Masked Jigsaw Puzzle: A Versatile Position Embedding for Vision Transformers

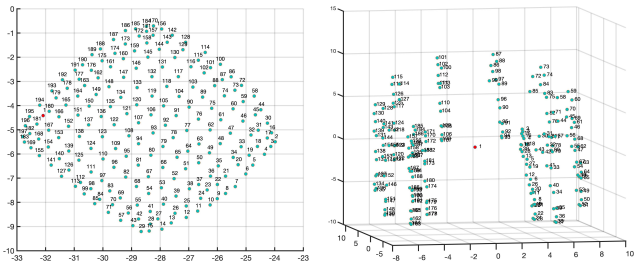
Bin Ren[1,2*] Yahui Liu[2*] Yue Song[2] Wei Bi[3] Rita Cucchiara[4] Nicu Sebe[2] Wei Wang[5†]
[1]University of Pisa, Italy [2]University of Trento, Italy
[3]Tencent AI Lab , China [5]Beijing Jiaotong University, China
[4]University of Modena and Reggio Emilia, Italy

Abstract

Position Embeddings (PEs), an arguably indispensable component in Vision Transformers (ViTs), have been shown to improve the performance of ViTs on many vision tasks. However, PEs have a potentially high risk of privacy leakage since the spatial information of the input patches is exposed. This caveat naturally raises a series of interesting questions about the impact of PEs on accuracy, privacy, prediction consistency, etc. *To tackle these issues, we propose a Masked Jigsaw Puzzle (MJP) position embedding method. In particular, MJP first shuffles the selected patches via our block-wise random jigsaw puzzle shuffle algorithm, and their corresponding PEs are occluded. Meanwhile, for the non-occluded patches, the PEs remain the original ones but their spatial relation is strengthened via our dense absolute localization regressor. The experimental results reveal that 1) PEs explicitly encode the 2D spatial relationship and lead to severe privacy leakage problems under gradient inversion attack; 2) Training ViTs with the naively shuffled patches can alleviate the problem, but it harms the accuracy; 3) Under a certain shuffle ratio, the proposed MJP not only boosts the performance and robustness on large-scale datasets (*i.e.*, ImageNet-1K and ImageNet-C, -A/O) but also improves the privacy preservation ability under typical gradient attacks by a large margin. The source code and trained models are available at* https://github.com/yhlleo/MJP.

1. Introduction

Transformers [38] demonstrated their overwhelming power on a broad range of language tasks (*e.g.*, text classification, machine translation, or question answering [22, 38]), and the vision community follows it closely and extends it for vision tasks, such as image classification [7, 37], object detection [2, 51], segmentation [47], and image generation [3, 24]. Most of the previous ViT-based methods focus on designing different pre-training objectives [9, 11, 12] or variants of self-attention mechanisms [28, 39, 40]. By contrast, PEs receive less attention from the research community and have not been well studied yet. In fact, apart from the attention mechanism, how to embed the position information into the self-attention mechanism is also one indispensable research topic in Transformer. It has been demonstrated that without the PEs, the pure language Transformer encoders (*e.g.*, BERT [6] and RoBERTa [25]) may not well capture the meaning of positions [43]. As a consequence, the meaning of a sentence can not be well represented [8]. Similar phenomenon of PEs could also be observed in vision community. Dosovitskiy *et al.* [7] reveals that removing PEs causes performance degradation. Moreover, Lu *et al.* [29] analyzed this issue from the perspective of user privacy and demonstrated that the PEs place the model at severe privacy risk since it leaks the clues of reconstructing sequential patches back to images. Hence, it is very interesting and necessary to understand how the PEs affect the accuracy, privacy, and consistency in vision tasks. Here the consistency means whether the predictions of the transformed/shuffled image are consistent with the ones of the original image.

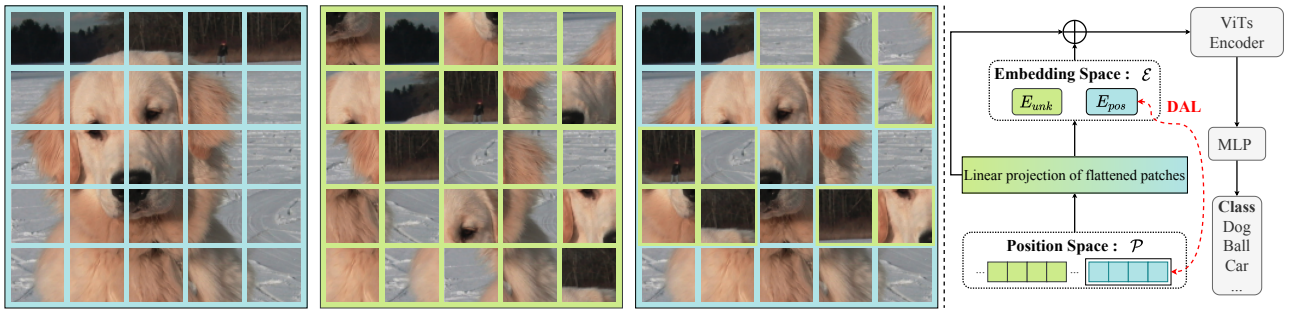


(a) UMAP (b) PCA

Figure 1. Low-dimensional projection of position embeddings from DeiT-S [37]. (a) The 2D UMAP projection, it shows that reverse diagonal indices have the same order as the input patch positions. (b) The 3D PCA projection, it also shows that the position information is well captured with PEs. Note that the embedding of index 1 (*highlighted in red*) corresponds to the [CLS] embedding that does not embed any positional information.

To study the aforementioned effects of PEs, the key is to figure out what explicitly PEs learn about positions from input patches. To answer this question, we project the high-dimensional PEs into the 2D and 3D spaces using Uni-

*Equal contribution. Email: {bin.ren, yahui.liu}@unitn.it
†Corresponding author. Email: wei.wang@bjtu.edu.cn



(a) $\mathbf{x}$ (b) $\widetilde{\mathbf{x}}$ $(\gamma = 1)$ (c) $\widetilde{\mathbf{x}}$ $(\gamma = 0.4)$ (d) Overview of our MJP

Figure 2. (a) The original input patches; (b) Totally random shuffled input patches; (c) Partially random shuffled input patches; (d) An overview of the proposed MJP. Note that we show the random shuffled patches and its corresponding *unknow* position embedding in green and the rest part in blue. DAL means the self-supervised *dense absolute localization* regression constraint.

form Manifold Approximation & Projection (UMAP) [30] and PCA, respectively. Then for the first time, we visually demonstrate that the PEs can learn the 2D spatial relationship very well from the input image patches (the relation is visualized in Fig. 1). We can see that the PEs are distributed in the same order as the input patch positions. Therefore, we can easily obtain the actual spatial position of the input patches by analyzing the PEs. Now it explains why PEs can bring the performance gain for ViTs [7]. This is because the spatial relation in ViTs works similar as the inherent intrinsic inductive bias in CNNs (*i.e.*, it models the local visual structure) [45]. However, these correctly learned spatial relations are unfortunately the exact key factor resulting in the privacy leakage [29].

Based on these observations, one straightforward idea to protect the user privacy is to provide ViTs with the randomly transformed (*i.e.*, shuffled) input data. The underlying intuition is that the original correct spatial relation within input patches will be violated via such a transformation. Therefore, we transform the previous visually recognizable input image patches $\mathbf{x}$ shown in Fig. 2(a) to its unrecognizable counterpart $\widetilde{\mathbf{x}}$ depicted in Fig. 2(b) during training. The experimental results show that such a strategy can effectively alleviate the privacy leakage problem. This is reasonable since the reconstruction of the original input data during the attack is misled by the incorrect spatial relation. However, the side-effect is that this leads to a severe accuracy drop.

Meanwhile, we noticed that such a naive transformation strategy actually boosts the **consistency** [31, 34, 44] albeit the accuracy drops. Note that here the consistency represents if the predictions of the original and transformed (*i.e.*, shuffled) images are consistent. Given the original input patches $\mathbf{x}$ and its corresponding transformed (*i.e.*, shuffled) counterpart, we say that the predictions are consistent if $\arg\max P(\mathcal{F}(\mathbf{x})) = \arg\max P(\mathcal{F}(\widetilde{\mathbf{x}}))$, where $\mathcal{F}$ refers to the ViT models, and P denotes the predicted logits.

These observations hint that there might be a trade-off solution that makes ViTs take the best from both worlds (*i.e.*, both the accuracy and the consistency). Hence, we propose the Masked Jigsaw Puzzle (MJP) position embedding method. Specifically, there are four core procedures in the MJP: (1) We first utilize a block-wise masking method [1] to randomly select a partial of the input sequential patches; (2) Next, we apply jigsaw puzzle to the selected patches (*i.e.*, shuffle the orders); (3) After that, we use a shared *unknown* position embedding for the shuffled patches instead of using their original PEs; (4) To well maintain the position prior of the unshuffled patches, we introduce a *dense absolute localization* (DAL) regressor to strengthen their spatial relationship in a self-supervised manner. We simply demonstrate the idea of the first two procedures in Fig. 2(c), and an overview of the proposed MJP method is available in Fig. 2(d). In summary, our main contributions are:

- We demonstrate that although PEs can boost the accuracy, the consistency against image patch shuffling is harmed. Therefore, we argue that studying PEs is a valuable research topic for the community.
- We propose a simple yet efficient Masked Jigsaw Puzzle (MJP) position embedding method which is able to find a balance among accuracy, privacy, and consistency.
- Extensive experimental results show that MJP boosts the accuracy on regular large-scale datasets (*e.g.*, ImageNet-1K [32]) and the robustness largely on ImageNet-C [18], -A/O [19]. One additional bonus of MJP is that it can improve the privacy preservation ability under typical gradient attacks by a large margin.

2. Related Work

2.1. Vision Transformers

Transformers [38], originally designed for Nature Language Processing (NLP) tasks, have recently shown promising performance on computer vision tasks [13, 22]. Benefiting from the strong representation power of modelling global relations between image patches, Vision Transformers (ViTs) [7] have achieved superior performance than their counterpart CNNs on image classification and various other

downstream tasks (*e.g.*, object detection [2, 51], object re-identification [17], dense prediction [41,42,46,50] and image generation [3,4,20,24]).

As a core module in ViTs, multi-head self-attentions (MSAs) [7, 38] aggregate sequential tokens with normalized attentions as: $\boldsymbol{z}_j = \sum_i \texttt{Softmax}(\frac{\boldsymbol{QK}}{\sqrt{d}})_i \boldsymbol{V}_{i,j}$ where $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ are query, key and value matrices, respectively. d is the dimension of query and key, and $\boldsymbol{z}_j$ is the j-th output token. In theory, when the position information is not considered, the outputs of MSAs should be strictly invariant to the input sequence order (*i.e.*, *position-insensitivity*). This indicates that a visually recognizable image can be transformed into its unrecognizable counterpart by permuting the order of image patches while maintaining the performance delivered by ViTs compared with the ViTs trained on the original non-permuted image. However, the usage of PEs hinders such implementations, where the outputs of the ViTs vary dramatically with the mentioned naive transformations shown in Fig. 2(b). In this work, our main focus is to explicitly figure out what PEs actually learn from input patches about positions, and how the PEs affect the accuracy, privacy, and the consistency properties of ViTs.

2.2. Position Embeddings

In Transformer networks, both the attention and the (individual token based) feed-forward layers are permutation invariant when the position information is not considered. In this way, the spatial relationships between image patches could not be modeled as the position information is completely discarded. As compensation, PEs are naturally introduced into ViTs to provide information about the token order during the learning process, since it offers possibilities for dependency modeling between elements at different positions. For example, previous works [10,33,38] indicated that PEs are useful to give the model a sense of which portion of the sequence in the input/output it is currently dealing with. Inspired by this, some works [15,21,23,27,35] showed diverse application scenarios that benefit from the usage of suitable PEs. In addition, Chu *et al.* [5] proposed correlating the PEs with their local neighborhood of the input sequence. Liu *et al.* [26] proposed to enhance the spatial prior (*i.e.*, relative localization) in the final content embedding to indirectly enrich the inductive bias. Obviously, although these methods enhance the position information learnt by PEs, they indeed degenerate the position-insensitive property of MSAs.

Especially, Wang *et al.* [43] revealed that Transformer encoders (*e.g.*, BERT [6] and RoBERTa [25]) may not well capture the meaning of positions (absolute and relative positions). They showed that Transformer encoders learn the local position information that can only be effective in masked language modeling. In contrast, there does not exist such a similar "masked language modeling" procedure in ViTs (and VTs) in the typically supervised pre-training. Moreover, Lu *et al.* [29] revealed that the learnable PEs place the model at severe privacy risk, which leaks the clues of reconstructing sequential patches to images. In this paper, we dive into the usage of PEs and propose an efficient position embedding method, MJP, to improve the position-insensitive property of ViTs without hurting the positive effects of PEs.

3. Preliminaries: 2D/3D Spatial Priors in PEs

Although numerous works claim that the PEs can learn the 2-D spatial relationship of image patches, this claim has not been demonstrated visually or mathematically. To visualize the concrete relation of image patches captured in the high-dimensional position embedding, we project them into the 2-D and 3-D spaces using Uniform Manifold Approximation and Projection (UMAP) [30] and PCA, respectively. Fig. 1 displays the projections of PEs from DeiT-S [37]. Note that [7] only shows the cosine similarity between the PEs, which is not exactly the 2D spatial information mentioned in our paper. Here, we explicitly specify the spatial relationship as the relative distance in the 2D/3D coordinate space. As shown in Fig. 1, the spatial relationship can be projected into a 2D/3D coordinate system and indicate the localizations of these embeddings. For the UMAP, the projected positional embedding emerges as grid-like structures with the distances between each point roughly the same, which is coherent with the relation of embedded image patches. For the PCA, the position embedding that corresponds to neighboring image patches groups together and aggregates to the stick-like form. This phenomenon demonstrates that the spatial relationship is indeed learned by the PEs. Moreover, this also implies that *the spatial relationship learned in the high-dimensional space still manifests in the low-dimensional space*.

The learnable PEs, which work as a lookup table of a dictionary, maps the 1-dimensional data into the sparse high-dimensional space. By nature, it is a sparse large matrix. Dimensionality reduction techniques can capture the structural information of sparse matrices using low dimensionalities. For the PCA, the amount of information retained in the projection can be measured by the ratio of explained variance. Formally, we have the definition as follows:

Definition 1 (Explained Variance) *Let* $\mathbf{P}$ *and* $\mathbf{V}$ *denote the data matrix and the PCA projection matrix. The ratio of explained variance that* $\mathbf{PV}$ *accounts for is defined as* $\sum \sigma(\mathbf{PV})^2 / \sum \sigma(\mathbf{P})^2$ *where* $\sigma(\cdot)$ *denotes the singular value.*

In practice, we observe that the 3-dimensional PCA projection explains 54.6% of the total variance of the DeiT-S embedding matrix. Given that a large amount of information can be captured by the low-dimensional projection, we propose that explicitly enforcing low-dimensional positional prior can help the positional learning in the high-dimensional space, which might accelerate the convergence rate of training and improve the performance (See Sec. 4.2).

Algorithm 1 Block-wise Random Jigsaw Puzzle Shuffle

Input: Input image: $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$;
Shuffle Ratio: γ;
Patch Size: P
Output: Shuffled image patches: $\widetilde{\boldsymbol{x}}_p$

$\mathbf{x}_p \in \mathbb{R}^{N\times(P^2\cdot C)} \longleftarrow Patchlize(\mathbf{x}, P)$
$\mathbf{m} \in \mathbb{R}^{\frac{H}{P}\times\frac{W}{P}} \longleftarrow BinaryInitialize(\mathbf{x}_p, 0)$
$\widetilde{\mathbf{m}} \in \mathbb{R}^{\frac{H}{P}\times\frac{W}{P}} \longleftarrow BlockwiseMask(\mathbf{m}, \gamma)$ [25]
$\widetilde{\boldsymbol{x}}_p \in \mathbb{R}^{N\times(P^2\cdot C)} \longleftarrow JigsawPuzzle(\mathbf{x}_p, \widetilde{\mathbf{m}})$
return $\widetilde{\boldsymbol{x}}_p$

4. Method

Based on the design principle of MSAs, the outputs of MSAs should be entirely position-agnostic. However, PEs hinder such a property because it can learn the strong 2-D spatial relation of the input patches (as we demonstrated in Sec. 3). To this end, we propose the block-wise random jigsaw puzzle shuffle algorithm (See Alg. 1) to transform the input patches with different shuffle ratios γ for intermingling the original correct spatial relation.

Since we experimentally demonstrate that the totally shuffled strategy (*i.e.*, $\gamma = 1.0$) will degenerate the accuracy a lot albeit the consistency increases. As a remedy, we only shuffle portion of the sequence patches, and *strengthen* the spatial relation of the rest part with a dense absolute localization regression. Finally, a versatile position embedding method MJP is proposed. The detailed analysis of each module is available in the ablation study in Sec. 5.3. In the following sub-sections, we will introduce the Jigsaw Puzzle Shuffle method (Sec. 4.1), the spatial relation strengthen method (Sec. 4.2), and our final MJP method (Sec. 4.3).

4.1. Block-wise Random Jigsaw Puzzle Shuffle

Specifically, given an input image $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$, we first reshape it into a sequence of flattened 2D patches $\mathbf{x}_p \in \mathbb{R}^{N\times(P^2\cdot C)}$, where (H, W) is the resolution of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and then we have $N = HW/P^2$, which denotes the number of patches. Then instead of directly applying block-wise masking method [1] to the image, we first initialize a binary mask matrix $\mathbf{m} \in \mathbb{R}^{\frac{H}{P}\times\frac{W}{P}}$ with the same size as the image patches in $\mathbf{x}_p$. Next, we use [1] to update the binary mask $\mathbf{m}$ in which the masked positions will be set to 1 and the rest untouched positions remains 0. The hyper-parameter γ is used to control the ratio of selected positions. After that, a jigsaw puzzle shuffle operation is applied to $\mathbf{x}_p$ conditioned on the updated binary mask $\widetilde{\mathbf{m}}$.

Finally, we get the shuffled patch sequence $\widetilde{\mathbf{x}}_p$, where the shuffled patches are actually $\{\mathbf{x}_p^i | \widetilde{\mathbf{m}}_i = 1\}$. Notably, in [1], the patches covered by the sampled mask are not visible to the encoder module, while in our method, the masking strategy [1] is only used to mask out the positions/indices of the selected patches. These patches are still visible to the encoder module and they are randomly shuffled in a jigsaw puzzle manner. One intuitive shuffled toy example is shown in Fig. 2(c), where the green part means the selected shuffled region while the blue part remains unchanged. Actually, this algorithm shares a similar idea with the data augmentation method as the training images are always changing as their patches are randomly shuffled during each iteration.

4.2. Strengthening Spatial Prior in PEs

Liu *et al.* [26] noticed that by *enhancing* the 2-D spatial information of the output embeddings of the last layer of ViTs, the training convergence speed can be accelerated. Inspired by their work, we observe that similar gains can be obtained by applying a low-dimensional spatial prior in PEs. Different from the dense relative localization constraint in [26] which samples relative pairs from the whole output sequential embeddings, we propose a much simpler *dense absolute localization* (DAL) regression method to directly use self-supervised absolute location to enhance the spatial information in PEs, which avoids sampling relative pairs.

Since the PEs capture the absolute position of the input patches, to some extent, the position information could be reconstructed via a reversed mapping function $g(\cdot) : \mathcal{E} \to \mathcal{P}$, where $\mathcal{E}$ and $\mathcal{P}$ are embedding space and position space, respectively. Given that PEs have one-to-one correspondence with the sequential image patches, we can reshape them into $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{K\times K\times D}$, where K refers to height/weight of the grid and D refers to latent vector size. Then we can compute the reverse mapping from $\mathcal{E} \to \mathcal{P}$ via:

$$(\widetilde{i}, \widetilde{j})^T = g(\mathbf{E}^{i,j}_{\text{pos}}), \tag{1}$$

in which $(\widetilde{i}, \widetilde{j})^T$ is the predicted patch position, and $\mathbf{E}^{i,j}_{\text{pos}}$ is the position embedding of the patch (i, j) in the $K \times K$ grid. The dense absolute localization (DAL) loss is:

$$\mathcal{L}_{\text{DAL}} = \mathbb{E}_{\mathbf{E}^{i,j}_{\text{pos}}, 1\le i,j\le K}[\|(i, j)^T - (\widetilde{i}, \widetilde{j})^T\|_1], \tag{2}$$

where the expectation is computed by averaging the ℓ_1 loss between the correspond $(i, j)^T$ and $(\widetilde{i}, \widetilde{j})^T$. Then, $\mathcal{L}_{\text{DAL}}$ is added to the standard cross-entropy loss ($\mathcal{L}_{\text{CE}}$) of the native ViTs. The final loss is: $\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\text{DAL}}$, where we set $\lambda = 0.01$ for all experiments. Note that the mapping function can be either linear or nonlinear. Throughout this work, we mainly discuss three implementations, including non-parametric PCA, learnable linear (LN), and nonlinear (NLN) projection layers. They will be discussed in details in Sec. 5.3.

4.3. MJP Position Embedding

The main goal of MJP position embedding in our work is to enhance the consistency (*i.e.*, position-insensitive property) of ViTs and preserve the accuracy (keeping the spatial

Algorithm 2 The pipeline of the proposed MJP.

Input: Input image: $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$;
Shuffle Ratio: γ; Patch Size: P
1: $\widetilde{\boldsymbol{x}}_p \leftarrow Alg.\ 1(\mathbf{x}, P, \gamma)$ // 1st & 2nd procedures
2: $\mathbf{E}_{\text{unk}}(\widetilde{\boldsymbol{x}}_p)$ // 3rd procedure
3: $\mathbf{DAL}(\mathbf{x} - \mathbf{x} \cap \widetilde{\boldsymbol{x}}_p)$ // 4th procedure, only for **training**

relation modeling functionality of PEs) for the standard classification tasks. Usually, with original input image patches $\mathbf{x}$ we can formulate the function of the input layer which is located before the Transformer block as follows:

$$\mathbf{z}_0 = [\mathbf{x}_{\text{CLS}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}, \cdots, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{\text{pos}}, \tag{3}$$

where $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$ is a trainable linear projection layer and $\mathbf{x}_p^k\mathbf{E}$ refer to the output of linear projection (*i.e.*, the patch embeddings), and $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$ refers to PEs (the additional one is applied to the [CLS] embedding).

Next, we apply the proposed block-wise random jigsaw puzzle shuffle algorithm to $\mathbf{x}_p$ and produce the transformed patch sequences $\widetilde{\mathbf{x}}_p$ with the shuffle ratio γ. In this scenario, if we maintain the original position embedding sequence, it leads to a mismatch issue between the shuffled patches and the position embedding sequence. Therefore, we introduce a shared *unknown* position embedding to the shuffled positions to alleviate the mismatching issue. With the corresponding updated mask $\widetilde{\mathbf{m}}$ from Alg. 1 (1st & 2nd procedures), we propose our MJP PEs:

$$\widetilde{\mathbf{E}}_{\text{pos}}^i = \begin{cases} \mathbf{E}_{\text{pos}}^i, & \text{if } \widetilde{\mathbf{m}}_i = 0 \\ \mathbf{E}_{\text{unk}}, & \text{if } \widetilde{\mathbf{m}}_i = 1 \end{cases} \tag{4}$$

where $\mathbf{E}_{\text{unk}} \in \mathbb{R}^{1 \times D}$ refers to a share learnable embedding (*i.e.*, *unknown* position embedding). It represents that the image patch in this position has random permutation and its position should be occluded (3rd procedure). $\mathbf{E}_{\text{pos}}$ is the original position embedding for the rest image patches.

Besides, we revisit the remaining PEs (*i.e.*, corresponding to the non-selected patches) and apply low-dimensional prior on them. The low-dimensional prior is imposed by the proposed DAL (Sec. 4.2) regression method for strengthening the spatial relation (4th procedure). A toy illustration of the proposed MJP is shown in Fig. 2(d), where the green color represents the randomly permuted patches and its corresponding *unknown* PEs, while the blue color indicates the rest regular patches and its related original PEs. We also formalize these procedures as an algorithm in Alg. 2.

Thus, we replace the input layer with a new formulation:

$$\widetilde{\mathbf{z}}_0 = [\mathbf{x}_{\text{CLS}}; \widetilde{\mathbf{x}}_p^1\mathbf{E}; \widetilde{\mathbf{x}}_p^2\mathbf{E}, \cdots, \widetilde{\mathbf{x}}_p^N\mathbf{E}] + \widetilde{\mathbf{E}}_{\text{pos}}. \tag{5}$$

the following procedures and modules are exactly the same as the ones in the original ViTs.

Table 1. Comparisons of different backbones on ImageNet-1K classification. Note that the image size here are all set to 224x224.

Method	Param.	Top-1 Acc. ↑	Diff. Norm. ↓	Consistency ↑
ResNet-50 [16]	25	79.3	11.77	51.5
ResNet-50 + MJP	25	79.4	7.11	69.3
DeiT-S [37]	22	79.8	16.21	64.3
DeiT-S + MJP	22	80.5	8.96	82.9
Swin-T [28]	29	81.3	15.49	41.5
Swin-T + MJP	29	81.3	12.36	66.9

5. Experiments

We follow the typical supervised pre-training procedure, where all the compared models are trained on ImageNet-1K [32] to show the capacity of our proposed MJP method. We also benchmark the proposed MJP method on ImageNet-1K, which contains 1.28M training images and 50K validation images of 1,000 classes. The training details mostly follow the training protocols[1] from Touvron *et al.* [37].

5.1. Regular ImageNet-1K training

We mainly compare with three typical existing methods, including two state-of-the-art Visual Transformers (*i.e.*, DeiT [37] and Swin [28]) and one widely-used CNN-based ResNet-50 [16]. All these methods are of comparable sizes (i.e., less than 30M parameters). Besides the common Top-1 accuracy (**Top-1 Acc.**), we also report another two evaluation metrics (*i.e.* Diff. Norm. and Consistency) to show the position invariance of a model to the jigsaw puzzle transformation. For a given input image $\mathbf{x}$, we collect its counterpart $\widetilde{\mathbf{x}}$ by applying masked jigsaw puzzle to shuffle a portion of selected patches. Next, we calculate the difference ℓ_2-norm (*i.e.*, **Diff. Norm.**) between the [CLS] embedding inferred from $\mathbf{x}$ and $\widetilde{\mathbf{x}}$:

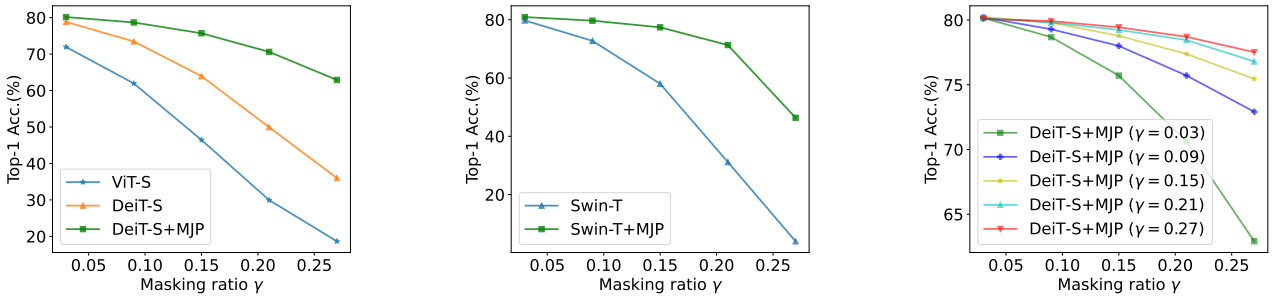
$$\|\mathcal{F}_{\text{CLS}}(\mathbf{x}) - \mathcal{F}_{\text{CLS}}(\widetilde{\mathbf{x}})\|_2^2. \tag{6}$$

For **Consistency**, we measure how many classification results stay the same (%) given the perturbation:

$$\mathbb{E}_{\mathbf{x}}[\mathbb{1}\{\arg\max P(\mathcal{F}(\mathbf{x})) = \arg\max P(\mathcal{F}(\widetilde{\mathbf{x}}))\}]. \tag{7}$$

$P(\mathcal{F}(\mathbf{x}))$ denotes the predicted classification probability of image $\mathbf{x}$, and $\arg\max P(\mathcal{F}(\mathbf{x}))$ represents the predicted class index. For a fair comparison, we use $\gamma = 0.15$ to create $\widetilde{\mathbf{x}}$ in Table 1. According to these results, the proposed MJP does not have a *negative* effect on the top-1 accuracy (MJP even brings a marginal improvement for DeiT-S). More importantly, it improves Diff. Norm. and Consistency by a large margin. Besides, MJP works well in the variants of ViTs (*e.g.*, Swin [28]), which shows a good potential generalization ability. Note that we add MJP to ResNet50 by only shuffling the image patches with Alg. 1 (No PEs). For Swin-T, we add MJP to the absolute PEs of Swin-T.

[1] https://github.com/facebookresearch/deit



(a) (b) (c)

Figure 3. Ablation on the mask ratio γ during inference: (a) comparisons among ViT-S, DeiT-S and our method (trained with $\gamma = 0.03$); (b) comparisons between Swin-T and our method (trained with $\gamma = 0.03$); (c) comparisons of our method on DeiT-S trained with different γ.

Table 2. Comparisons on robustness to common corruptions and adversarial examples.

Method	ImageNet-C	ImageNet-A		ImageNet-O
	mCE ↓	Acc ↑	AURRA ↑	AUPR ↑
DeiT-S	54.6	19.2	25.1	20.9
DeiT-S + MJP	**40.78**	**21.6**	**29.8**	**22.6**

Table 3. Ablation study on the proposed MJP method trained with different masking ratio γ.

Metric	Masking Ratio					
	0	0.03	0.09	0.15	0.21	0.27
Top-1 Acc.	80.0	80.5	80.3	80.4	80.2	80.3
Diff. Norm.	16.56	8.96	6.36	5.23	4.39	3.97
Consistency	64.0	82.9	88.1	90.5	92.3	93.1

5.2. Robustness on Challenging Sets

Besides the strength on standard classification, we also observe an auxiliary benefit of the proposed MJP on robustness. ImageNet-C [18] benchmarks a classifier's robustness to common corruptions[2]. The mean Corruption Error (mCE) is used to measure the generalization of a model at corrupted distributions (the lower the better).

ImageNet-A/O [19] focus on adversarial examples, and it enables us to either test image classification performance when the input data distribution shifts (*i.e.*, Acc and AURRA), or test out-of-distribution detection performance when the label distribution shifts (*i.e.*, AUPR)[3]. As shown in Table 2, the proposed MJP not only boosts the accuracy on the standard evaluation on ImageNet-1K validation set but also improves the robustness largely on the adversarial samples.

The underlying reason might be that MJP enforces the ViTs aware of both local and global context features, and it helps ViTs to get rid of some unnecessary sample-specific local features during the training. This has been verified by the visualization maps (see Fig. 3 of our Supp. Mat.).

5.3. Ablation Analysis

Results with different MJP ratios. As shown in Table 3, we test different masking ratios used in the block-wise masking strategy during the training. Obviously, Diff. Norm. has the inverse tendencies compared to Consistency, where a smaller Diff. Norm. usually indicates a larger (better) consistency score. Comparing the accuracy trained with $\gamma > 0$, it shows our model is not sensitive to different γ. In particular, a small ratio (*e.g.*, $\gamma = 0.03$) is sufficient for boosting the accuracy. In addition, a large ratio can reduces the Diff. Norm. and improves the consistency by a large margin. Moreover, the model trained with a larger γ is inclined to be more consistent as shown in Fig. 3 (c). In addition, we observe that the consistency keeps increasing when the mask ratio increases from 3% to 27%. However, further increasing the mask ratio will not bring consistency improvement and what is worse is that the accuracy marginally decreases.

For a model trained with a fixed γ, we also test its accuracy with different masking ratios during the inference. As shown in Fig. 3 (a) and (b), the performances of original models drop significantly when we shuffle more image patches. In contrast, our proposed method shows more consistent performances.

Comparison of variants of MJP. We test several variants of the proposed MJP, including (1) Removing the PEs from the original DeiT-S; (2) SPP: shuffling both the 16×16 patches and pixels within the patches, which is tested in MLP-Mixer [36] to validate the invariance to permutations; (3) JP: applying masked jigsaw puzzle to the sequence patches; (4) IDX: using an additional collection of embeddings to indicate the global indexes for the input patches; (5) UNK: replacing the PEs in the masked positions with a shared *unknown position embedding*; (6) DAL: jointly learning *dense absolute localization* regression in an self-supervised manner during the pretraining, where we provide PCA, linear (LN) and nonlinear (NLN) projections, respectively.

Table 4 shows the detailed ablation studies on the variants of our proposed MJP method. First, as expected, when we remove the PEs from the original DeiT-S model, the accuracy decreases by 2.3%, but the consistency achieves 100%. It

[2] https://github.com/hendrycks/robustness

[3] https://github.com/hendrycks/natural-adv-examples

Table 4. Ablation study on the variants of the proposed MJP.

Method	Top-1 Acc. ↑	Consistency ↑
A: DeiT-S [37]	79.8	64.3
B: A - PEs	77.5 (-2.3)	100.0
C: A + SPP [36]	74.9 (-4.9)	74.8
D: A + DAL (NLN)	80.0 (+0.2)	64.0
E: A + JP	79.2 (-0.6)	73.8
F: A + JP + IDX	79.9 (+0.1)	79.6
G: A + JP + UNK	80.1 (+0.3)	83.8
H: A + JP + UNK + DAL (PCA)	79.9 (+0.1)	83.4
I: A + JP + UNK + DAL (LN)	80.0 (+0.2)	83.8
J: A + JP + UNK + DAL (NLN)	80.5 (+0.7)	82.9

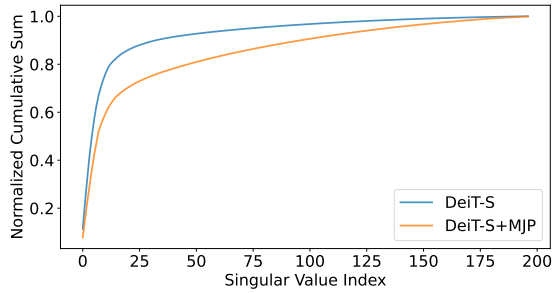


Figure 4. Distributions of accumulated eigenvalues of PEs.

verifies that the ViTs are naturally position-insensitive once without using PEs.

We observe that the previous SPP strategy harms the accuracy of the model (*i.e.*, -4.9%), which indicates it is insufficient to simply shuffle both patches and pixels in the input image. As we expect, the usage of UNK embedding alleviates the confusion between shuffled and unshuffled positions, which boosts both the accuracy and consistency.

Finally, we can find that using *dense absolute localization* regression on the unmasked PEs can marginally boost the accuracy. Although PCA does not involve more parameters to the model, it significantly increases the computation latency (e.g., $+46\%$ with only 5 iterations). In general, it is better to use a nonlinear projection (*e.g.*, a 3-layers MLP with negligible computation latency) to allow the learned PEs to aggregate more additional information.

Informativeness of the PEs. For a position space $\mathcal{P}$ (*i.e.*, a 1-dim or 2-dim space), we may not need a high-dimension embedding space $\mathcal{X}$ to model the positions. To measure the informativeness of the learned PEs, we apply singular value decomposition (SVD) to PEs and analyze their eigenvalue distributions. Fig. 4 plots the curves of accumulated energy/sum of top-n eigenvalues versus the energy of all the eigenvalues. Supposing that we are using PCA to project the PEs of both matrices, to achieve the same explained variance ratio, our MJP needs more singular values (*i.e.,* large dimensionality) than DeiT-S. This indicates that our positional embedding matrix is more informative. Similar observation is also revealed by Wang *et al.* [43].

Table 5. Comparisons on gradient leakage by analytic attack [29] with ImageNet-1K validation set, where we test (1) ViT-S, DeiT-S and our model in the setting (a); (2) ViT-S, DeiT-S and our model in the setting (b) (*i.e.*, MJP with $\gamma = 0.27$); (3) ablation on without (w/o) using $\mathbf{E}_{\text{unk}}$ in setting (a); and (4) Our model in setting (c).

	Model	Set.	Acc. ↑	MSE ↑	$\text{FFT}_{2\text{D}}$ ↑	PSNR ↓	SSIM ↓	LPIPS ↑
(1)	ViT-S [7]	a	78.1	.0278	.0039	19.27	.5203	.3623
	DeiT-S [37]		79.8	.0350	.0057	18.94	.5182	.3767
	DeiT-S (w/o PEs)		77.5	.0379	.0082	20.22	.5912	.2692
	DeiT-S+MJP		**80.5**	**.1055**	**.0166**	**11.52**	**.4053**	**.6545**
(2)	ViT-S [7]	b	18.7	.0327	.0016	18.44	.6065	.2836
	DeiT-S [37]		36.0	.0391	.0024	17.60	.5991	.3355
	DeiT-S (w/o PEs)		**77.5**	.0379	.0025	20.25	.6655	.2370
	DeiT-S+MJP		62.9	**.1043**	**.0059**	**11.66**	**.4493**	**.6519**
(3)	DeiT-S+MJP (w/o)	a	40.6	.1043	.0059	11.66	.4493	.6519
(4)	DeiT-S+MJP	c	62.9	.1706	.0338	8.07	.0875	.8945

5.4. Privacy Preservation

The fundamental principle of the gradient attack methods in federated learning is that each sample activates only a portion of content-related neurons in the deep neural networks, leading to one specific backward gradients for one related samples (*i.e.*, 1-to-1 mapping). Based on such an observation, we argue that feeding ViTs with input patches with permuted sequences may intuitively mislead the attack. This is because now both the original and the transformed inputs may be matched to the same backward gradients (*i.e.*, n-to-1 mapping).

To validate such an assumption, we utilize the public protocols[4] to recover image with gradient updates in the privacy attack. In this privacy attack, we apply the Analytic Attack proposed in APRIL [29], which is designed for attacking the ViTs. We randomly sample 1K images from the validation set of ImageNet-1K (*i.e.*, one image per category). To evaluate the anti-attack performance of a model, we introduce image similarity metrics to account for pixel-wise mismatch, including Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), cosine similarity in the Fourier space ($\text{FFT}_{2\text{D}}$), and Learned Perceptual Image Patch Similarity (LPIPS) [49]. Different from the evaluation in gradient attacks [14, 29, 48], *we suppose a model is with better capacity of privacy preservation when the recovered images from its gradient updates are less similar to the ground truth images*.

Given an image $\mathbf{x}$ and its transformed (*i.e.*, patch shuffled) version $\tilde{\mathbf{x}}$, a ViT model $\mathcal{M}$, and automatic evaluation metrics ϕ, we conduct three different settings for fair comparisons: (a) $\phi(\nabla\mathcal{M}(\mathbf{x}), \mathbf{x})$, (b) $\phi(\nabla\mathcal{M}(\tilde{\mathbf{x}}), \tilde{\mathbf{x}})$, and (c) $\phi(\nabla\mathcal{M}(\tilde{\mathbf{x}}), \mathbf{x})$, where ∇ refers to recovering input image through gradient attacks. Table 5 shows the quantitative comparisons between our method and the original ViTs for batch gradient inversion on ImageNet-1K. APRIL [29] enables a viable, complete recovery of original images from the gradient updates of the original ViTs. However, it per-

[4] https://github.com/JonasGeiping/breaching

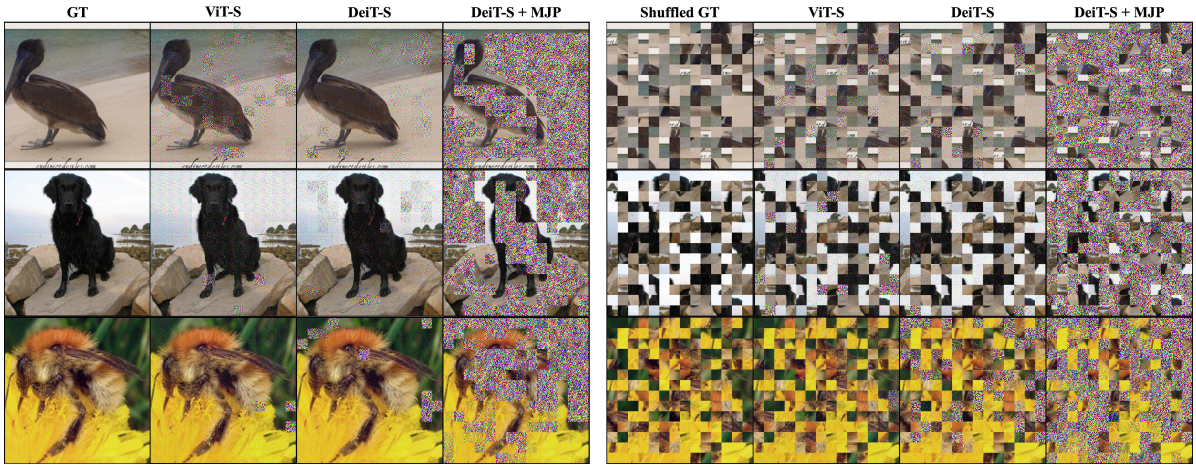


Figure 5. Visual comparisons on image recovery with gradient updates [29]. Our proposed DeiT-S+MJP model significantly outperforms the original ViT-S [7] and DeiT-S [37] models.

Table 6. Explained variance versus PCA projected dimensionality.

Projected Dimension	3	4	5	6	7
Deit-S EV (%)	54.61	68.55	77.95	85.54	90.74
DeiT-S+MJP EV (%)	46.74	58.36	69.10	78.13	84.55

forms worse in recovering from "DeiT-S+MJP", leading to best performances on all evaluation metrics and outperform others by a large margin.

More surprisingly, our proposed method makes APRIL yield unrecognizable images and fail in recovering the details in the original images (*i.e.*, noisy patches in the outputs), as shown in Fig. 5. The left four columns in Fig. 5 are tested on original images, where all PEs are standard and correspond to their patch embeddings. Meanwhile, the right four columns are tested with transformed ones, where the shuffled patches are with the shared unknown PEs. Both the visual and quantitative comparisons verify that our MJP alleviates the gradient leakage problem. We also notice that DeiT-S without using PEs is inclined to be at higher risk of privacy leakage (*i.e.*, easier to be attacked by gradients). These promising results indicate that our MJP is a promising strategy to protect user privacy in federated learning.

5.5. Discussion

PCA Projected Dimensionality. Table 6 presents the explained variance (EV) of our DeiT-S+MJP and Deit-S versus different projection dimension. A low dimensionality can explain a large amount of information, which proves that the embedding matrix is sparse in nature. Moreover, to achieve the same explained variance ratio, our DeiT-S+MJP needs a large dimensionality than Deit-S. This indicates that the positional embedding matrix of DeiT-S+MJP is less sparse but more informative.

Accuracy *Vs.* Shuffle Ratio. Intuitively, with the increase of the shuffle ratio from the proposed MJP method, the original intrinsic inductive bias will be undermined. However, from the experimental results, the performance of ViTs is actually boosted via the proposed MJP. To figure out the reason behind such counter-intuitive phenomenon, we visualize the last self-attention of our proposed method in Fig. 3 of our Supp. Mat. It shows that the attention heads of our method present more diverse and content-aware attentions than original DeiT-S. We think that it might be attributed to the proposed DAL, which strengthens the spatial information of the unmasked PEs in a more efficient manner. To this end, such learned content-aware attention becomes more meaningful and results in the better accuracy.

6. Conclusion

In this paper, we first visually demonstrate that PEs can explicitly learn the 2D spatial relationship from the input patch sequences. By feeding ViTs with transformed input, we identify the issue that PEs may weaken the position-insensitive property. Based on this observation, we propose an easy-to-reproduce yet effective Masked Jigsaw Puzzle (MJP) position embedding method to alleviate the conflict in PEs (preserving the consistency *versus* maintaining the accuracy). Experimental results show the proposed MJP can bring the the position-insensitive property back to ViTs without degenerating the accuracy on the large-scale dataset (*i.e.*, ImageNet-1K). In a certain sense, the proposed MJP is also a data augmentation technique, which boosts the robustness to the common corruptions (*e.g.*, ImageNet-C) and adversarial examples (*e.g.*, ImageNet-A/O). Surprisingly, MJP can improve the privacy preservation capacity of ViTs under typical gradient attacks by a large margin, which may pilot a new direction for privacy preservation.

Acknowledgments. This work was partly supported by the National AI Ph.D. for Society Program of Italy, and the EU H2020 project AI4Media (No. 951911).

References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations (ICLR)*, 2022. 2, 4

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 3

[3] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3

[4] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning (ICML)*, 2020. 3

[5] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 3

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018. 1, 3

[7] 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 (ICLR)*, 2020. 1, 2, 3, 7, 8

[8] Philipp Dufter, Martin Schmitt, and Hinrich Schütze. Position information in transformers: An overview. *Computational Linguistics*, 48(3):733–763, 2022. 1

[9] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197, 2021. 1

[10] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning (ICML)*, 2017. 3

[11] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 1

[12] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253, 2019. 1

[13] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on visual transformer. *arXiv*, 2020. 2

[14] Ali Hatamizadeh, Hongxu Yin, Holger Roth, Wenqi Li, Jan Kautz, Daguang Xu, and Pavlo Molchanov. Gradvit: Gradient inversion of vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 7

[15] 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 (CVPR)*, 2022. 3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[17] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[18] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 2, 6

[19] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 6

[20] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[21] Yunjae Jung, Donghyeon Cho, Sanghyun Woo, and In So Kweon. Global-and-local relative position embedding for unsupervised video summarization. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[22] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022. 1, 2

[23] Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. Shape: Shifted absolute position embedding for transformers. *arXiv preprint arXiv:2109.05644*, 2021. 3

[24] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 3

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 1, 3, 4

[26] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3, 4

[27] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022. 3

[28] 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 (ICCV)*, 2021. 1, 5

[29] Jiahao Lu, Xi Sheryl Zhang, Tianli Zhao, Xiangyu He, and Jian Cheng. April: Finding the achilles' heel on privacy for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10051–10060, 2022. 1, 2, 3, 7, 8

[30] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 2, 3

[31] Samrudhdhi B Rangrej, Chetan L Srinidhi, and James J Clark. Consistency driven sequential transformers attention model for partially observable scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2518–2527, 2022. 2

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2, 5

[33] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018. 3

[34] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 2

[35] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 3

[36] 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 (NeurIPS)*, 2021. 6, 7

[37] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, 2021. 1, 3, 5, 7, 8

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1, 2, 3

[39] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021. 1

[40] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 1

[41] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[42] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[43] Yu-An Wang and Yun-Nung Chen. What do position embeddings learn? an empirical study of pre-trained language model positional encoding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. 1, 3, 7

[44] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020. 2

[45] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34:28522–28535, 2021. 2

[46] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[47] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10502–10511, 2019. 1

[48] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 7

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[50] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[51] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 3