Suplementary Material for
MAGMA : Manifold Regularization for MAEs

Alin Dondera*,1, Anuj Singh*,1,2, Hadi Jamali-Rad1,2
1Delft University of Technology (TU Delft), The Netherlands
2Shell Global Solutions International B.V., Amsterdam, The Netherlands
a.e.dondera@student.tudelft.nl, {a.r.singh, h.jamalirad}@tudelft.nl

1. Experimental setup

Environment details. MAGMA builds upon the solo-learn [3] library of self-supervised methods for unsupervised visual representation learning. All methods are implemented using PyTorch 1.13 and PyTorch Lightning 1.7.7. The following GPUs are used, depending on availability: NVIDIA GeForce RTX 2080 Ti, NVIDIA Tesla V100, and NVIDIA A40.

Datasets. We conduct our experiments on the following four benchmark datasets:

- **CIFAR-100** [5] consists of 60,000 color images (32x32 pixels) divided into 100 classes, with 500 training images and 100 test images per class. This large number of classes with relatively few images per class pushes models to learn nuanced, discriminative representations for robust classification.
- **STL-10** [2] Contains 5,000 labeled training images, 8,000 test images, and 100,000 unlabeled images (96x96 pixels) across 10 classes. This setting of abundant unlabeled data allows the exploration of self-supervised representation learning techniques, offering a valuable testbed for scenarios where labeled data is scarce.
- **Tiny-ImageNet** [6] is a downsized version of ImageNet with 200 classes, featuring 100,000 training images, 10,000 validation images, and 10,000 test images (64x64 pixels). This dataset bridges the gap between smaller benchmarks and full ImageNet, allowing experimentation with larger-scale image recognition tasks while maintaining computational feasibility.
- **ImageNet-100** [7] is a curated subset of the full ImageNet with approximately 130,000 images (variable resolutions) across 100 classes. It provides a standard train/test split, offering a manageable platform to test the scalability and efficiency of models before moving to the full complexity of ImageNet.

This collection of datasets provides a larger range of image classification challenges by varying scales, class complexities, and train/test splits. This suite enables a robust evaluation of the effectiveness of representation learning methods and their generalization across diverse scenarios.

Pretraining hyperparameters. We split the parameters into three categories: (i) common parameters across all methods and datasets, (ii) parameters used for the MAE-based methods (MAE [4], M-MAE, U-MAE [8], and MU-MAE), (iii) parameters used for SimCLR [7], M-SimCLR, VICReg [1], and M-VICReg. The complete configuration files for all combinations of datasets and methods can also be found in the attached code archive.

(i) Common parameters. All methods use AdamW as an optimizer, with an initial warmup phase of 10 epochs, and an initial learning rate of $3e-5$ decaying to 0 via cosine annealing. Normalization is applied using the specific mean and standard deviation computed across each given dataset.

(ii) MAE-based methods. Mask ratio for all parameters is 0.75, following [4]. For U-MAE and MU-MAE, the uniformity weight is set to 0.01, following [8]. The weight for the MAGMA loss is set to 1. For augmentations, we use a random resized crop (scale ranging between 0.08 and 1), followed by a random horizontal flip with a probability of 0.5. The crop is resized to 32×32 for CIFAR-100, 64×64 for Tiny-ImageNet, 96×96 for STL-10, and 224×224 for ImageNet-100. All other parameters unrelated to the regularization terms are shared between all methods, and only depend on the dataset. These can be seen in Table 1.

(iii) Non-generative SSL methods. For SimCLR and M-SimCLR we use a temperature of 0.2. For VICReg and M-VICReg, we use the best weights from [1] for the similarity, variance, and covariance loss terms (25, 25, and 1). The hidden dimensionality of the projector is equal to 2048 for all. For augmentations, each method follows the parameters described in the original paper. The rest of the relevant parameters can be found in Table 2.

* equal contribution

Table 1. Sets of differing parameters for MAE, M-MAE, U-MAE, and MU-MAE across the given datasets

Dataset	Backbone	Patch Size	Epochs	Batch Size	lr	e_{st} (Reg. warmup)
CIFAR-100	ViT-Tiny	4	2000	256	$1.5e^{-4}$	60
Tiny-ImageNet	ViT-Tiny	8	800	512	$1.0e^{-3}$	10
STL-10	ViT-Tiny	12	800	512	$3.0e^{-4}$	10
ImageNet-100	ViT-Base	16	400	256	$1.5e^{-4}$	10

Table 2. Sets of differing parameters for SimCLR, M-SimCLR, VICReg, and M-VICReg across the given datasets

Dataset	Backbone	Patch Size	Epochs	Batch Size	lr	e_{st} (Reg. warmup)
CIFAR-100	ViT-Tiny	4	1000	256	$1.0e^{-3}$	10
Tiny-ImageNet	ViT-Tiny	8	1000	256	$1.0e^{-3}$	10
STL-10	ViT-Tiny	12	1000	256	$1.0e^{-3}$	10
ImageNet-100	ViT-Tiny	16	200	256	$1.0e^{-3}$	10

Table 3. Parameter, Throughput in Images/seconds (Img/sec) and GPU Memory for MAE, U-MAE w and w/o regularization.

Method	Parameters	Img/sec	GPU Memory
MAE	122M	489	28.0 GB
M-MAE (ours)	122M	481	28.1 GB
U-MAE	122M	486	28.0 GB
MU-MAE (ours)	122M	476	28.1 GB

2. Computational Complexity

We compute the total number of parameters, time efficiency measured by throughput (images per second) and memory efficiency by peak GPU-memory consumption of MAE, UMAE with and without our regularization, and compare them in Table 3. Given the additional cost of computing the sample-wise similarity matrix / laplacian across an entire batch, M-MAE offers a 1.5% and MU-MAE a 2% drop in throughput as compared to their respective MAE and U-MAE baselines. This translates to a minor increase of 100MB GPU-memory consumption during training, thus adding an insignificant extra computation cost to the baseline methods. The parameter count of all methods remain the same since our regularization method operates on the same architecture as the MAE and U-MAE baselines.

3. Additional visualizations

We provide additional visualizations of the PCA and attention map results in Figure 1 and Figure 2, as presented in Section 5.4, using a broader selection of images sampled from the ImageNet validation set. These additional visualizations further confirm the previously observed patterns and trends.

References

[1] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 1

[2] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 1

[3] Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. solo-learn: A library of self-supervised methods for visual representation learning. *Journal of Machine Learning Research*, 23(56):1–6, 2022. 1

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 1

[5] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[6] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 1

[7] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020. 1

[8] Qi Zhang, Yifei Wang, and Yisen Wang. How mask matters: Towards theoretical understandings of masked autoencoders. *Advances in Neural Information Processing Systems*, 35:27127–27139, 2022. 1

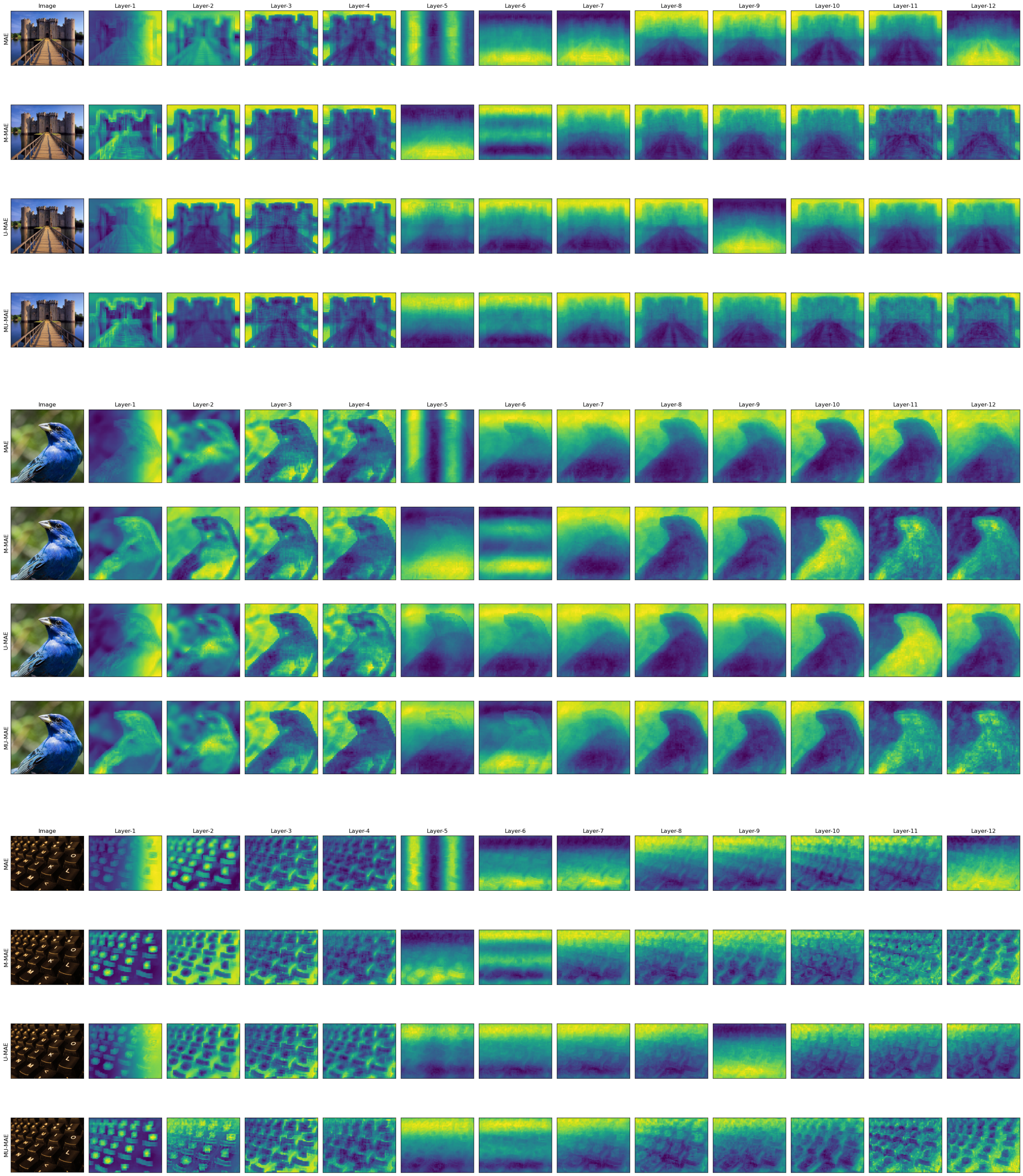

Figure 1. Additional visualization of PCA's leading component for features extracted from different layers of a ViT-B pretrained using MAE, M-MAE (ours), U-MAE, and MU-MAE (ours).

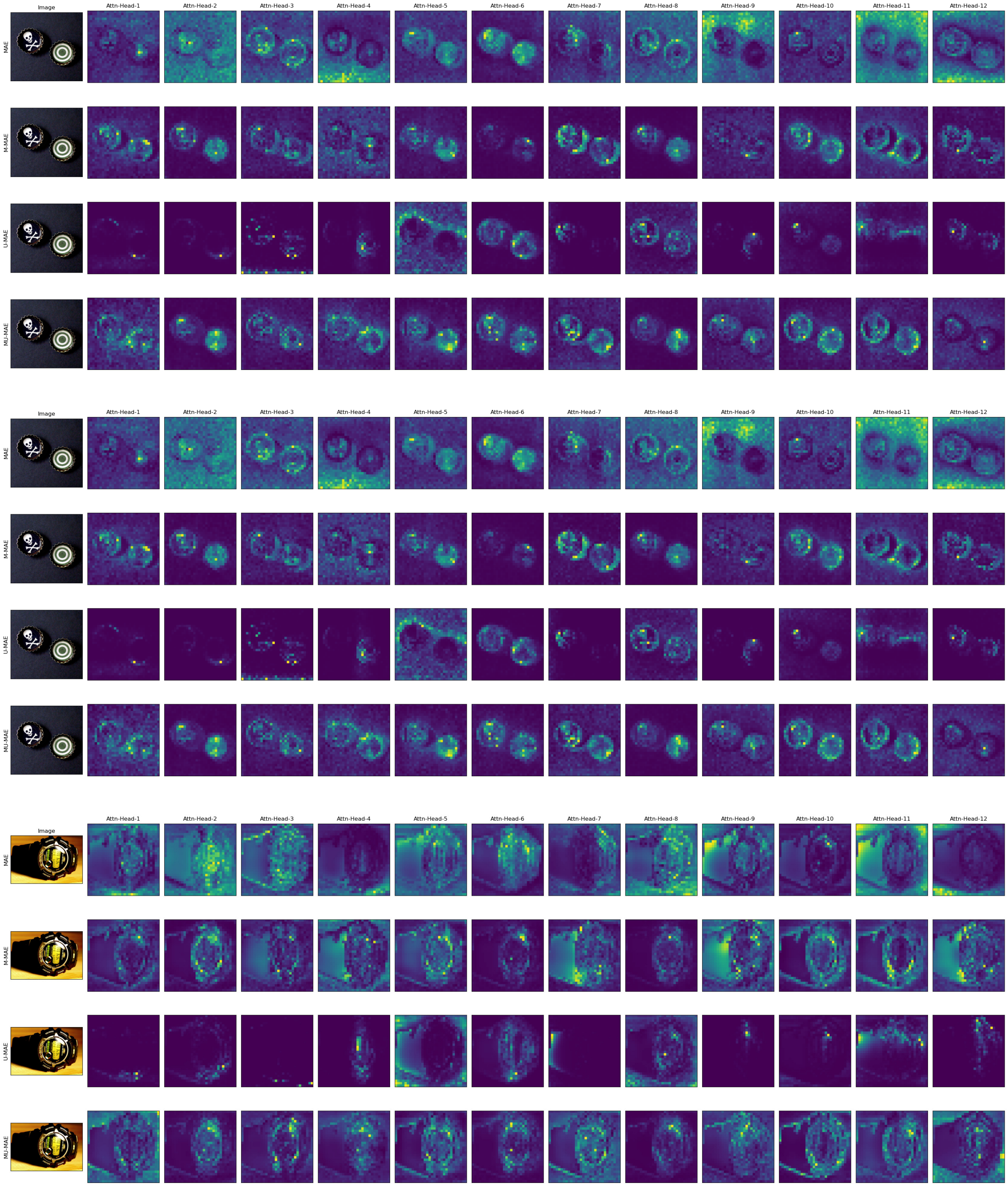

Figure 2. Additional attention maps from the 12 attention heads of the last layer of a ViT-B. The attention maps come from three different images, and for each image, we extract them over the four MAE-based methods evaluated: MAE, M-MAE (ours), U-MAE, MU-MAE (ours)