both training model for proxy tasks and evaluating them on target tasks.

F.1 Training for proxy tasks

With VGG16 backbones. We start training VGG16 networks on the Visual Genome (VG) or MS-COCO datasets by solving the rotation prediction task [5]. Note that we do not use any of the existing RotNet [5] pretrained models as they all have processed millions of images. Contrarily, we want to restrict all the training steps of our pipeline to access only a small dataset of images (103K and 118K training images of VG and COCO respectively). For that, first, we train separate VGG16 networks on VG or COCO for 120K iterations using RAdam [12] with batches of size 128, initial learning rate $1e-3$, weight decay $1e-5$, and the learning rate is decayed by 0.1 after 100K and 110K iterations. Once the networks are trained for the rotation prediction task, we remove the fully-connected layers from the networks and fine-tune the CNN backbones by solving the proxy tasks we defined in Secs. 3.1 and 3.2 of the main paper.

We train $\text{TP}_{\star}$ models for 100K iterations using RAdam optimizer [12] with batches of size 128, initial learning rate 1e-4, weight decay 1e-3, and the learning rate is decayed by 0.1 after 80K and 90K iterations. For $\text{TP}_{\star}$ models, the number of data samples is equal to the number of images in the training sets (103K in VG and 118K in COCO). The number of unique triplets (image, caption, masked token) that we use during training ICMLM models varies from 2.5M to 13M depending on the dataset and the label set used, because we design the triplets in a way that for each (image, caption) pair, there is only one masked token so many triplets are built for a single (image, caption) pair. To reduce the training time, we train them for 200K iterations using batches of size 896 (distributed over 4 NVIDIA V100 GPUs). We note that in early ICMLM trainings, attention heads (`att` modules in $\text{ICMLM}_{\texttt{att-fc}}$ and self-attention attention heads in $\text{ICMLM}_{\texttt{tfm}}$) produce almost uniform attention distributions over the spatial grid of visual features. Therefore, in $\text{ICMLM}_{\texttt{att-fc}}$ models, we find that warming up the attention heads for 50K iterations while freezing VGG16 backbones prevents noisy gradients to flow through backbones.

With ResNet50 backbones. We train TP_{Label} and $\text{TP}_{\text{Postag}}$ models from scratch for 100K iterations using SGD with momentum (0.9) optimizer with batches of size 128, initial learning rate 3e-2, weight decay 1e-4, and the learning rate is decayed by a cosine-based schedule. We initialize ResNet50 backbones in $\text{ICMLM}_{\star}$ models with pretrained $\text{TP}_{\text{Postag}}$ checkpoints then train $\text{ICMLM}_{\star}$ models for 500K iterations using the same optimizer configuration except that batch size is 512.

We validate all hyper-parameters and design choices on the validation sets of VG and COCO. As we note in Sec. 3.2 of the main paper, while training $\text{ICMLM}_{\star}$ models, we freeze the pretrained $\text{BERT}_{\text{base}}$ model available in HuggingFace repository[1]. We use PyTorch [13] and the mixed-precision functionality provided by NVIDIA Apex[2] to perform all experiments.

F.2 Evaluation on target tasks

We follow two different evaluation practices to compare models:

(i) Probing linear logistic regression classifiers after various layers in VGG16 backbones and training them with SGD updates and data augmentation. For this evaluation, we use the publicly-available code of [3] and slightly modify it such that heavier data augmentation is applied and classifiers are trained for more iterations. We will share the training configuration for each

[1] `https://github.com/huggingface/transformers`

[2] `https://github.com/NVIDIA/apex`

setting. For the details of the evaluation practice, please refer to the code repository of [3][3].

(ii) Extracting image features from the last convolutional layer of ResNet50 backbones and training linear SVMs and logistic regression classifiers using these pre-extracted features.

Note that in both cases, backbones are frozen.

Feature extraction. To extract image features, we resize images such that their smallest dimension is 224 pixels, then apply central-crops of size 224×224. This gives us $7 \times 7 \times 2048$-dimensional visual tensors output for ResNet-50 backbones. For training SVMs on VOC and COCO, following [6], we apply 2×2 spatial average pooling and flattening to obtain 8192-dimensional visual features, then ℓ_2-normalize the features. However, storing and training classifiers on 8192-dimensional features for the 1.28M images of the IN-1K dataset was computationally challenging. Therefore, for training logistic regression classifiers on IN-1K, we apply global average pooling and obtain 2048-dimensional visual features.

SVM classifiers. Following the convention of [6], we train linear SVMs to evaluate visual representations on the 2007 split of Pascal-VOC and the 2017 split of MS-COCO datasets, in a one-*vs.*-all manner. Please refer to [6] for details in training binary SVMs. Different from [6], we tune the cost parameter of SVMs by sampling 40 cost values log-uniformly between 10^{-5} and 10^{5} and find the optimal value by Optuna [1].

Logistic regression classifiers. We train linear logistic regression classifiers by performing SGD updates with momentum 0.9 and batch size 1024. We validate the learning rate and weight decay hyper-parameters using Optuna [1] over 25 trials. We log-uniformly sample learning rates between 10^{-1} and 10^{2}, and apply cosine-based learning rate annealing, whereas we uniformly sample weight decays between 0 and 10^{-5}.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proc. ICKDDM (2019) 10
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv:1607.06450 (2016) 8
3. Caron, M., Bojanowski, P., Mairal, J., Joulin, A.: Unsupervised Pre-Training of Image Features on Non-Curated Data. In: Proc. ICCV (2019) 9, 10
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results 1
5. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: Proc. ICLR (2018) 8
6. Goyal, P., Mahajan, D., Gupta, A., Misra, I.: Scaling and benchmarking self-supervised visual representation learning. In: Proc. ICCV (2019) 10
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. CVPR (2016) 8

[3] https://github.com/facebookresearch/DeeperCluster

8. Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting Self-Supervised Visual Representation Learning. In: Proc. CVPR (2019) 3
9. Kornblith, S., Shlens, J., Le, Q.V.: Do better imagenet models transfer better? In: Proc. CVPR (2019) 5
10. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV (2017) 4
11. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Proc. ECCV (2014) 1, 4
12. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J.: On the variance of the adaptive learning rate and beyond. In: Proc. ICLR (2020) 9
13. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Proc. NeurIPS (2019) 9
14. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV **115**(3) (2015) 1
15. Sariyildiz, M.B., Cinbis, R.G.: Gradient matching generative networks for zero-shot learning. In: Proc. CVPR (2019) 4
16. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. JMLR **15**(1) (2014) 8
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proc. NeurIPS (2017) 7, 8
18. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011) 3
19. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. PAMI **41**(9) (2018) 3, 4