Supplementary Materials
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

1. Training Settings

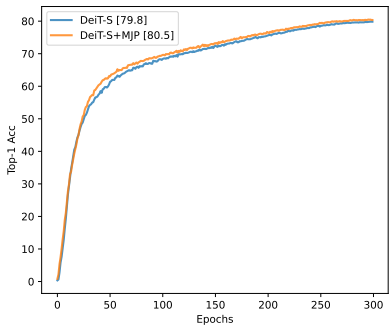
In the experiments on ImageNet-1K, we employ an AdamW [5] optimizer for 300 epochs using a cosine decay learning rate scheduler and 20 epochs of linear warm-up. A batch size of 1024, an initial learning rate of 0.001, and a weight decay of 0.05 are used. We include most of the augmentation and regularization strategies (e.g., repeated augmentation [4], CutMix [9], and Mixup [10]) of [7] in training, as shown in Table 1.

Table 1. Ingredients and hyper-parameters for our method.

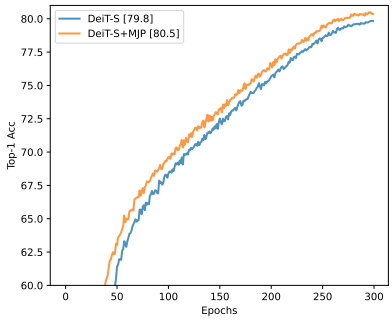
Epochs	300
Batch size	1024
Optimizer	AdamW
learning rate	$0.0005 \times \frac{\text{batchsize}}{512}$
Learning rate decay	cosine
Weight decay	0.05
Warmup epochs	20
Label smoothing ϵ	0.1
Dropout	$\times$
Repeated Aug	$\surd$
Gradient Clip.	$\surd$
Rand Augment	9/0.5
Mixup prob.	0.8
Cutmix prob.	1.0
Erasing prob.	0.25

2. Training Efficiency

We train our proposed method on ImageNet-1K with 8 V100 NVIDIA GPUs. We note that the computational consumption of the MJP procedure is negligible (*i.e.*, $+2\%$ time consumption per epoch). Meanwhile, MJP accelerates the convergence during the training as show in Figure 1.



(a)



(b)

Figure 1. Comparisons the top-1 accuracy of DeiT-S and DeiT-S+MJP during the training: (a) the whole training and (b) a zoom-in screenshot for the accuracy larger than 60%.

3. Position Embeddings

3.1. Position Regression

Inspired by Wang *et al.* [8], we also check whether a position embedding can actually capture its absolute position. To some extent, such position information could be reconstructed by a reversed mapping function $g : \mathcal{X} \rightarrow \mathcal{P}$, where $\mathcal{X}$ and $\mathcal{P}$ are embedding space and position space, respec-

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

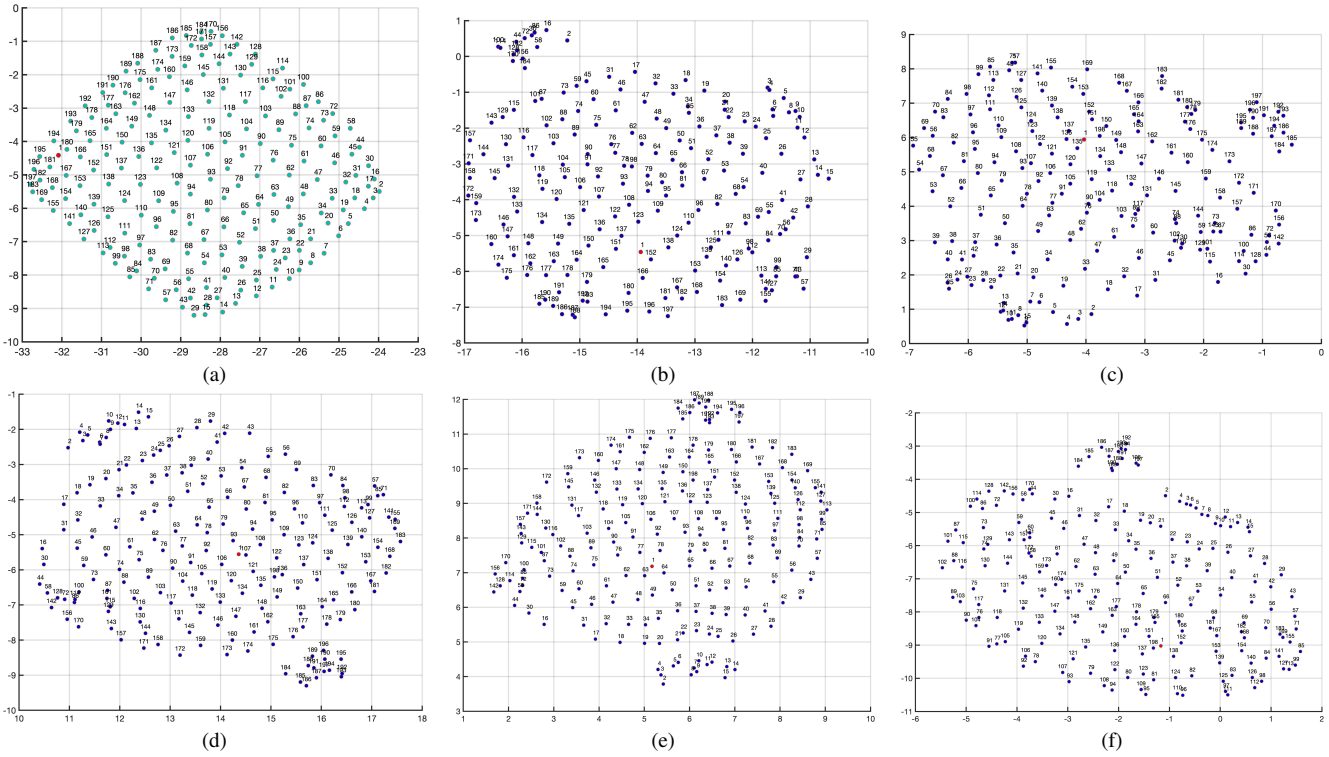


Figure 2. UMAP projections of the position embeddings collected from: (a) the original DeiT-S [2], (b) - (f) are DeiT-S+MJP trained with masking ratio $\gamma = \{0.03, 0.09, 0.15, 0.21, 0.27\}$ respectively.

Table 2. Mean absolute error of the reversed mapping function learned by linear regression.

Method	1D MAE	2D MAE
DeiT-S [7]	.945±.031	.076±.004
DeiT-S + MJP (DAL - LN)	.566±.013	.042±.002
DeiT-S + MJP (DAL - NLN)	1.301±.035	.134±.003

Table 3. Segmentaion results on ADE20K dataset (Pre-trained on ImageNet-1K).

Method	Top-1 Acc	mIoU	mAcc
Swin-Tiny [1]	81.3	43.87	55.22
Swin-Tiny + MJP	81.3	44.03 (+0.16)	55.50 (+0.28)

tively. Thus, we use linear regression to learn such a function g that transfers the embeddings to the original positions. For the position embeddings in ViTs, we can map them into either 1-D sequence space or 2-D patch grid space. Given we only have 196 data points (*i.e.*, 224×224 image resolution with 16×16 patch size) for each learned embedding, a 5-fold cross-validation is applied to avoid overfitting. The reversed mapping functions are evaluated by Mean Absolute Error (MAE), and the result is shown in Table 2.

From the results, the reversed mapping function of learnt position embeddings by "DeiT-S + MJP (DAL-LN)" can better represent the absolute positions. Meanwhile, the embeddings learned by the original DeiT-S and "DeiT-S + MJP (DAL-NLN)" also well learn the information about the absolute positions. Similar to [8], we have also tried some more complicated non-linear models such as SVM or MLP to map the embeddings back, which suffer from overfitting issue and perform worse. This implies that the position information in ViTs can actually be modeled by a linear model, which is consistent with Transformer encoders used in NLP field. Besides, the MAE of "DeiT-S+MJP (DAL - NLN)" is larger than the MAE of "DeiT-S+MJP (DAL - LN)" in Table 2. It indicates the nonlinear regression during the training aggregate more information beyond the position information (*i.e.*, more informative). The results are consistent with the Fig.4 in our main paper.

3.2. Projections of Position Embeddings

As shown in Figure 2, the position embeddings learned by the original DeiT-S is in a form of structured grids. Once we introduce MJP strategy to the training, it makes the projection of these position embeddings less structured. Meanwhile, the spatial-wise relative position information is preserved. We assume that the additional information (*i.e.*, more informative) in the position embeddings leads to such a dif-

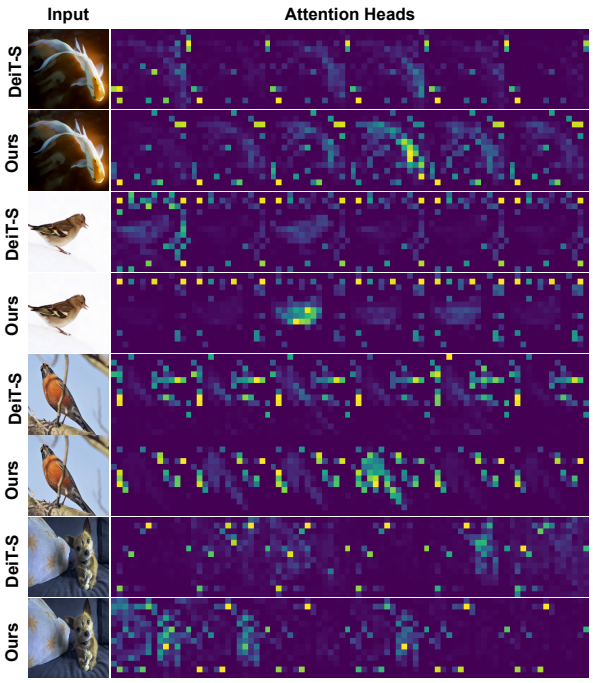


Figure 3. Visual comparisons between visualization maps of the last self-attention in DeiT-S [7] and our proposed DeiT-S+MJP.

ference.

4. Fine-tuning for Semantic Segmentation

A fine-tuning experiment on ADE20K dataset for semantic segmentation is presented. We use the popular UperNet architecture with a Swin-Tiny backbone pre-trained on ImageNet-1K. The re- sults shown in Tab. 3 indicate that our MJP doesn't have negative effects on other regular position-sensitive tasks.

5. Robustness to Corruptions

We show the details on the evaluation with ImageNet-C [3], as shown in Table 4. Compared to the original DeiT-S, our method achieves better performance on most tested corruptions.

6. Visualization of the last self-attention

The visualization the last self-attention of our method in Fig. 3. It shows that the attention heads of our method present more diverse and content-aware attentions than original DeiT-S.

7. More details on Privacy Preserving

7.1. Privacy Protection by Random Patch Permutation

Existing analytic gradient attack algorithms mainly model the problem as a linear system with closed-form solutions [6, 11]. For ViTs, the linear system is defined as:

$$\frac{\partial l}{\partial \mathbf{z}_0}\mathbf{z}_0^T = \boldsymbol{Q}^T\frac{\partial l}{\partial \boldsymbol{Q}} + \boldsymbol{K}^T\frac{\partial l}{\partial \boldsymbol{K}} + \boldsymbol{V}^T\frac{\partial l}{\partial \boldsymbol{V}} \tag{1}$$

where $\mathbf{z}_0$ denotes the image embedding that consists of patch embedding and positional embedding (i.e., $\mathbf{z}_0{=}\boldsymbol{x}_p\mathbf{E}{+}\mathbf{E}_{\text{pos}}$, where $\boldsymbol{x}_p$ denotes the sequence of flattened 2D patches and $\mathbf{E}$ represents the trainable linear projection). Since we have $\partial l/\partial \mathbf{z}{=}\partial l/\partial \mathbf{E}_{\text{pos}}$, the positional embedding layer is thus vulnerable to the gradient leakage attack. When the gradient $\partial l/\partial \mathbf{E}_{\text{pos}}$ is accessible, the image can be reconstructed as:

$$\boldsymbol{x}_p = \Big((\frac{\partial l}{\partial \mathbf{E}_{\text{pos}}})^{-1}(\boldsymbol{Q}^T\frac{\partial l}{\partial \boldsymbol{Q}}{+}\boldsymbol{K}^T\frac{\partial l}{\partial \boldsymbol{K}}{+}\boldsymbol{V}^T\frac{\partial l}{\partial \boldsymbol{V}}){-}\mathbf{E}_{\text{pos}}\Big)\mathbf{E}^{-1} \tag{2}$$

As indicated above, the gradient leakage of the PEs make the image easily reconstructed with closed-form solutions. To resolve this issue, we propose to randomly permute a portion of the image patches via our proposed block-wise random jigsaw shuffle algorithm **A.1**. The random permutation will drastically change both $\mathbf{E}_{\text{pos}}$ and $\partial l/\partial \mathbf{E}_{\text{pos}}$ in the above equation. This could significantly increase the difficulties to solve the linear system and reconstruct the image.

7.2. More visual results

Figure 4 and Figure 5 shows more visual results on image recovery with the gradient updates. Both these to figures clearly show that our method alleviates the privacy leakage issue a lot compared to the baselines, where most of the details are lost.

References

[1] Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2

[2] 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. 2

[3] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 3

[4] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

Table 4. Comparisons on robustness to common corruptions with ImageNet-C.

Method	Gaussian noise	Shot noise	Impulse noise	Defocus blur	Glass blur	Motion blur	Contrast	Elastic	Pixelate	JPEG	**mCE** ↓
DeiT-S	41.0	42.7	42.8	50.5	59.4	**45.5**	36.1	**43.0**	42.4	36.6	44.0
DeiT-S + MJP	**39.9**	**41.8**	**41.2**	**49.9**	**58.6**	47.4	**34.5**	43.3	**40.0**	**35.8**	**41.5**

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[6] 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. 3, 5, 6

[7] 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, 2, 3

[8] 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, 2

[9] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1

[10] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 1

[11] Junyi Zhu and Matthew Blaschko. R-gap: Recursive gradient attack on privacy. In *International Conference on Learning Representations (ICLR)*, 2021. 3

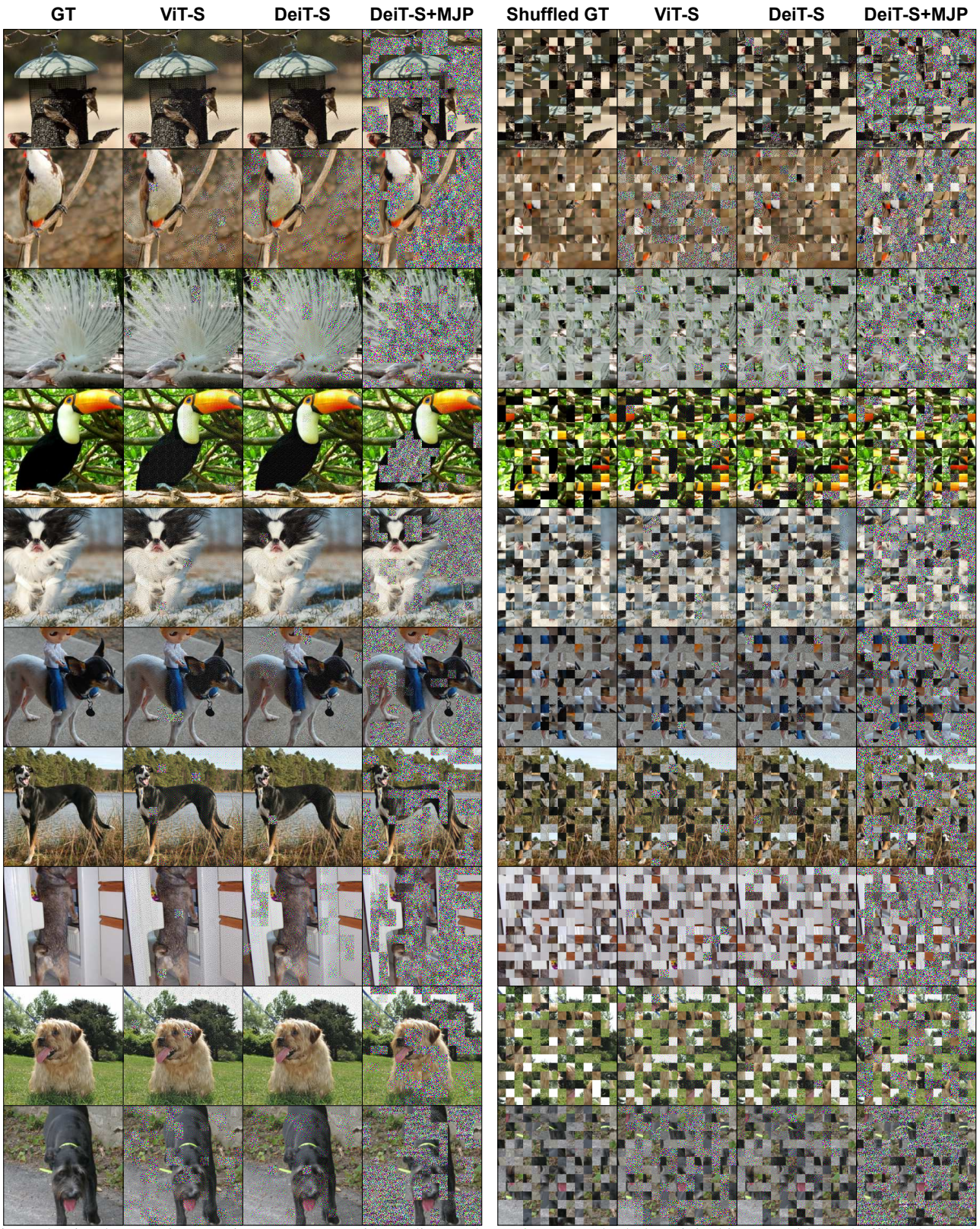


Figure 4. Visual comparisons on image recovery with gradient updates [6]. We test both the original images without shuffling the patches and images shuffled with a masked ratio $\gamma = 0.9$.

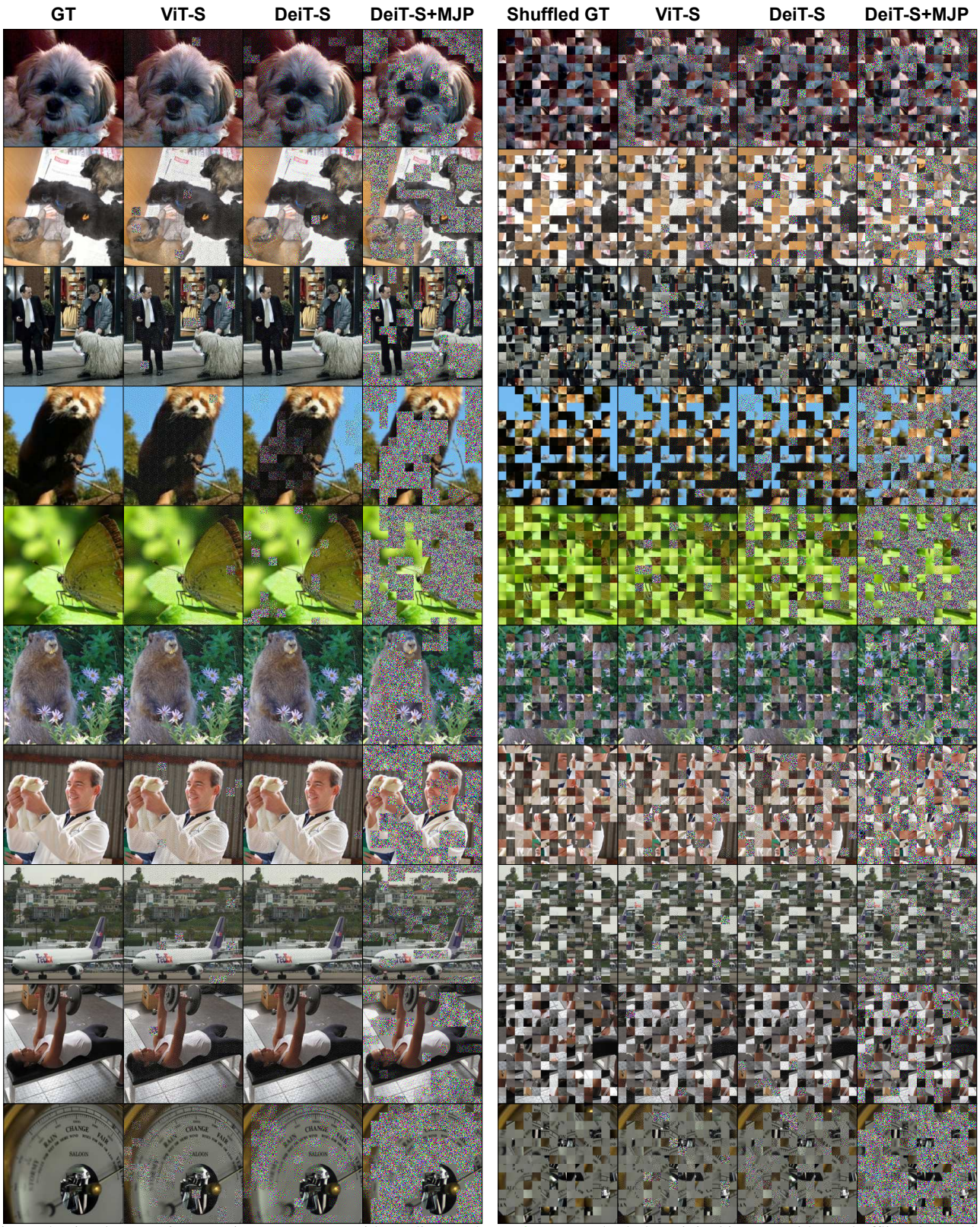


Figure 5. Visual comparisons on image recovery with gradient updates [6]. We test both the original images without shuffling the patches and images shuffled with a masked ratio $\gamma = 0.9$.