PART: Self-supervised Pretraining with Pairwise Relative Translations

2024 Submission # x

Images are often composed of objects and object parts that are related to each other but are not necessarily related to their *absolute* position in the image frame. For instance, the pose of a person's nose is consistent relative to the forehead, while that same nose can be anywhere in absolute position in the image frame. To capture these underlying relative relationships, we introduce *PART*, a novel pretraining approach that predicts pairwise *relative* translations between randomly sampled input patches. Through this process, the original patch positions are masked out. The pretraining objective is to predict the pairwise translation parameters for any set of patches, just using the patch content. Our object detection experiments on COCO show improved performance over strong baselines such as MAE and DropPos. Our method is competitive on the ImageNet-1k classification benchmark. Beyond vision, we also outperform baselines on 1D time series prediction tasks. The code and models will be available soon.

1 Introduction

Self-supervised learning (SSL) has shown great progress in visual representation learning without relying on expensive labeled data. Many existing SSL methods for images, e.g. MAE [16], Jigsaw [23], MP3 [32], and DropPos [31], extract patches from images using a *grid* structure. MAE [16] masks part of this grid, and the pretext task is to generate the original unmasked image with a reconstruction loss. Other approaches that shuffle or mask patches aim to predict the original position index of the patches. The nature of these tasks imposes *patchifying* images into a grid. However, real-world objects do not naturally align with this rigid grid structure. Thus, we develop a method that learns from randomly sampled patches, moving away from the fixed grid structure.

Random *off-grid* sampling entails that each patch can be at any position in the image, naturally masking the unsampled parts (Figure 4). Due to altering the sampling strategy, we are prompted to reconsider the objective function. Instead of a classification objective as used in absolute position prediction, we propose a regression objective to model the relative relationships between randomly sampled patches solely based on the content of the patches.

We introduce PART: **PA**irwise **R**elative **T**ranslations a pretraining method that predicts *relative* translations between randomly sampled patches. The pretext objective is set up as a regression task to predict the translation $(\Delta x, \Delta y)$ between each pair of patches (Figure 1). We also introduce a novel cross-attention architecture that serves as a projection head.

We empirically show that PART outperforms baselines in object detection and 1D EEG classification and remains competitive for image classification. We also perform ablation studies that compare different sampling strategies and projection head architectures.

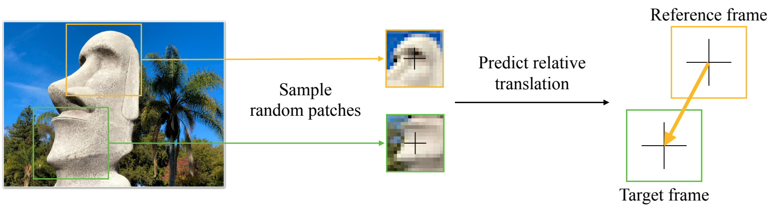

Figure 1: *PART sampling and objective:* a pair of patches are sampled from the image at random positions. Consider the yellow patch as the reference and the green one as the target. The pretext task is learning the underlying translation between any pair of patches, given only the pixel content. The translation maps the reference frame into the target frame.

2 Related Work

Self-supervised learning. SSL techniques are categorized into two main families [31]. The first family are the contrastive learning methods, where different views or representations of the same datapoint are given to one or two parallel models. The objective is maximizing the agreement between the two views [3, 4, 6, 14, 15, 25].

Masked prediction as a pretext task. The second family are the masked prediction methods in which certain information about the input is masked out and the model's task is to either reconstruct the original input or predict the masked-out portion. For instance in natural language processing, BERT [9] proposed training a transformer by solving masked token prediction. In computer vision, some early SSL methods apply degradations to training images, such as decolorization [33], rotation [13], or noise [30] and train models to undo or predict these degradations. In [26] the network is trained to inpaint the contents of a masked image region by understanding the content of the entire image. This group of methods has also been used to pretrain vision transformers [11] and has improved performance in downstream tasks over supervised and constrastive learning baselines. A popular masking method is the MAE work [16], which is based on BeiT [2] where a random subset of the image patches are masked out, and the pretext task involves reconstructing the entire image in pixel space. In I-JEPA [1], the pretext task is given a single context block, predicting the representation of the rest of the image blocks. The methods mentioned so far can be grouped into *generative-based* methods in which the model reconstructs the original input using generative models such as VAEs [18].

Position prediction as a pretext task. Certain challenges arise with generative-based masked prediction, such as longer training time and the increased complexity that the reconstruction task brings with itself [32]. To address these challenges, alternative models have emerged with the pretext task of predicting the ***absolute*** position of the masked patches instead of content reconstruction [31, 32]. In MP3 [32], the corresponding keys to a random set of patches are masked out, whereas in DropPos [31], the position embeddings of a random portion of the image are masked out. The pretext task in both methods is predicting the exact position of each patch, requiring it to solve the puzzle of determining where each patch originated from. The idea behind these methods originates from the [10, 21] and later on the Jigsaw [23, 24] works, where masking is performed by making a puzzle from a part of the image and pretraining a CNN to solve the jigsaw puzzle by predicting the absolute position

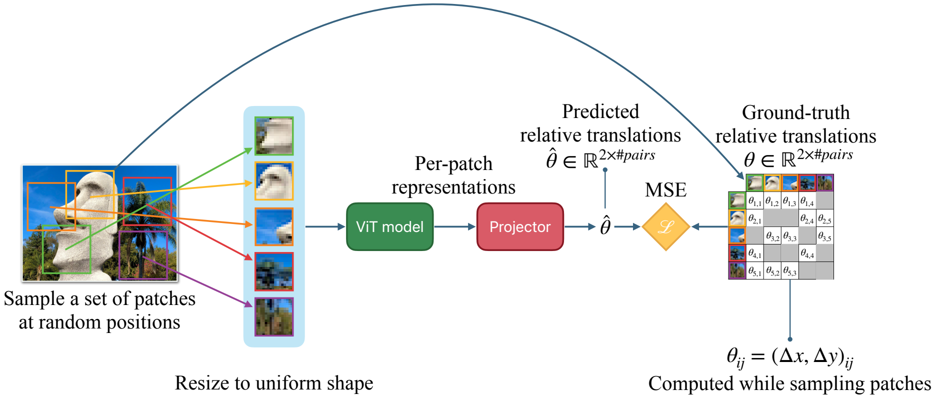

Figure 2: Illustration of PART on 2D image data: We first sample a set of patches from random positions. These patches are chosen randomly for each image at each iteration. Then, all patches are resized to a uniform patch size and given to the ViT model. The ViT model predicts a representation for each patch. The cross-attention projector returns a $\hat{\theta}_{ij}$ for each pair of (reference, target) patch (i, j). $\hat{\theta}_{ij}$ is the relative translation that converts reference frame i to target frame j.

of each piece. DILEMMA [28] enforces predicting the position of patches that have been artificially misplaced. In [5], the pretext task is the absolute position prediction of a random portion of the image given the input image as a reference. While vision transformers typically exhibit insensitivity to the input tokens order [1, 22], leading to the hypothesis that they tend to model the relationship between a set of unordered input tokens, the above-mentioned models focus explicitly on absolute position awareness. In contrast, PART is trained on ***relative*** translations between random input patches.

Relative information as a pretext task. The notion of relative information has been used in self-supervised learning in various tasks and domains. In graph representation learning, [27] proposed predicting the local relative contextual position of one node to another. For single image depth estimation, [17] proposed estimating the relative depth using the motion in the video. For object detection, [34] proposed a self-supervised spatial context learning module that learns the internal object structure by predicting the relative positions within the extent of that object. The above-mentioned methods learn with respect to one reference frame. In contrast, PART learns the relative information of any reference frame to any target frame.

3 Pairwise Relative Translations

3.1 Random Off-grid Sampling

Given an image $I \in \mathbb{R}^{H\times W\times C}$, we extract N random patches from the image. With (x_s, y_s) as the coordinates of the top left corner of the frame and $(x_s + D, y_S + D)$ as the coordinates of the bottom right corner of the frame, respectively. These patches are of shape $D \times D$ and are in random positions of the image. H and W are the height and width of the image, and C is the number of channels. P is the patch size, and $N = \frac{H\times W}{P^2}$ is the number of patches. Each sampled patch of shape $D \times D$ is then resized to $P \times P$ with C channels. Now we have N samples of $P \times P \times C$ that can be reshaped into the original image size $\hat{I} \in \mathbb{R}^{H \times W \times C}$. This reshaped image would have looked like a puzzled version of the original image if the random samples were on-grid and with a $P \times P$ shape. During random sampling, parts of the image are masked out. Also, some information about each patch's spatial frequency is masked by resizing all samples to the patch size. The pretext task is set up such that the ViT model consumes images with incomplete information.

The reshaped patches $\hat{I}$ are then given to the ViT model. In the ViT model, $\hat{I}$ is reshaped into a sequence of patches $I_p \in \mathbb{R}^{N \times (P \times P \times C)}$. A linear projection is then applied to I_p, mapping it to d dimensions to get patch embeddings $X \in \mathbb{R}^{N \times d}$. Also, a [CLS] token $x_{CLS} \in \mathbb{R}^D$ is used to aggregate the information. Following [32], $[x_{CLS}; x]$ are given as an input to the transformer blocks without the position embeddings. The ViT model returns the learned patch embeddings $X' \in \mathbb{R}^{N \times d}$.

3.2 Relative Translation Parameterization

A pair of patches (reference, target) are sampled from the image at random positions with $(x_{\text{ref}}, y_{\text{ref}})$ and $(x_{\text{tgt}}, y_{\text{tgt}})$ as the center pixel coordinates of the two patches in image space. The two patches are then resized to a uniform patch size P, masking their original position in the image space, as well as their pixel content. The goal is to learn the underlying translation between any pair of patches. The translation converts the reference coordinate frame into the target coordinate frame with the width w_{ref} and height h_{ref} of the reference patch. The task is to predict

$$\theta_{\text{ref,tgt}} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}_{\text{ref,tgt}} = \begin{bmatrix} (x_{\text{tgt}} - x_{\text{ref}})/w_{\text{ref}} \\ (y_{\text{tgt}} - y_{\text{ref}})/h_{\text{ref}} \end{bmatrix}_{\text{ref,tgt}} \tag{1}$$

with Δx and Δy capturing *relative position*. In simple terms, the goal is to move the reference frame so that it translates into the target frame. In this context, when referring to a "frame", we specify the bounding box itself rather than the actual pixel contents in the bounding box (Figure 1).

The emphasis on predicting the relative translation is crucial because information about the pixel space is lost after resizing to a uniform patch size. Here, the model no longer possesses details about the original image space. Thus, the two terms we seek to predict are the translation in x normalized by the width of the reference frame w_{ref} and the translation in y normalized by the height of the reference frame h_{ref}.

3.3 Cross-attention Projection Head

The ViT model outputs a per-patch representation $X' \in \mathbb{R}^{N \times d}$. The projection head maps the per-patch representations to the relative translations between a random number of patch pairs (#*pairs*), resulting in $\hat{\theta} \in \mathbb{R}^{2 \times \#pairs}$. The two outputs per patch pair are the relative positions between the reference and target patches.

Given X', this module selects random index pairs (#*pairs*) of patches $S \in \mathbb{N}^{2 \times \#pairs}$ with S_0 as the index of the reference patch and S_1 as the index of the target patch. The representations of reference patches S_0 and S_1 are then concatenated:

$$\hat{X} = \texttt{concat}(X'_{S_0}, X'_{S_1}) \tag{2}$$

$\hat{X}$ goes through a linear projection to convert from $\mathbb{R}^{\#pairs \times 2 \times d}$ to $\mathbb{R}^{\#pairs \times d}$. $\hat{X}$ is fed into a cross-attention module [29] as the query, and X' is fed as both the key and the value.

$$\hat{\theta} = \texttt{cross\_attention}(Q = \hat{X}, K = X', V = X') \tag{3}$$

The cross-attention module allows for information dissemination between all patch representations and enables the model to focus on predicting the relative translation only for a subset S of patch pairs. This imposes further masking of information given to the model. θ is only calculated for the subset S of patch pairs. The model is trained with a mean squared error loss between the predicted relative translations $\hat{\theta}$ and the ground-truth relative translations θ.

3.4 Supervised Finetuning

After self-supervised pretraining, we finetune the network end-to-end using labeled data in a supervised setup. The model is initialized with the learned weights from pretraining. Following the standard ViT configuration, we eliminate the projection head and substitute it with a linear classification or detection head. Unlike the pretraining phase, where no positional embedding is trained, we incorporate randomly initialized learnable position embeddings into the patch embeddings in this stage. Additionally, instead of the random sampling and masking in the pretraining phase, we perform fixed grid sampling when finetuning.

4 Experiments

In the vision domain, we experiment with a medium-sized classification dataset, CIFAR-100 [19], ImageNet-1K [8]. We report accuracy, euclidean distance error, and the mean squared error in x and y dimensions. We finetune with COCO [20] on models pretrained on ImageNet-1K for detection. In the 1D signal domain, we experiment with single-channel electroencephalography (EEG) signals extracted from the PhysioNet 2018 "You Snooze You Win" Challenge Dataset [12]. We report on our method and a grid sampling variant of PART (PART-grid) for all experiments. Implementation details are in the Supplementary Material.

4.1 Object Detection

Table 1 compares PART with MAE [16], the recent MP3 [32] and DropPos [31] pretraining methods in the downstream object detection performance. It shows that the random sampling of patch positions in PART pretraining benefits the detection task, which is sensitive to location information compared with the PART-grid. On the other hand, DropPos, MP3, and PART-grid all sample patches from a fixed position grid and perform worse than PART in this task.

4.2 Image Classification

In Table 2, we compare PART with supervised and state-of-the-art SSL alternatives on the ImageNet-1K [8] classification benchmark. Our method outperforms the supervised results as well as MP3 [32] and shows competitive performance with respect to DropPos [16] and MAE [16] that use position embedding during pretraining. DropPos employs extra position smoothing and attentive reconstruction techniques that could further accelerate training.

Table 1: COCO detection performance after finetuning for 1× schedule (12 epochs, or 90k iterations). All methods use Mask R-CNN with ViTB/16 as the backbone.

	AP^b	AP^b_{50}	AP^b_{75}
MAE(from DropPos)	40.1	60.5	44.1
MP3	41.8	61.4	46.0
DropPos	42.1	62.0	46.4
PART-grid	41.4	60.8	45.5
PART	**42.4**	**62.5**	**46.8**

Table 2: ImageNet-1k classification with ViT-B as backbone. Pos Embed indicates using position embedding. PT and FT are the number of pretraining and finetuning epochs.

	Pos Embed	PT	FT	Accuracy
Supervised	✓	0	300	81.8
Supervised		0	300	79.1
MP3		400	300	82.59
MAE	✓	150	150	82.7
DropPos	✓	200	100	**83**
PART-grid		400	300	82.43
PART		400	300	<u>82.66</u>

4.3 1D Time Series Classification

PART can also be used to model 1D time-series data by predicting relative time shifts between patches sampled from a longer sequence. To test this approach, we pretrained a ViT on biosignals from the PhysioNet 2018 "You Snooze You Win" Challenge Dataset [12]. Our method improves performance over supervised and self-supervised baseline (Table 3).

Table 3: Sleep stage classification accuracy represented using Cohen's Kappa. PT and FT are the number of pretraining and finetuning epochs.

	PT	FT	Cohen's Kappa
Supervised w/ Pos Embed	0	200	0.431
MP3	500	200	0.508
DropPos	500	200	0.522
PART-grid	500	200	0.500
PART	500	200	**0.557**

4.4 Ablation Studies

Sampling strategies. An essential component of our method is the patch sampling process. Besides random sampling, we ablate on on-grid sampling similar to MP3 and DropPos (Figure 3). In the grid sampling, all patches are arranged in a grid form, with a fixed size at fixed positions. PART-grid has a similar patch sampling to MP3 but with a relative objective function. The results in Tables 1, 2, and 3 suggest that random sampling improves performance in different tasks and domains compared to PART-grid, while introducing more masking.

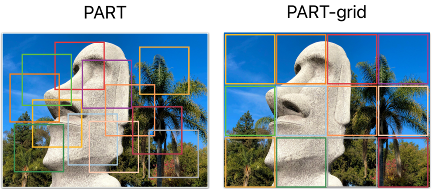

Figure 3: PART adopts a random sampling strategy. Grid sampling strategy (PART-grid) is performed as an ablation.

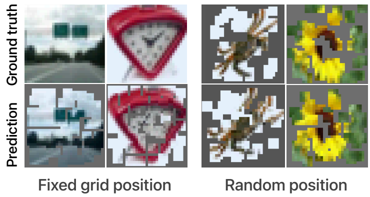

Figure 4: Reconstructing the input image given the predicted relative translations.

Projection head. Besides the cross-attention projection head in the method, we perform an ablation study on two other ways to learn this mapping. The most straightforward approach is a fully connected MLP that receives all patches concatenated as an input and predicts the translation for any two patches. So, given N patches with d dimensions, the projection head would have $N*d*N^2*2$ parameters. Although the weights are not shared in this approach like in the cross-attention head, the projection head can access all patch representations. This helps the model to converge faster because it can use extra information from other patches. However, the classification head will replace the projection head during finetuning. The time spent on training the fully connected MLP can be spent on training better representations instead. We propose an alternative projection head that compensates for the high parameter count in the fully connected MLP approach through weight sharing, which we term a pairwise MLP. The pairwise MLP receives two concatenated patches as its input and predicts their relative translation. Although this approach uses only $2*d*2$ parameters, the projection head does not have access to all the patches, thus predicting the translations solely based on the content of these two patches. Table 4 shows the results for different projection heads. The results suggest that the cross-attention head (83%) outperforms pairwise MLP (82.52%) and MLP (82.38%). MLP is computationally more expensive than pairwise MLP and cross-attention.

Table 4: Ablation on different projection heads for CIFAR-100 pretrained for 1000 epochs.

	MSE x	MSE y	Euclidean error	Accuracy
PART MLP	3.18	2.02	1.68	82.38
PART pairwise MLP	2.84	1.76	1.59	82.52
PART Cross-attention	1.14	0.77	0.81	83

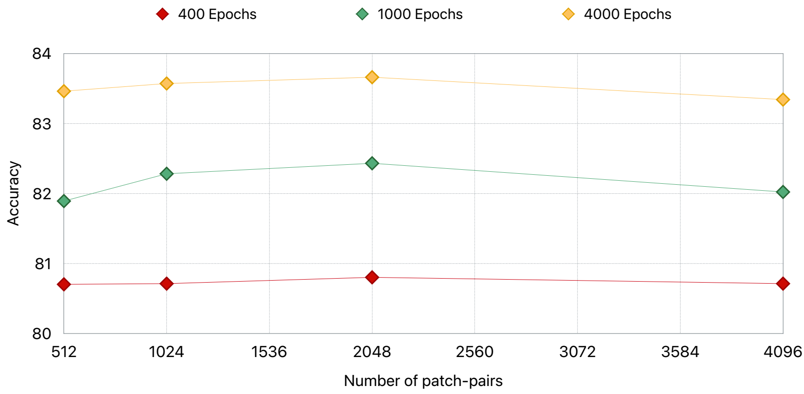

Figure 5: Ablation on the #pairs for CIFAR-100 with 400, 1000, 4000 pretraining epochs.

Number of patch pairs. As explained in Section 3.3, a subset S is randomly chosen from the patch representations. *#pairs* is the parameter that determines the length of S. We study the effect of *#pairs* in Figure 5 after 400, 1000, and 4000 epochs of pretraining. We observe that curves follow similar patterns for different epochs of pretraining, while more pretraining epochs result in higher accuracy. We also observe a trade-off in *#pairs*. Higher *#pairs* means the model sees more patch information but also needs to predict the relative translations for more contradicting patch pairs. Whereas smaller *#pairs* means the model has access to less information, thus overfitting on the task leading to less general representations. There is a sweet spot with 2048 patch pairs, where enough global patch information is given to the model, and the training task is neither easy nor difficult.

4.5 Qualitative Analysis

Reconstructions. Figure 4 illustrates the ground truth in the first row and the predictions for different sampling strategies in the second row. These images are generated by fixing one random reference patch and positioning all other patches relative to that patch. In the first row, other patches are positioned based on the ground truth relative position. In the second row, other patches are positioned based on the model's output relative position. In the grid sampling strategy, the ground truth relative positions reconstruct the full image because, in this sampling, the patches that form the grid cover the whole image. The model's prediction almost matches the ground truth, even in small details. The model has learned the general structure of the scenes. For instance, the sky is on top, and the road is at the bottom of the image. It has also learned the triangular structure of the clock. However, some details are missing, such as the hands and the numbers on the clock. It also has difficulties placing mono-color patches because the model only sees the pixel content of the patches. In PART, the ground truth patch visualization includes only a subset of the patches, thus providing a masked input to the model.

Patch uncertainty. We visualize patch uncertainty as a byproduct of our method to check whether different reference patches agree with each other relative to a single target patch. Our model predicts the relative translation for both cases where patch i is a reference patch and a target patch. We visualize Figure 6 by fixing one reference patch and positioning all other

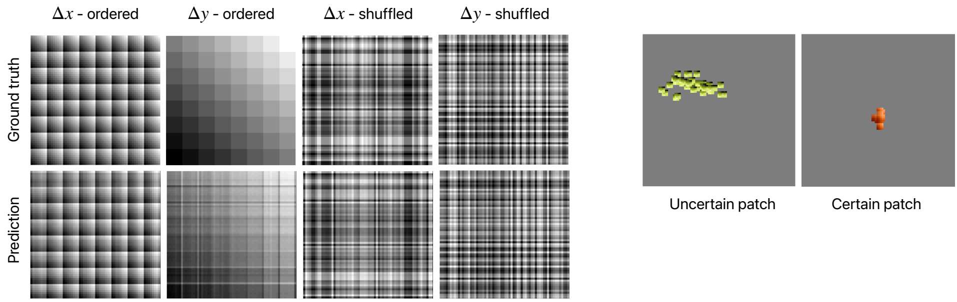

Figure 6: Left: The ground truth and the output prediction matrices for translations in x and y axis for ordered versus unordered patch indices. The ordered matrices are sorted based on patch positions from the top-left side of the image to the bottom down. The matrices are of size $N \times N$, where element i, j is the relative translation between the reference patch i and target patch j. The bright colors are positive numbers, the dark colors are negative numbers, and the grey color is 0. The color intensity shows the magnitude. Right: We fix a single target patch and place that patch relative to all reference patches. If the model is certain, it will always place the patch at the same location.

patches with respect to that patch. Here, we fix one target patch and place that patch relative to all other reference patches. If all patches are placed at the same location, all reference patches agree, thus depicting a more certain patch. Patch uncertainty comes as a byproduct of our method.

Ground truth vs. prediction. Figure 6 shows the final prediction matrix of the model versus the ground truth matrix. We can see that the ground truth matrix matches the model prediction for translation in both x and y axes. The most prominent property that emerges from this figure is the negative symmetry. The negative symmetry is an indication that the model learns that given two patches P_i and P_j with P_i as the reference patch, the model predicts Δx and Δy. Whereas, with P_j as the reference patch, the model predicts $-\Delta x$ and $-\Delta y$, meaning that even considering the heavy masking and no global patch information, the model positions two patches correctly relative to each other.

5 Conclusion

In this work, we introduce PART, a pretraining method that predicts pairwise relative translations between input patches. By employing a random off-grid sampling strategy and relative coordinate prediction as a pretext task, PART aims to model the relative spatial relationships of objects. Our experiments span 2D and 1D data, where PART's application indicates a positive impact. Upon finetuning on various downstream tasks such as object detection, image classification, and time series classification, PART has shown promising results compared with existing supervised and self-supervised baselines methods. Future work could extend the application of PART to more diverse datasets and tasks, further refining its capabilities and understanding its full potential.

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023.

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2021.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[5] Mathilde Caron, Neil Houlsby, and Cordelia Schmid. Location-aware self-supervised transformers for semantic segmentation. In *WACV*, 2024.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[7] X Chen, S Xie, and K He. An empirical study of training self-supervised vision transformers. In *ICCV*.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

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

[10] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[11] 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 *ICLR*, 2021.

[12] Mohammad M Ghassemi, Benjamin E Moody, H Lehman, Li-wei, Christopher Song, Qiao Li, Haoqi Sun, Roger G Mark, M Brandon Westover, and Gari D Clifford. You snooze, you win: the physionet/computing in cardiology challenge 2018. In *IEEE Computing in Cardiology Conference (CinC)*, 2018.

[13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 2020.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

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

[17] Huaizu Jiang, Gustav Larsson, Michael Maire Greg Shakhnarovich, and Erik Learned-Miller. Self-supervised relative depth learning for urban scene understanding. In *ECCV*, 2018.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[21] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *CVPR*, 2018.

[22] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *NeurIPS*, 2021.

[23] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[24] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 2018.

[25] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[27] Zhen Peng, Yixiang Dong, Minnan Luo, Xiao-Ming Wu, and Qinghua Zheng. Self-supervised graph representation learning via global context prediction. *arXiv preprint arXiv:2003.01604*, 2020.

[28] Sepehr Sameni, Simon Jenni, and Paolo Favaro. Representation learning by detecting incorrect location embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[30] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[31] Haochen Wang, Junsong Fan, Yuxi Wang, Kaiyou Song, Tong Wang, and Zhaoxiang Zhang. Droppos: Pre-training vision transformers by reconstructing dropped positions. In *NeurIPS*, 2023.

[32] Shuangfei Zhai, Navdeep Jaitly, Jason Ramapuram, Dan Busbridge, Tatiana Likhomanenko, Joseph Y Cheng, Walter Talbott, Chen Huang, Hanlin Goh, and Joshua M Susskind. Position prediction as an effective pretraining strategy. In *ICML*, 2022.

[33] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[34] Mohan Zhou, Yalong Bai, Wei Zhang, Tiejun Zhao, and Tao Mei. Look-into-object: Self-supervised structure modeling for object recognition. In *CVPR*, 2020.