SeiT++: Masked Token Modeling Improves Storage-efficient Training (Supplementary Material)

A Additional Experimental Results

Table A: Exploring other token-based learning. We report top-1 accuracies (ViT-S) on ImageNet-100. Note that "our aug." denotes the proposed token augmentation strategies.

Input	MTM		MAGE [7]	
	w/ naïve aug.	w/ our aug.	w/ naïve aug.	w/ our aug.
Token (ViT-VQGAN [14])	81.4	**83.5** (+2.1)	81.5	**83.9** (+2.4)

A.1 Exploring Other Token-based Learning

To demonstrate the effectiveness of our token augmentation strategies, we explore another token-based learning approach, MAGE [7]. MAGE introduced a unified training framework for both generative training and representation learning. However, its focus is mainly on self-supervised representation learning, using online tokenization to convert images into tokens during each training iteration. This approach incurs significant memory and computational cost, limiting the storage-efficiency of discrete tokens. Unlike MAGE, our scenario is limited access to image data. Therefore, we trained MAGE with our tokenized data and compared two augmentation strategies (naïve aug. vs. our aug.) for contrastive loss adopted in MAGE. The ImageNet-100 (DeiT-S) classification results in Table A

show that contrastive loss is only helpful when combined with our token augmentation strategies, highlighting its effectiveness in token-based representation learning approaches.

Table B: Exploring other tokenizer. We report top-1 accuracies (ViT-S) on ImageNet-100. For VQGAN, we used a publicly available VQGAN [5] trained on OpenImages [6] as a tokenizer.

Method	Backbone	ViT-VQGAN [14]	VQGAN [5]
SeiT++ w/ MTM	ViT-S	83.5	**85.4** (+1.9)

A.2 Exploring Other Tokenizer for MTM

It is worth noting that we selected the ImageNet-1k-trained ViT-VQGAN [14] as our tokenizer to ensure fair comparisons on the ImageNet-1k benchmark and to leverage its strong generation performance. Additionally, we present MTM results using a different tokenizer in Table B: VQGAN [5] trained on OpenImages [6], which contains mostly complex scenes with multiple objects, as opposed to object-centric datasets like ImageNet. Despite the domain shift in data distribution, the OpenImages-trained tokenizer improves the classification performance by +1.9% thanks to its robust generation capability, consistent with the results in SeiT [8].

Table C: Impact of our method on the data-hungry scenario. We report top-1 accuracies (ViT-S) on ImageNet-100 with varying amounts of training data. Our method benefits more under less training data compared to its counterpart.

# of images Storage	127k 138 MB	76k 83 MB	25k 28 MB	13k 14 MB
SeiT [8]	77.3	70.3	53.3	43.2
SeiT++	**81.4** (+4.1)	**76.5** (+6.2)	**61.6** (+8.3)	**52.3** (+9.1)

A.3 Impact of Tokenadapt in Limited Data Scenario

Data augmentation becomes more critical in limited data scenarios. To assess the impact of our approach in such situations, we conduct experiments on the ImageNet-100 benchmark by randomly sampling data based on varying data ratios. The results (top-1 accuracies), as shown in Table C, consistently demonstrate the superiority of SeiT++ over SeiT. Notably, the performance gap increases as the amount of training data decreases. With only 13k images, our

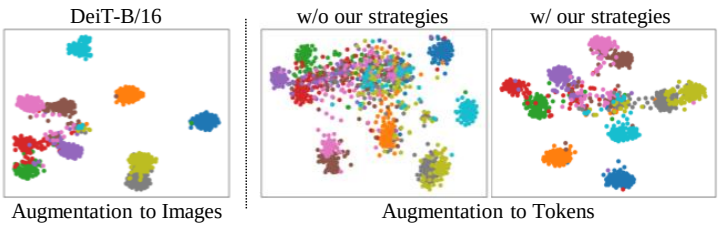

Fig. A: Linear separability visualization. We visualize augmented data points in the penultimate layer on ImageNet-1k, randomly sampling 10 classes. Different colors represent distinct classes. Notably, our proposed augmentation strategies exhibit enhanced linear separability compared to the direct application of pixel-based data augmentation to token embedding. This behavior aligns with the observed trend when applying data augmentation to images.

method shows an improvement of approximately 21% over the baseline, indicating the effectiveness of our approach, especially in data-scarce scenarios.

A.4 Linear Separability of TokenAdapt

To demonstrate the efficacy of our token augmentation strategies, we visualize the embedded feature representation in Figure A. This illustration offers insight into the linear separability of augmented data points, particularly when pixel-based data augmentation is applied through our proposed method. When pixel-based data augmentation is directly applied to token embedding without our method, there is significant overlap among augmented data points from different classes, indicating low linear separability. This overlap suggests that augmenting tokens without our method leads to substantial shifts in the distribution of training data. Maintaining distributional similarity between clean and augmented data is crucial for model performance, as witnessed in Cubuk *et al*. [3], and significant distribution shifts can result in performance degradation. In contrast, our method shows high linear separability, demonstrating its effectiveness in diversifying the data while minimizing the distributional gap between clean and augmented data. Moreover, this behavior aligns with the observed trend in the case of applying data augmentation to images (DeiT-B [11] in Figure A).

A.5 Loss Analysis of TokenAdapt

We analyze the influence of our approach on the training of ViT-B for both MTM pre-training and token-based ImageNet-1k classification, as illustrated by the loss curves in Figure B. Observing the training loss, our analysis shows that SeiT++, enhanced by our token augmentation strategy, maintains higher training loss values than SeiT [8] in both MTM pre-training and token-based ImageNet-1k classification. This pattern indicates that our method's token augmentation

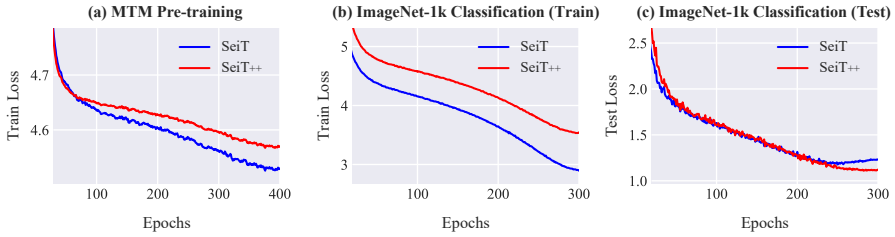

Fig. B: Loss curve visualizations. We provide loss curves for ViT-B during MTM pre-training and during token-based ImageNet-1k classification for both SeiT++ and SeiT [8]. Notably, SeiT++ shows more regularized results than SeiT, effectively leveraging pixel-based data augmentations on tokens to avoid overfitting. This evidences that token-based learning requires much stronger data augmentation for a more generalizable representation learning.

Table D: Data efficiency of the TokenAdapt module. We investigate the data efficiency of our TokenAdapt module by training it with varying amounts of training data. We validate the trained TokenAdapt module's efficacy in storage-efficient ImageNet-1k classification. Notably, the TokenAdapt module can be trained even with 100 images, indicating its robustness to data scarcity.

# of training data (TokenAdapt)	100	1k	5k	256k	1281k
Top 1 Acc. (IN-1k CLS.)	75.1	75.4	75.4	75.4	75.5

enriches the data diversity, which helps to mitigate overfitting. Meanwhile, Figure B(c) shows that the test loss aligns with the trends of SeiT. Notably, while SeiT's test loss increases in later training epochs, our approach consistently decreases, highlighting its effectiveness in preventing the overfitting problem.

A.6 Data Efficiency of TokenAdapt

To investigate the data efficiency of the TokenAdapt module, we explore the impact of reduced training data for the TokenAdapt module training. Specifically, we vary the number of training data for the TokenAdapt module training and subsequently train a token-based ImageNet-1k classification model using the trained TokenAdapt module. Table D shows that even with a reduced number of training data for our TokenAdapt module, the performance degradation is minimal. Notably, the top-1 accuracy remains more than 1%p higher than the baseline SeiT (74.0% top-1 accuracy), even with only 100 training data. This robust performance suggests that our TokenAdapt module is not significantly affected by variations in the training data size.

Table E: Computational costs comparison. We report computational costs and top-1 accuracy by training ViT-S on ImageNet-100. Baseline† denotes that (1) tokens are decoded to pixel-level images, (2) pixel-based data augmentations (*e.g.*, hFlip, affine transformations, Mixup, and CutMix) are applied to the decoded images, and then (3) augmented images are encoded to tokens back. GPU hours refer to the number of hours required for model training when using V100 4 GPUs.

Method	#params	GPU hours	Top 1 Acc.
SeiT [8]	22M	5.9	77.3
SeiT [8]†	22M+172M	49.2 (+734%)	80.1 (+2.8)
Ours	22M+3.7M	6.4 (+8%)	**81.4 (+4.1)**

A.7 Computational Costs

To validate the computational efficiency of TokenAdapt, we compare ViT-S models trained on the tokenized ImageNet-100 with different training strategies (Table E). Three scenarios are considered: (1) token-based augmentation only (*i.e.*, SeiT), (2) pixel-based augmentations in decoded images (SeiT†), and (3) pixel-based augmentations using our TokenAdapt module. For SeiT†, using the ViT-VQGAN decoder (86M) and encoder (86M) during each forward computation significantly increases training time by over 8 times compared to the original SeiT training, making it impractical for addressing the data augmentation challenge. In contrast, our TokenAdapt module, requiring only 3.7M parameters, increases training time by only 8% compared to SeiT while achieving a remarkable performance improvement over 3%p in top-1 accuracy. Notably, TokenAdapt even outperforms SeiT†, indicating that the full decoding-augmentation-encoding process may introduce undesirable noise during tokenization. These results demonstrate the efficient handling of the data augmentation challenge by our TokenAdapt module. Notably, the TokenAdapt module can be removed during the inference.

B More Qualitative Examples

TokenAdapt. We demonstrate the effectiveness of the proposed TokenAdapt and ColorAdapt by decoding the augmented tokens into images using the ViT-VQGAN decoder [14]. In Figure C, we compare the direct application of pixel-based data augmentations to token embedding with augmentations using our TokenAdapt module. As mentioned in Section 3.3, direct application of hFlip or affine transformations leads to disruptions in both the diagonal line and the silhouette of the object due to spatial information collapse. In addition, data augmentations related to interpolation (*e.g.*, Resize or Mixup) can result in undesired artifacts. Mixup, in particular, occasionally causes significant disruption of objects, leading to substantial performance degradation as shown in Figure 3(b).

The occurrence of these unexpected artifacts makes it difficult to use the data augmentations that are widely used in existing pixel image domains. Moreover,

in tasks such as MTM and semantic segmentation, where pixel-level details are essential, such artifacts significantly affect learning stability and model performance. In contrast, The tokens that are augmented using our TokenAdapt module exhibit more reasonable results, mitigating the image degradation caused by the direct application of pixel-based augmentations to tokens. Consequently, TokenAdapt consistently improves model performance across various scenarios by effectively leveraging pixel-based data augmentation in a token domain. Notably, as shown in Table 2, we observe that SeiT++ shows more significant performance gains with MTM, highlighting the importance of minimizing undesired artifacts to overcome the challenges of data augmentation in the tasks where pixel-level information is important.

ColorAdapt. In Figure D, we present the images decoded from color-augmented tokens by various existing color-based data augmentations. We employ brightness and contrast for pixel-based color augmentation, which are widely adopted for pixel-based vision model training. Specifically, we used brightness and contrast functions following the implementation [9]. For token-based color augmentation, we employ Emb-Noise [8] with the same optimized hyperparameters used in token-based vision model training [8]. Figure D illustrates the impact of the proposed ColorAdapt on the visual characteristics of augmented tokens. Unlike the existing color-based augmentations, our ColorAdapt effectively preserves object structure while introducing significant color variations. Recognizing the importance of maintaining object structure in diverse vision tasks (*e.g.*, fine-grained classification, semantic segmentation), our ColorAdapt opens up new possibilities for training more robust and adaptable vision models.

C Implementation Details

Masked Token Modeling. For masked token modeling (MTM), we follow the training recipe from MAGE [7], adjusting the pre-training epochs and the masking ratio. We pre-train the ViT-B model [4] for 400 epochs with a batch size of 4096 and fine-tune the model for 100 epochs with a batch size of 1024. During pre-training, we randomly mask out certain input tokens with a variable masking ratio ranging from 0.4 to 1. The base learning rate is 0.00015 and 0.001 for pre-training and fine-tuning, respectively. In MTM, we replace the Conv 4×4 Stem-Adapter module from SeiT [8] with Conv 2×2 as the patch embedding layer for ViT models. This adjustment is made because Conv 4×4 creates overlapping input patches, which hinders representation learning based on masked token modeling. Regarding data augmentation, recognizing the crucial role of pixel-level information in MTM, we apply geometric pixel-data augmentations using only the proposed TokenAdapt module, enhancing our training paradigm's effectiveness.

Token-based Image Classification. For token ImageNet-1k training, we follow the training recipe from SeiT [8], adjusting only the warm-up epochs for

Fig. C: TokenAdapt provides more reasonable results when augmenting tokens. We present ViT-VQGAN decoded images to verify the quality of tokenizations after applying pixel-based data augmentations to tokens. The direct application of pixel-based data augmentation to token embedding (w/o TokenAdapt) results in undesired artifacts. In contrast, our TokenAdapt yields more reasonable results. Token w/ Affine indicates the application of affine transformations (*e.g.*, rotation, translation, shear, etc.) to token embedding. For Token w/ Mixup, we mixed the two tokens with a 1:1 ratio (*i.e.*, interpolation ratio $\lambda = 0.5$).

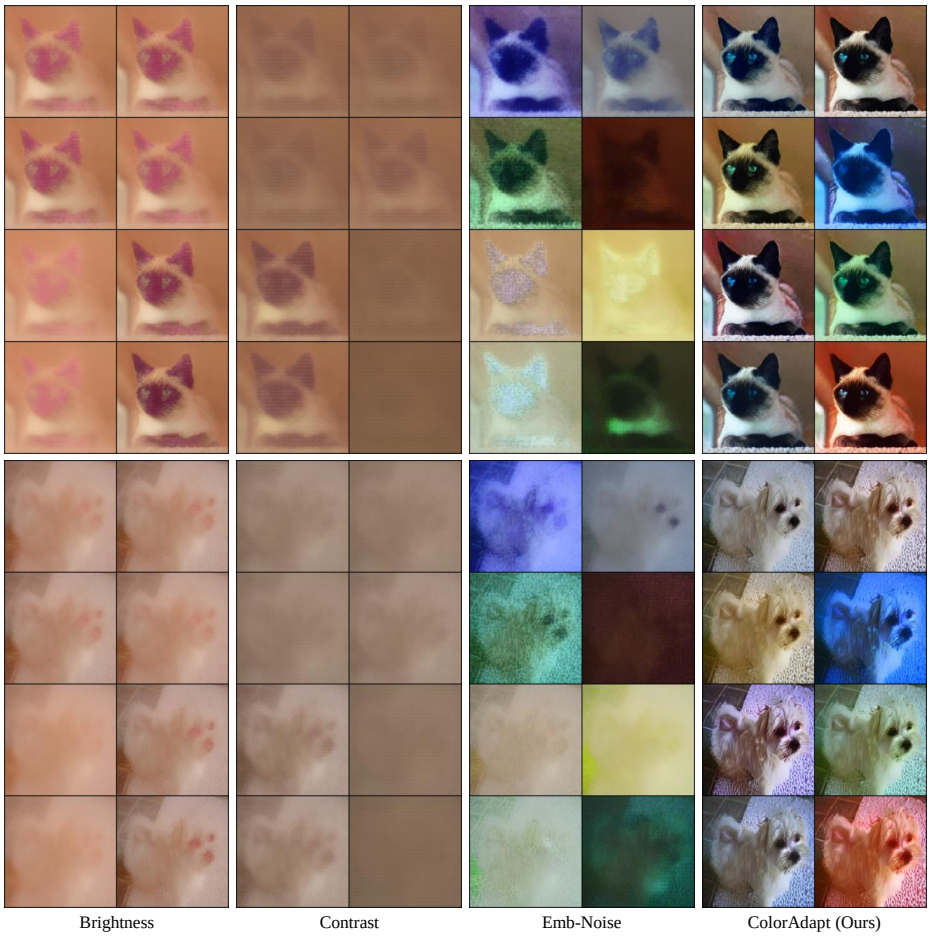

Fig. D: ColorAdapt provides more reasonable results related to color changes. We present ViT-VQGAN decoded images to verify the quality of tokenizations after color changes. We use the brightness and contrast function following the implementation [9]. Emb-Noise is the color-based token augmentation [8]; we use the same optimized hyper-parameters. Our ColorAdapt effectively preserves object structure in contrast to the failure of the counterparts.

stable convergence. We conducted token-based image classification on the ViT-B/16 model [4, 11] and used a learning rate of 0.0015 with cosine scheduling and a weight decay of 0.1. The model was trained for 300 epochs with a batch size of 1024. Regarding data augmentation, SeiT [8] incorporates Token-RRC, Token-EDA, Token-CutMix, and Emb-Noise. As a default, we integrate these token-based data augmentation methods and further enhance token diversity by applying additional augmentations using our TokenAdapt and ColorAdapt. Specifically, we apply pixel-based data augmentations (*e.g.*, RRC, hFlip, affine transformations, Mixup [16], and CutMix [15]) to tokens using our TokenAdapt module with a probability of 0.5. We adopt the hyperparameters for data augmentation proposed in DeiT [11]. For token-based fine-grained classification, following SeiT, we use DeiT's training recipe. When the number of data points decreased in all experiments, we adjust the number of total training iterations to ensure a fair comparison.

Token-based Semantic Segmentation. For the tokenized ADE-20k dataset preparation, we initially resize the entire ADE-20k dataset to 512×512. Following the procedure described in SeiT, we use the ImageNet-1k-trained ViT-VQGAN tokenizer [14] to extract tokens from the resized images. Token-based semantic segmentation on ADE-20k follows the training recipe of mmsegmentation [2], using the DeiT-B/16 model with UperNet [12]. The training involves a learning rate of 6e-5 with polynomial scheduling and a weight decay of 0.01. The model was trained for 80k iterations with a batch size of 16. Regarding data augmentation, we do not employ mixup-based augmentations in token-based semantic segmentation. This behavior is consistent with recent studies [1, 10, 13] that exclude mixup-based augmentation in their training recipes. Furthermore, similar to MTM, we observe that token-level Random Resized Crop (RRC) does not improve performance due to the importance of maintaining structural integrity in pixel-level classification tasks. Thus, we apply geometric pixel-based data augmentations only using our TokenAdapt module.

References

1. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. arXiv preprint arXiv:2205.08534 (2022)
2. Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation` (2020)
3. Cubuk, E.D., Dyer, E.S., Lopes, R.G., Smullin, S.: Tradeoffs in data augmentation: An empirical study (2021)
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
5. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
6. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. International Journal of Computer Vision **128**(7), 1956–1981 (2020)
7. Li, T., Chang, H., Mishra, S., Zhang, H., Katabi, D., Krishnan, D.: Mage: Masked generative encoder to unify representation learning and image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2142–2152 (2023)
8. Park, S., Chun, S., Heo, B., Kim, W., Yun, S.: Seit: Storage-efficient vision training with tokens using 1% of pixel storage. arXiv preprint arXiv:2303.11114 (2023)
9. Riba, E., Mishkin, D., Ponsa, D., Rublee, E., Bradski, G.: Kornia: an open source differentiable computer vision library for pytorch. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3674–3683 (2020)
10. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7262–7272 (2021)
11. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. pp. 10347–10357. PMLR (2021)
12. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European conference on computer vision (ECCV). pp. 418–434 (2018)
13. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34**, 12077–12090 (2021)
14. Yu, J., Li, X., Koh, J.Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., Wu, Y.: Vector-quantized image modeling with improved vqgan. arXiv preprint arXiv:2110.04627 (2021)
15. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6023–6032 (2019)
16. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)