Transformed Autoencoder: Pre-training with Mask-Free Encoder and Transformed Decoder

Yichen Zhou[1,2] **Pan Zhou**[1] **Chenyang Si**[1] **Weihao Yu**[1,2]
Teck Khim Ng[2] **Shuicheng Yan**[1]
[1]Sea AI Lab [2]National University of Singapore
`{zhouyc,zhoupan,sicy,yuweihao,yansc}@sea.com` `ngtk@comp.nus.edu.sg`

Abstract

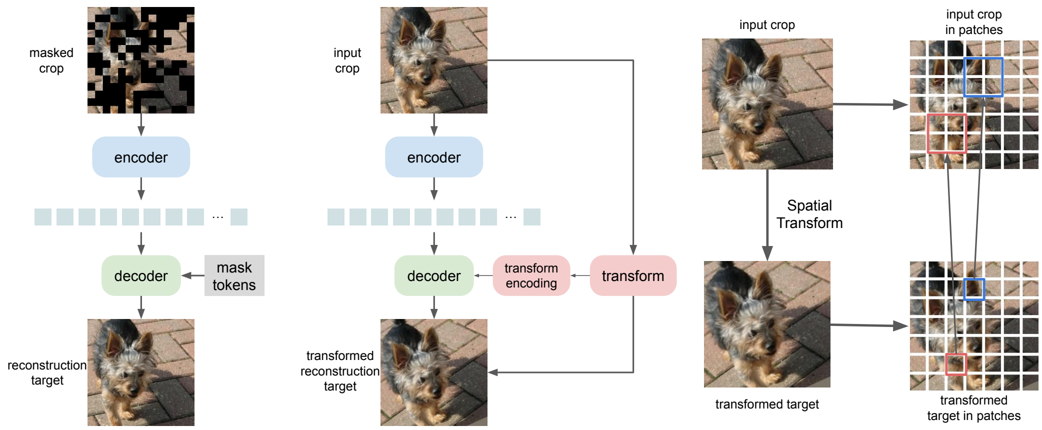
In this work, **T**ransformed **A**uto**E**ncoder (TAE) is proposed towards using mask-free encoder and transformed decoder for effective self-supervised learning with consistent encoder for downstream tasks. Specifically, TAE feeds a full input $\boldsymbol{x}$ into an encoder, and then feeds the randomly masked output patch tokens along with a parameter embedding $\boldsymbol{e}_{\mathcal{T}_{\mathrm{s}}}$ of a random spatial transformation $\mathcal{T}_{\mathrm{s}}$ into a decoder. Next, TAE requires the decoder to predict the pixels or the semantic feature of the transformed target $\mathcal{T}_{\mathrm{s}}(\boldsymbol{x})$. This mask-free encoder pre-training differentiates TAE from the existing masked image modeling frameworks in two aspects. First, TAE is training-tuning consistent, *i.e.* taking a full input image for both encoder pre-training and fine-tuning, while the MAE family takes masked input for pre-training while non-masked input for fine-tuning. Secondly, TAE enjoys high encoder architecture compatibility to popular ViTs, CNNs and MLP-based networks, compared to MAE with its masking strategy on encoder. Furthermore, the design of transformed decoder in TAE is unique, and can be used as an extra training objective by most existing algorithms for self-supervised learning to further boost performance. Extensive experiments well verified the effectiveness of TAE and its variants.

1 Introduction

Self-supervised learning (SSL) [1–13] aims to train a highly transferable deep model on unlabeled data by solving a well-designed pretext task which can generate pseudo targets for the task itself. Among current SSL approaches, the performance of the recently proposed masked image modeling frameworks [14–18], *e.g.* MAE [14] and data2vec [18], has surpassed that of "end-to-end supervised learning" by a significant margin on classification, object detection and segmentation tasks, and its success is increasingly scaled to other tasks thanks to its compatibility and effectiveness. For pre-training phase, as shown in Fig. 1a, MAE feeds a randomly masked input image into an encoder, and then requires a decoder to reconstruct the pixels or features of the masked patches from the latent representation of the encoder and mask tokens. One can observe that the core of this framework is the masking on the encoder input, which unfortunately causes inconsistency between the pre-training and fine-tuning phases. Specifically, for the encoder, the input is a masked or incomplete one in the pre-training phase, while it is complete without mask in the fine-tuning phase. This inconsistency may impair the performance of the masked image modeling frameworks. Moreover, though being well compatible to the vision transformers (ViT) [19, 20] encoder, the masking strategy on the encoder input employed by MAE prohibits the pre-training of other popular and effective encoder architectures, *e.g.* CNN [21, 22], MLP-based architectures [23, 24], or others [25, 26]. As these popular architectures cannot handle incomplete input due to convolutions & pooling operations in CNNs, or fully-connected layers in MLP-based architectures.

In this work, for the first time, we propose a self-supervised mask-free encoder pre-training framework, termed as Transformed **T**ransformed **A**uto**E**ncoder (TAE). The core idea of TAE is to defer the mask

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) Masked Autoencoder (b) Transformed Autoencoder (c) Patch correspondence in TAE

Figure 1: **Comparison between MAE and our TAE**.

on the encoder input of the MAE-like framework to the decoder input while enhancing the encoder to learn the data semantics and patch interdependencies by injecting spatial transformations.

2 Transformed Autoencoder

In this section, we will elaborate on the proposed Transformed Autoencoder (TAE) for mask-free encoder pre-training. As shown in Fig. 1b, TAE uses an encoder f to encode a full input image crop $\boldsymbol{x}$ into a set of latent patch tokens $\boldsymbol{z}$, and then adopts a decoder g to recover the spatially transformed pixels $\mathcal{T}_{\mathrm{s}}(\boldsymbol{x})$ of the input $\boldsymbol{x}$ from randomly masked latent patch tokens $\boldsymbol{z}$ and the parameter embeddings of the spatial transformation $\mathcal{T}_{\mathrm{s}}$ along with learnable mask tokens.

TAE Decoder. Our decoder g consists of a series of standard transformer blocks. In the following, we introduce how our decoder processes the latent patch tokens $\boldsymbol{z}$ given by the encoder f.

To begin with, we randomly select a spatial transformation $\mathcal{T}_{\mathrm{s}}$ for an image $\boldsymbol{x}$, and encode the hyper-parameters $\boldsymbol{\sigma}$ of $\mathcal{T}_{\mathrm{s}}$ into an embedding $\boldsymbol{e}_{\mathcal{T}_{\mathrm{s}}}$ via a 2-layered MLP: $\boldsymbol{e}_{\mathcal{T}_{\mathrm{s}}} = \mathsf{MLP}(\boldsymbol{\sigma}) \in \mathbb{R}^d$, where d denotes the dimension of the latent patch tokens $\boldsymbol{z}$. Here to implement spatial transformation $\mathcal{T}_{\mathrm{s}}$, we use homography transformation with 8 degrees of freedom.

Then we randomly mask the latent patch tokens $\boldsymbol{z}$ by replacing the selected masked tokens with a shared and learned mask token [14, 27]. Next, we add positional embeddings to all tokens in $\boldsymbol{z}$ to tell the locations of all tokens in the vanilla image $\boldsymbol{x}$. Finally, we concatenate the hyper-parameter embedding $\boldsymbol{e}_{\mathcal{T}_{\mathrm{s}}}$ of $\mathcal{T}_{\mathrm{s}}$ to each token in $\boldsymbol{z}$ so as to tell each token what spatial transformation is performed. Actually, we also explore directly adding $\boldsymbol{e}_{\mathcal{T}_{\mathrm{s}}}$ to each token in $\boldsymbol{z}$, which works equally well as shown in our experiments. Then we feed $\boldsymbol{z}$ into the decoder g to obtain the prediction $\boldsymbol{y}'$.

Reconstruction Target. Regarding the reconstruction target in TAE, as shown in Eqn. (1), there are two possible solutions: i) we recover a spatially transformed pixels $\boldsymbol{y}$ in the image $\boldsymbol{x}$; and ii) we reconstruct a spatially transformed semantic feature $\boldsymbol{y}$ of the image $\boldsymbol{x}$.

$$\boldsymbol{y} = \begin{cases} \mathcal{T}_{\mathrm{s}}(\boldsymbol{x}), & \text{if the target is pixel reconstruction;} \\ \mathcal{T}_{\mathrm{s}}(f'(\boldsymbol{x})), & \text{if the target is feature reconstruction;} \end{cases} \tag{1}$$

where f' is the exponentially moving average of f. Now we are ready to define the training loss of TAE as $\min_{f,g} \sum \ell(\boldsymbol{y}_i, \boldsymbol{y}'_i)$ where $\boldsymbol{y}_i$ denotes the i-th token in $\boldsymbol{y}$. Here the loss function ℓ measures the discrepancy between the prediction $\boldsymbol{y}'_i$ and the ground truth $\boldsymbol{y}_i$, *e.g.*. the mean-square-error (MSE), cosine distance and KL Divergence.

The spatial transformation $\mathcal{T}_{\mathrm{s}}$ on the reconstruction target is a key component in TAE. It helps the encoder to better learn the dependency among different patches in an image and also enhances data semantics learning. Specifically, as illustrated in Fig. 1c, TAE needs to first partition the encoder input into the non-overlap patches to tokenize them into a series of patch tokens $\boldsymbol{z}$, which are then fed into the Autoencoder to obtain a series of patch pixels $\boldsymbol{y}'$ for predicting a spatially transformed target $\mathcal{T}_s(\boldsymbol{x})$. Since the encoder input $\boldsymbol{x}$ differs from the target $\mathcal{T}_s(\boldsymbol{x})$ due to the spatial transformation $\mathcal{T}_s$, the spatial partition for the patch tokens $\boldsymbol{z}$ in the encoder and decoder distinguishes from the one in the target $\mathcal{T}_s(\boldsymbol{x})$. This means that there is no exact one-to-one correspondence between the patches in the patch tokens $\boldsymbol{z}$ and the target $\mathcal{T}_s(\boldsymbol{x})$. Actually, as shown in Fig. 1c, the content of one token in $\boldsymbol{z}$ can be separated into several nearby patches in $\mathcal{T}_s(\boldsymbol{x})$. Therefore, the prediction content of one token in $\boldsymbol{y}$ actually comes from several nearby patches in $\boldsymbol{z}$. This indicates that TAE encoder and decoder need to exchange sufficient information among tokens for fusing several nearby token patches together to achieve small reconstruction loss. This accordingly induces patch dependency learning and also enhances learning of data semantics. Moreover, by applying masks on the decoder input, some of the necessary nearby tokens may be masked. This further boosts the *encoder* to exchange sufficient information among tokens such that each unmasked token in the decoder has contained enough information of other tokens and the decoder can use them to well predict the masked patches.

Table 1: **Fine-tuning** classification accuracy (%) on ImageNet-1K. All methods are pre-trained for 300 epochs, and fine-tuned with class label supervision for 200 epochs on ImageNet-1K.

Supervision	RGB pixel values			Unsupervised feature		
Method	MAE [28]	SimMIM [15]	TAE (ours)	MAE-like	data2vec [18]	TAE (ours)
Mask at encoder	75%	60%	0%	75%	60%	0%
ViT-Small	80.8	80.5	80.7	81.2	80.9	81.0
ResNet-50	-	78.0	**78.4**	-	Failed	**77.8**
MLPMixer-B/16	-	79.1	**79.3**	-	78.5	**78.8**

2.1 Discussion

Now we are ready to emphasize the three advantages of our proposed TAE.

Firstly, with the mask-free encoder pre-training mechanism, for both pre-training and and fine-tuning phases, TAE always feeds the full input image crop into the encoder. In this way, the TAE encoder always sees the whole picture of the input, and thus can consistently process the input tokens. Differently, for the MAE-like framework, the encoder input is masked in the pre-training phase but not masked in the fine-tuning phase, which causes inconsistency between the two phases.

Secondly, TAE encoder can be well compatible to many popular and effective network architectures, including not only ViTs but also CNNs and MLP-based networks. This compatibility also comes from the mask-free strategy on the TAE encoder. In contrast, the previous MAE-like framework is often not suitable for non-ViT architectures and suffers from an architecture compatibility issue due to convolutions & pooling operations.

Thirdly, TAE with transformed image reconstruction is actually a very general framework and is orthogonal to many SSL families, such as MAE-like frameworks and contrastive learning methods [13, 29–33]. One can combine TAE with other SSL approaches to enjoy merits of both sides. Indeed, our experimental results in Sec. 3 show that integrating the transformed image reconstruction task in TAE with the MAE-like framework, *e.g.*MAE, can improve their performance.

3 Experiments

3.1 Experiment Settings

Following existing works [15, 18, 28], we perform pre-training on the ImageNet-1k [34] dataset. The spatial input dimension is 224×224 for all backbone models. We pre-train each model for 300 epochs with a total batch size 2,048 distributed on 16 GPUs, using AdamW [35] optimizer with weight decay at 0.05. For learning rate, we warmup during the first 20 epochs to 0.00015 per 256 batch size, followed by a cosine decay schedule. We apply horizontal flipping and random resized cropping of scale [0.2, 1.0] as the default data augmentation for all experiments.

For evaluating the performance on ImageNet for pre-trained models, we follow the common practice of end-to-end fine-tuning settings in similar works [28, 36] by applying layer-wise learning rate decay [37]. We perform fine-tuning of 200 epochs for ViT-Small [38], ResNet-50 [39] and MLPMixer-B/16 [23], and 100 epochs for ViT-Base [38].

3.2 Effectiveness of Transformed Autoencoder

For evaluating the effectiveness of the proposed TAE framework, we compare different backbone models pre-trained and fine-tuned under the same settings with different methods. The implementation of TAE does not apply masking of any kind on the encoder. MAE [28] and SimMIM [15] are two similar methods that also aim to reconstruct pixel values. We mask 60% and 75% of tokens at the encoder for SimMIM and MAE respectively. Note that as MAE removes tokens before the decoder, it can only work with ViT, but not with CNNs or MLP based models which require regular and fixed spatial shape to perform convolution or spatial-MLP operation. Table 1 shows the fine-tuning accuracies of different self-supervised learning methods on 3 different types of backbone models. Without using masks at the encoder, our TAE can perform similarly or even favourably on ViT-S, ResNet-50, and MLPMixer-B/16 than MAE and SimMIM. Notably, the MLPMixer-B/16 model pre-trained with our TAE can achieve 79.3% accuracy on, higher than the 79.1% obtained by SimMIM, and significantly better than the supervised trained baseline at 76.4% [23].

We further evaluate the performance of TAE with unsupervised features as the reconstruction target. Similar to data2vec [18], we use the exponential moving average (EMA) of the encoder as the teacher network to extract the unsupervised features as the reconstruction target. We set the initial momentum parameter $\tau = 0.99$, which is increased to 1 during training with a cosine schedule following [40]. Here we also compare with another MAE-like variant for pre-training ViT, by removing some tokens at the encoder, and adds back a corresponding number of mask tokens at the decoder. The difference between this variant and standard MAE [28] training is that here the reconstruction target is unsupervised features extracted by an EMA encoder, rather than RGB pixel values. As shown in the right section of Table 1, our TAE works well on different backbones with the unsupervised feature as targets, without any masking on the encoder.

3.3 Ablation studies

Here we study the effects of proposed changes of the Transformed Autoencoder. We first ablate the proposed spatial transformation. As we can see, there is a non-trivial drop in performance from 80.7% to 80.3% and 81.0% to 80.7% when using the RGB and unsupervised feature as targets respectively. We also evaluate the vanilla autoencoder setting with no masking or any form of transformations for the two types of targets. Our proposed TAE can outperform vanilla AE by 1.2% and 1.0% respectively with the RGB and unsupervised feature as reconstruction targets.

Table 2: **Ablation studies** of TAE pre-training for ViT-S/16 on ImageNet-1K.

Variant	Top-1 Acc (%)	
	RGB	Unsup. Feat.
TAE	**80.7**	**81.0**
TAE without spatial transformation	80.3	80.7
Vanilla AutoEncoder	79.5	80.0

4 Conclusion

In this work, we introduce Transformed Autoencoder (TAE), a novel and general framework for self-supervised pre-training of vision models. Our TAE does not rely on masking at the encoder backbone during the pre-training phase. Unlike masked image modeling methods developed for pre-training of ViTs, our TAE framework is a general framework compatible with a wide range of vision backbone models. Experiment results show that our TAE can perform on par with existing state-of-the-art methods with masking at the encoder of ViTs, and favourably compared with other methods on CNN and MLP-based backbones. Moreover, TAE is orthogonal to most exiting self-supervised learning approaches, and can be combined with them to further boost their performance.

References

[1] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[2] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742. IEEE, 2006.

[3] R. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[4] A. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[5] P. Bachman, R. Hjelm, and W. Buchwalter. Learning representations by maximizing mutual information across views. In *Proc. Conf. Neural Information Processing Systems*, pages 15535–15545, 2019.

[6] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Proc. Conf. Neural Information Processing Systems*, volume 27, pages 487–495, 2014.

[7] M. Bautista, A. Sanakoyeu, E. Tikhoncheva, and B. Ommer. Cliquecnn: Deep unsupervised exemplar learning. In *Proc. Conf. Neural Information Processing Systems*, pages 3846–3854, 2016.

[8] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[9] M. Caron, P. Bojanowski, A. Joulin, and Matthijs M. Douze. Deep clustering for unsupervised learning of visual features. In *Proc. European Conf. Computer Vision*, pages 132–149, 2018.

[10] Z. Wu, Y. Xiong, S. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[11] J. Huang, Q. Dong, S. Gong, and X. Zhu. Unsupervised deep learning by neighbourhood discovery. *arXiv preprint arXiv:1904.11567*, 2019.

[12] X. Yan, I. Misra, A. Gupta, D. Ghadiyaram, and D. Mahajan. Clusterfit: Improving generalization of visual representations. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 6509–6518, 2020.

[13] S. Lin, P. Zhou, Z. Hu, S. Wang, R. Zhao, Y. Zheng, L. Lin, E. Xing, and X. Liang. Prototypical graph contrastive learning. *arXiv preprint arXiv:2106.09645*, 2021.

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[15] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*, 2021.

[16] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[17] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Peco: Perceptual codebook for bert pre-training of vision transformers. *arXiv preprint arXiv:2111.12710*, 2021.

[18] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022.

[19] 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. In *International Conference on Learning Representations*, 2020.

[20] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[22] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

[23] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34, 2021.

[24] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[26] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. *arXiv preprint arXiv:2111.11418*, 2021.

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[29] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[30] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int'l Conf. Machine Learning*, 2020.

[31] Pan Zhou, Caiming Xiong, Xiaotong Yuan, and Steven Hoi. A theory-driven self-labeling refinement method for contrastive representation learning. In *Neural Information Processing Systems*, 2021.

[32] J. Li, P. Zhou, C. Xiong, R. Socher, and S. CH Hoi. Prototypical contrastive learning of unsupervised representations. In *Int'l Conf. Learning Representations*, 2021.

[33] Pan Zhou, Yichen Zhou, Chenyang Si, Weihao Yu, Teck Khim Ng, and Shuicheng Yan. Mugs: A multi-granular self-supervised learning framework. In *arXiv preprint arXiv:2203.14415*, 2022.

[34] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[35] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018.

[36] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[37] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[38] 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.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[40] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[41] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.

[42] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[43] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[44] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[45] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *arXiv preprint arXiv:2112.09133*, 2021.

[46] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, pages 44–51. Springer, 2011.

[47] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017.

[48] Junying Li, Zichen Yang, Haifeng Liu, and Deng Cai. Deep rotation equivariant network. *Neurocomputing*, 290:26–33, 2018.

[49] Jan Eric Lenssen, Matthias Fey, and Pascal Libuschewski. Group equivariant capsule networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[50] Xifeng Guo, En Zhu, Xinwang Liu, and Jianping Yin. Affine equivariant autoencoder. In *IJCAI*, pages 2413–2419, 2019.

[51] Siamak Ravanbakhsh. Universal equivariant multilayer perceptrons. In *International Conference on Machine Learning*, pages 7996–8006. PMLR, 2020.

[52] Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Learning invariances in neural networks. *arXiv preprint arXiv:2010.11882*, 2020.

[53] T Anderson Keller and Max Welling. Topographic vaes learn equivariant capsules. *Advances in Neural Information Processing Systems*, 34:28585–28597, 2021.

[54] Jin Xu, Hyunjik Kim, Thomas Rainforth, and Yee Teh. Group equivariant subsampling. *Advances in Neural Information Processing Systems*, 34, 2021.

[55] Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. Lietransformer: Equivariant self-attention for lie groups. In *International Conference on Machine Learning*, pages 4533–4543. PMLR, 2021.

[56] Lingshen He, Yuxuan Chen, Yiming Dong, Yisen Wang, Zhouchen Lin, et al. Efficient equivariant network. *Advances in Neural Information Processing Systems*, 34, 2021.

5 Appendix

5.1 TAE for various Backbones

As aforementioned, TAE does not mask the encoder input, and thus can be easily used to train other types of popular and effective architectures, including CNNs (*e.g.* ResNet [39]) and MLP-based networks (*e.g.* MLP-Mixers [23]), *etc*. In principle, to pre-train a non-ViT backbone with TAE, one can directly use this non-ViT backbone to implement the TAE encoder. But for a CNN and MLP-based backbone, one needs to remove its global pooling and Fully Connected layers at the end of the network if any. Besides, for a CNN, *e.g.* ResNet, its output feature map is often of spatial-size 7×7 which is much smaller than the input size 224×224. To make the output feature map preserve more spatial details of the input image, we apply a transposed convolution to the last stage, and then sum it with the feature map from the second last stage to form a feature map of size 14×14. For a MLP-Mixer, its latent patch tokens are the output of the last block like ViT without any special operation. For TAE decoder, we always use standard transformer blocks to implement it for simplicity and consistency. The decoder is always discarded in fine-tuning phase after pre-training.

5.2 Implementation

For spatial transformation, directly applying a transformation on the cropped image might involve extra region(s) not included in the crop, which needs to be padded to keep a consistent size of 224×224. Therefore we perform the transformation on the original image such that the transformed crop takes contents from a region completely within the original image. More specifically, a base crop $\boldsymbol{x}_b$ is defined by the coordinates of its 4 vertices in the original image $p_0 = (x_{\min}, y_{\min})$, $p_1 = (x_{\max}, y_{\min})$, $p_2 = (x_{\max}, y_{\max})$, $p_3 = (x_{\min}, y_{\max})$. We first denote the scale of the crop to be the length of its shorter side $s_{\boldsymbol{x}} = min(x_{\max} - x_{\min}, y_{\max} - y_{\min})$. Then for each vertex p_i, we randomly choose a new point p_i^t within a small squared region of size $\lambda s_{\boldsymbol{x}}$ centered around p_i. By default we set $\lambda = 0.1$ for experiments involving spatial transformation. We then extract the corresponding region with the transformed vertices p_0^t, p_1^t, p_2^t, p_3^t, followed by resizing it to 224×224 to form the transformed crop. The transformation parameters are then obtained by calculating the perspective transformation matrix between the original coordinates p_0, p_1, p_2, p_3 to the new coordinates p_0^t, p_1^t, p_2^t, p_3^t. During pre-training, we linearly increase the probability of applying the spatial transform from 0 to 0.5. For image crops without the spatial transform, the same original crop is used as the reconstruction target.

For pre-training on all backbone models, we stack 3 transformer blocks as the decoder in TAE. The total number of channels in the decoder blocks are the same as output by the encoder, and we set 32 channels per head for the multi-head self-attention layers. For masking at the decoder . We apply two set of random masking of 50% and forward both set of masked tokens through the decoder for reconstruction.

5.3 TAE with Masking at Encoder

As our TAE framework mainly focuses on the decoder, it is orthogonal to operations applied on the encoder. Therefore we can combine TAE with methods applying masking at the encoder. In this work, we experiment with applying masking on Vision Transformers by removing tokens at the beginning of the encoder in the same style as MAE [28]. Specifically, we apply random masking of 60% on the ViT encoders for training together with our TAE framework. For fair comparison with other methods, we follow the standard practice of fine-tuning for 200 epochs and 100 epochs on ViT-S/16 and ViT-B/16 respectively. Results of our TAE compared to other state-of-the-art methods are shown in Table 3. TAE pre-trained ViT-B/16 model obtains a fine-tuning accuracy of 83.4% which is the highest among methods pre-trained for the same number of epochs. Note that CAE [41] achieves a similar accuracy of 83.3% by using extra data during the pre-training phase, while we only use ImageNet-1k data for pre-training.

5.4 Related Works

Self-Supervised Learning. As a representative family of self-supervised learning (SSL), contrastive learning [13, 29–33], *e.g.*, MoCo [29] and SimCLR [30], trains a network to bring the positive pair together, *i.e.* two random crops of the same image, and push the negative pair far away, *i.e.* two crops

Table 3: **Results on ViT**. Performance is evaluated in top-1 accuracy fine-tuned on ImageNet-1k. All methods are pre-trained on ImageNet-1k only, except for BEiT* [36] and CAE* [41] which use pre-trained discrete VAE by DALL-E [42] to generate the discrete tokens as training targets.

Method	Arch.	#Params	Pre-train Epo.	Top-1 Acc(%)
Supervised [43]	ViT-S/16	21M	-	79.9
MoCo-v3 [40]	ViT-S/16	21M	300	81.4
BEiT* [36]	ViT-S/16	21M	300	81.7
CAE* [41]	ViT-S/16	21M	300	81.8
TAE (ours)	ViT-S/16	21M	300	81.7
Supervised [43]	ViT-B/16	85M	-	81.8
MoCo-v3 [40]	ViT-B/16	85M	300	83.0
BEiT* [36]	ViT-B/16	85M	300	83.0
MAE [28]	ViT-B/16	85M	300	82.9
CAE* [41]	ViT-B/16	85M	300	83.3
TAE (ours)	ViT-B/16	85M	300	**83.4**

from different images. BYOL [44] trains a network by only bringing two positives close to simplify the method and also to save memory. Clustering learning [6–13] is another effective line of SSL. It first generates pseudo labels for each sample via clustering similar samples into the same group, and then encourages the crops of the same image to have the same pseudo label. These SSL approaches heavily rely on the alignment of positive samples. Our proposed TAE differs with them in that it depends on transformed image modeling to reconstruct the input image or its semantic feature.

The recently proposed masked image modeling SSL family, *e.g.* MAE [14] and SimMIM [15], feeds a randomly masked input image into an encoder, and then requires a decoder to reconstruct the pixels of the masked patches from the latent representation of the encoder and mask tokens. This mask-reconstruction pretext task is also known as masked image modeling. Given a specific downstream task, this SSL family fine-tunes the pre-trained encoder on the corresponding training data in a supervised manner. Later, to boost performance, MaskFeat [45] and data2vec [18] empirically find better performance by reconstructing the (semantic) feature, *e.g.* the HOG feature [16] or network feature. PeCo [17] replaces Autoencoder with discrete VAE to learn a more semantically perceptual codebook, which also helps learn more semantic features and thus benefits downstream tasks. However, as aforementioned in Sec. 1, for the encoder, the input is masked for pre-training and not masked for fine-tuning, which causes inconsistency in training and thus may impair performance. Except ViT, other widely used encoder architectures, *e.g.* CNNs and MLP-based architectures, is hardly compatible with the masking strategy on the encoder input, which limits wider application of the MAE-like SSL family. In contrast, our proposed TAE is mask-free at the encoder and thus well avoids the above two issues.

Equivariance Learning. Our work is also related to equivariance learning [46–56] in which the feature changes accordingly with the transformation of the input. Hinton *et al.* [46] are the first to emphasize the importance of the equivariant feature, and they proposed a transforming Autoencoder to approximate a given transformation function. The capsule network [47] employs a group of neurons to learn the instantiation parameters of a specific type of an object in a supervised manner and hopes the neurons to know the object and its scaling, rotation and transformation properties. Later, many works [49, 51, 53, 54, 56] follow the capsule network to handle spatial information in the images. The most related work to ours is [50] where Guo *et al.* proposed an affine equivariant Autoencoder that approximates affine transformation via a linear transformation and then encourages the feature to be equivariant to the linear transform. However, Guo *et al.* [50] only handled equivariance on a few specific transformations, which differs from ours that targets at general transformations. Our TAE hopes to build a large-scale unsupervised pre-training framework which is applicable to general network architectures, distinguishing it from these previous works.

5.5 Visualization of transformed reconstruction

To empirically verify the effectiveness of the transformed image reconstruction task, we visualize the reconstruction results on some images as shown in Figure 2. Note that as we used per-patch normalized pixel values [28] as the targets during training, the actual output by TAE are not within the standard color range for images. Therefore we calculate the mean and standard deviation for patches in the ground-truth target to de-normalize the raw output from the model for visualization purpose.

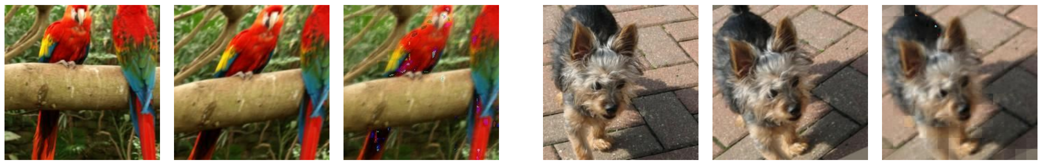


Figure 2: **Visualization of transformed reconstruction by TAE**. Reconstruction results with pre-trained TAE model on ImageNet validation images. For each group of three figures, we shown the original input crop to the autoencoder on the left, the transformed crop as reconstruction target in the middle, and the reconstruction prediction by TAE on the right.