Supplementary material for
Beyond Grids: Exploring Elastic Input Sampling for Vision Transformers

Adam Pardyl[1,2,3] Grzegorz Kurzejamski[1] Jan Olszewski[1,4]
Tomasz Trzciński[1,5,6] Bartosz Zieliński[1,2]
[1]IDEAS NCBR
[2]Jagiellonian University, Faculty of Mathematics and Computer Science
[3] Jagiellonian University, Doctoral School of Exact and Natural Sciences
[4] University of Warsaw [5] Warsaw University of Technology [6] Tooploox

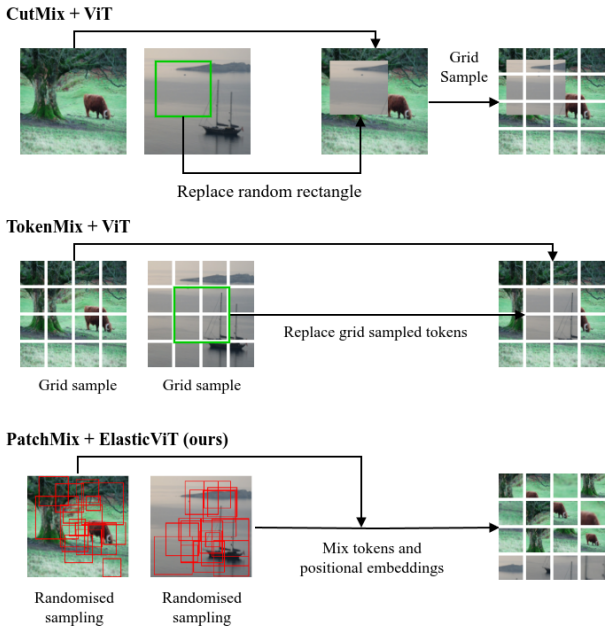

Figure 1. **PatchMix:** Standard CutMix poses an issue in ElasticViT training regime, as proportions of mixed targets may change in randomized sampling. TokenMix replaces patches after sampling, incidentally eliminating this issue, but nevertheless relies on grid sampling. Our PatchMix takes full advantage of the ElasticViT position and scale encoding, and enables mixing of randomly sampled patches of different scales.

1. Additional experiments

1.1. PatchMix ablation

To evaluate the effectiveness of PatchMix (see Fig. 1 for visualization) we perform an ablation study, training the model with the regime presented in Sec. 4 of the main paper, but without the PatchMix augmentation applied. The results are presented in Tab. 1. The PatchMix augmentation improves the accuracy of ElasticViT by over 1.5% on ImageNet-1k.

ElasticViT	Accuracy
without PatchMix	80.67%
with PatchMix	82.04%

Table 1. **PatchMix ablation study:** The effectiveness of the PatchMix augmentation evaluated on the Imagnet-1k dataset. The test was performed without any introduced perturbations. We observe that PachMix provides over 1.5% gain in accuracy.

1.2. Grid density

Continuing on scale elasticity experiments presented in Sec. 5.1, we decided to investigate the resistance of ViT architectures to change of grid density. In real-world scenarios, scale changes are quite common. However, when conducting synthetic experiments with ViT, the typical approach is to maintain a consistent input image resolution and grid layout density.

In this setup, we adjust the grid density, thereby altering the number of patches while using the same input image. This modification process is illustrated in Fig. 2 as *Grid density* and compared to the *Grid zoom* perturbation shown in Sec. 5.1. The outcomes for standard ViT, MAE and ElasticViT are presented in Fig. 3. PVT and Swin models were omitted in this evaluation, as changing the grid density in those models is not trivial due their internal structure. We observe, that all models perform similarly well when decreasing the density of the grid, while only ElasticViT can utilize denser sampling to its advantage.

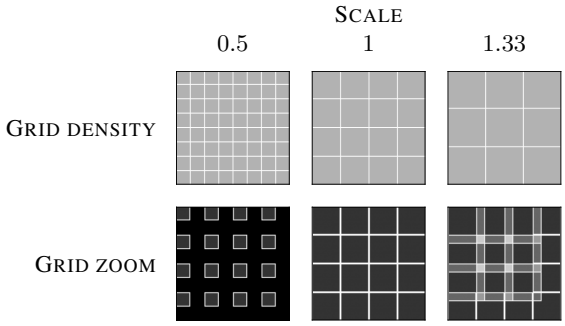

Figure 2. **Scale elasticities:** We use two perturbation scenarios, that change the size of a patch in a grid. The first (grid density) changes the number of patches. The second (grid zoom) keeps same number of patches but changes theirs size.

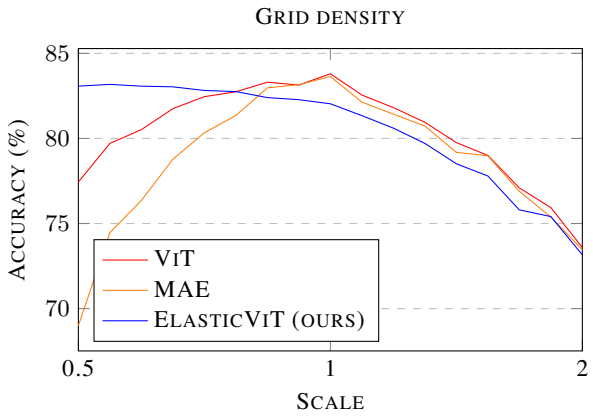

Figure 3. **Grid density**: The impact of changing the grid density (see Fig. 2) on accuracy. ElasticViT can utilize extra information from denser grid sampling (0.5), outperforming the original ViT.

1.3. Is it better to down-scale input or dropout patches?

The results from our previous experiments naturally lead to an important question: When faced with computational constraints, is it more beneficial to remove an entire patch from the input or to rescale neighboring patches in order to maintain complete image coverage, even at the cost of changing the token scale? To investigate this, we conduct experiments where we iteratively select a random 2×2 patch block from the uniform 14×14 patch grid and modify it in two ways. The first is to replace a 2×2 block of patches with a single larger patch, preserving the same coverage. In the second, we remove three out of four patches within the block. In both cases, these operations reduced the input token set by three, either through a dropout operation or by changing the scale of the patches.

The results of these experiments are depicted in Fig. 4. Surprisingly, ElasticViT exhibits almost no discernible difference in performance between the dropout and rescale operations, with dropout slightly outperforming rescaling at the extremes. Again, This would suggest a potential overfit with results better when having only 25% of the image covered by patches than having 100% coverage but with two times lower sampling resolution. In contrast, for the original ViT model, there is a visible distinction in performance between the two methods of limiting token counts, with the rescaling option proving to be a much more effective solution at the extremes.

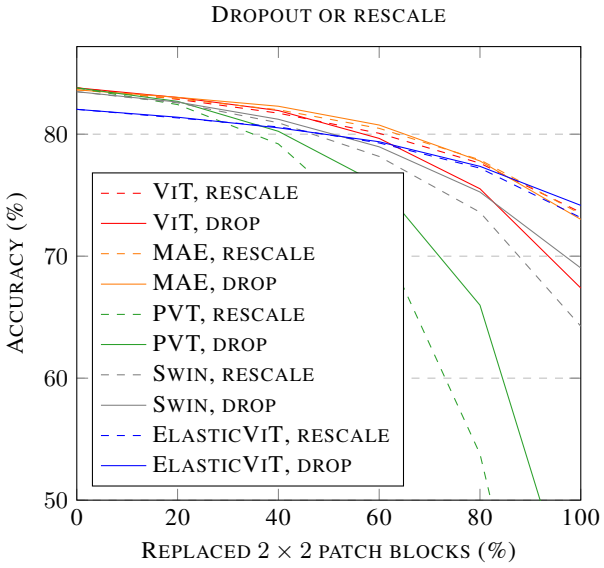

Figure 4. **Reducing number of patches:** Evaluation of the trade-off between lower resolution sampling and patch dropout. The X-axis represents the number of 2×2 patch blocks either replaced by a single lower-resolution patch or kept with only one patch while dropping the rest. We observe that the chosen strategy only affects ViT significantly at a high number of replacements, whereas ElasticViT remains unaffected due to its greater elasticity in handling missing data.

1.4. Transfer learning (continued)

1.4.1 Pascal VOC dataset

In this section we show additional results for transfer learning elasticity, evaluated on the Pascal VOC dataset. We follow the same training and evaluation setup as in Sec. 5.7 of the main paper. Results of the experiments are presented in Fig. 5. We observer, that standard ViT performs the best for the native resolution, but is outperformed by ElasticViT when elasticity is introduced. Both PVT and Swin performs slightly worse, and surprisingly, the self-supervised pre-trained MAE performs the worst, failing to achieve event half of the mean average precision score of ViT and ElasticViT.

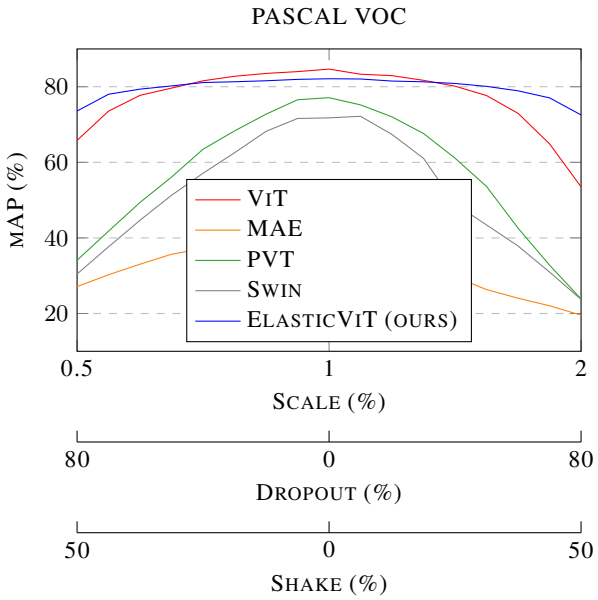

Figure 5. **PASCAL VOC transfer learning**: Results of fine tuning the last layers of models for multi-label classification of the PASCAL VOC 2007 dataset. All models were trained with standard grid sampling. We observe, that for high perturbation rate ElasticViT outperforms baselines. Note the surprisingly poor performance of the self-supervised pre-trained MAE.

Model	Sampling type	Accuracy
ViT	GRID	82%
ElasticViT	EDGE	87%

Table 2. **ColonCancer transfer learning:** Results of fine-tuning the last layers of models for binary classification of the ColonCancer dataset. The standard ViT model was run with grid sampling, while our ElasticViT model utilized EDGE sampling as described in the main paper. We observe that our model outperforms the standard ViT, benefiting from variable scale sampling.

1.4.2 ColonCancer dataset

We further test transfer learning properties of our model, evaluating it on the ColonCancer dataset. The dataset consists of histopathological images to be classified as malignant or benign. As previously, we train only the final linear layer of the model. For standard ViT we apply regular grid sampling, our ElasticViT uses EDGE sampling as described in the main paper. Results are presented in Tab. 2, showing superior performance of ElasticViT. We attribute this superior performance to EDGE sampling, which extracts more patches in regions containing cell nuclei, enabling ElasticViT to create a better representation of the data.

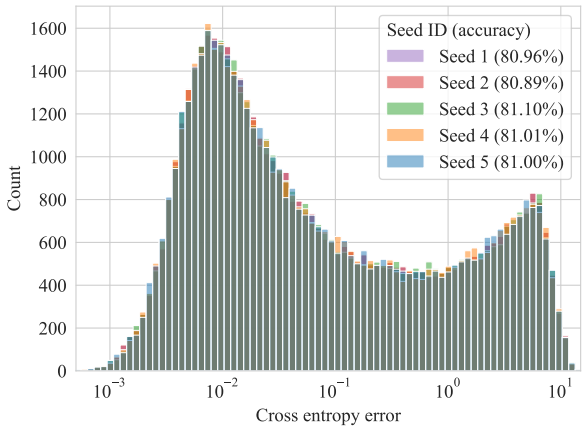

Figure 6. **Elastic sampling impact on performance across multiple runs:** The histogram represents the distribution of classification error for five inference runs with different random seeds used for elastic sampling. We observe that the differences between runs are insignificant. Note that the accuracy scores of particular runs are provided in the plot legend.

1.5. Stability of elastic evaluation

As our elasticity benchmark introduces randomness, we run the evaluation pipeline on the same trained model multiple times to check for any fluctuations in performance across different random seeds. The differences in performance are insignificant over multiple runs of elastic sampling, as shown in the histogram of performance standard deviations in Fig. 6. This consistency can be attributed to the large size of the ImageNet-1k validation set.

1.6. Patch redundancy

Elastic sampling allows for patch overlap, which provides redundant information to the model. In this experiment, we explored vision transformer capability to deal with redundant information. The model was provided with a standard full grid of 16×16 patches, that covered the entire image. Then, a number of redundant, randomly sampled patches with scales between 0.5 and 2 was added to the input. The results are presented in Fig. 7. We observe, that for standard ViT and MAE models those redundant patches essentially constitute noise, which reduces the overall performance. Our ElasticViT is capable of using the extra information to slightly increase the accuracy.

1.7. Overall performance comparison

To assess the overall elasticity of the compared methods, in Fig. 8 we present a critical difference diagram, aggregating results from Fig. 4 and Fig. 5 of the main paper.

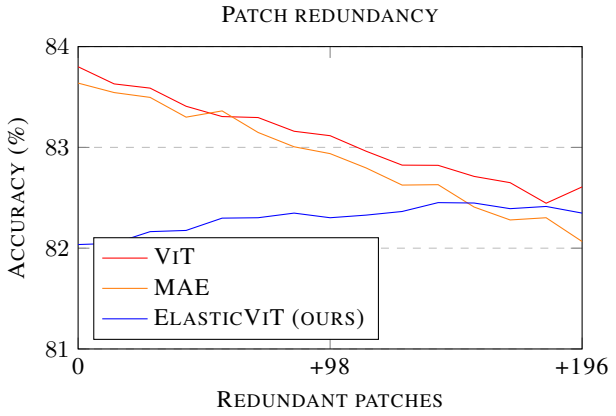

Figure 7. **Patch redundancy**: The impact of adding randomly sampled patches in addition of a standard sampling grid. We observe, that standard ViT and MAE lose performance, as those randomly sampled patches as treated as noise. In contrast, ElasticViT can utilize the extra information to improve the result.

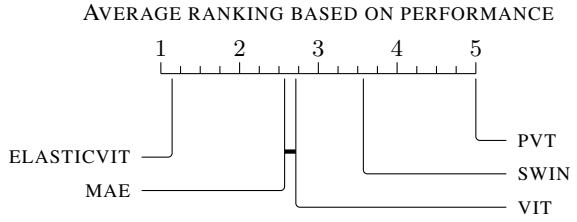

Figure 8. **Critical difference diagram**: Average performance ranking of ElasticViT and baseline methods for ImageNet with high perturbation (most extreme settings). ElasticViT significantly outperforms remaining approaches, while more structured methods (PVT and Swin) perform significantly worse than simple ViT. Moreover, the difference between ViT and MAE is insignificant. Notice that this diagram was generated for rankings generated based on results from Fig. 4 and Fig. 5 of the main paper.

2. Pipeline visualizations

In Fig. 9 we present visualization of our elasticity pipeline output. For clarity, patches are separated with red borders and patch overlap is highlighted in bright colors.

3. Theoretical analysis of input sampling strategies and their impact on positional embedding

3.1. Definition recall

To strictly define the evaluation pipeline, let us consider image I for which we generate set P of patches $p = (x, y, s)$, where x and y denote the top-left corner's coordinates, and s represents the relative scale (i.e. we sample a patch $r \cdot s \times r \cdot s$ and rescale it bilinearly to size $r \times r$). Initially, the coordinates x and y are from the regular grid and $s = 1$. However, in the next step, we perturb them with three functions corresponding to the considered elasticities:

- $\mathrm{E}_{\mathrm{scale}(s_1,s_2)}(P)$ - introduces the scale perturbations, sampling the s parameter of every patch $p \in P$ independently and uniformly from range $[s_1, s_2]$.
- $\mathrm{E}_{\mathrm{miss}(d)}(P)$ - adds missing data perturbations, dropping out d patches from P randomly with equal probability.
- $\mathrm{E}_{\mathrm{pos}(q)}(P)$ - applies positional perturbation, modifying x and y parameters of each patch $p \in P$, independently moving them by offsets sampled uniformly from range $[-r \cdot q, r \cdot q]$, where r is the size of the patch.

The patches are described by their upper left (x, y) and lower right corner $(x + rs, y + rs)$. Then each coordinate $\mathrm{pos} \in \{x, y, x + rs, y + rs\}$ is encoded by the sinusoidal positional encoding:

$$\mathrm{PE}_{(\mathrm{pos},2i)} = \sin\left(\frac{\mathrm{pos}}{10000^{2i/l}}\right)$$
$$\mathrm{PE}_{(\mathrm{pos},2i+1)} = \cos\left(\frac{\mathrm{pos}}{10000^{2i/l}}\right).$$

which are concatenated into a single vector.

3.2. Perturbations influence on positional embedding

The application of $\mathrm{E}_{\mathrm{miss}}$ does not affect the positional encoding of a patch. However, if a patch was not dropped by the $\mathrm{E}_{\mathrm{miss}}$ perturbation, then the other two perturbations can modify its position embedding.

After application of $\mathrm{E}_{\mathrm{pos}}$, the values of x and y get updated to $x+\Delta x$, $y+\Delta y$, while the lower right corner $x+rs$ and $y+rs$ get updated to values $x+rs+\Delta x$ and $y+rs+\Delta y$. The offset values Δx and Δy do not depend on x nor y. Thus, for x, we obtain the following positional embedding

$$\mathrm{PE}_{(x+\Delta x,2i)} = \sin\left(\frac{x}{10000^{2i/l}} + \frac{\Delta x}{10000^{2i/l}}\right)$$
$$\mathrm{PE}_{(x+\Delta x,2i+1)} = \cos\left(\frac{x}{10000^{2i/l}} + \frac{\Delta x}{10000^{2i/l}}\right),$$

which holds analogously for the other three coordinates $\{y, x + rs, y + rs\}$.

When it comes to $\mathrm{E}_{\mathrm{scale}}$, it modifies the s value to $s + \Delta s$ therefore $\mathrm{pos} \in \{x, y\}$ remains unchanged, while both coordinates $\mathrm{pos} \in \{x + rs, y + rs\}$ are offset by the value $\Delta = r\Delta s$. The formula is analogous to the $\mathrm{E}_{\mathrm{pos}}$ case.

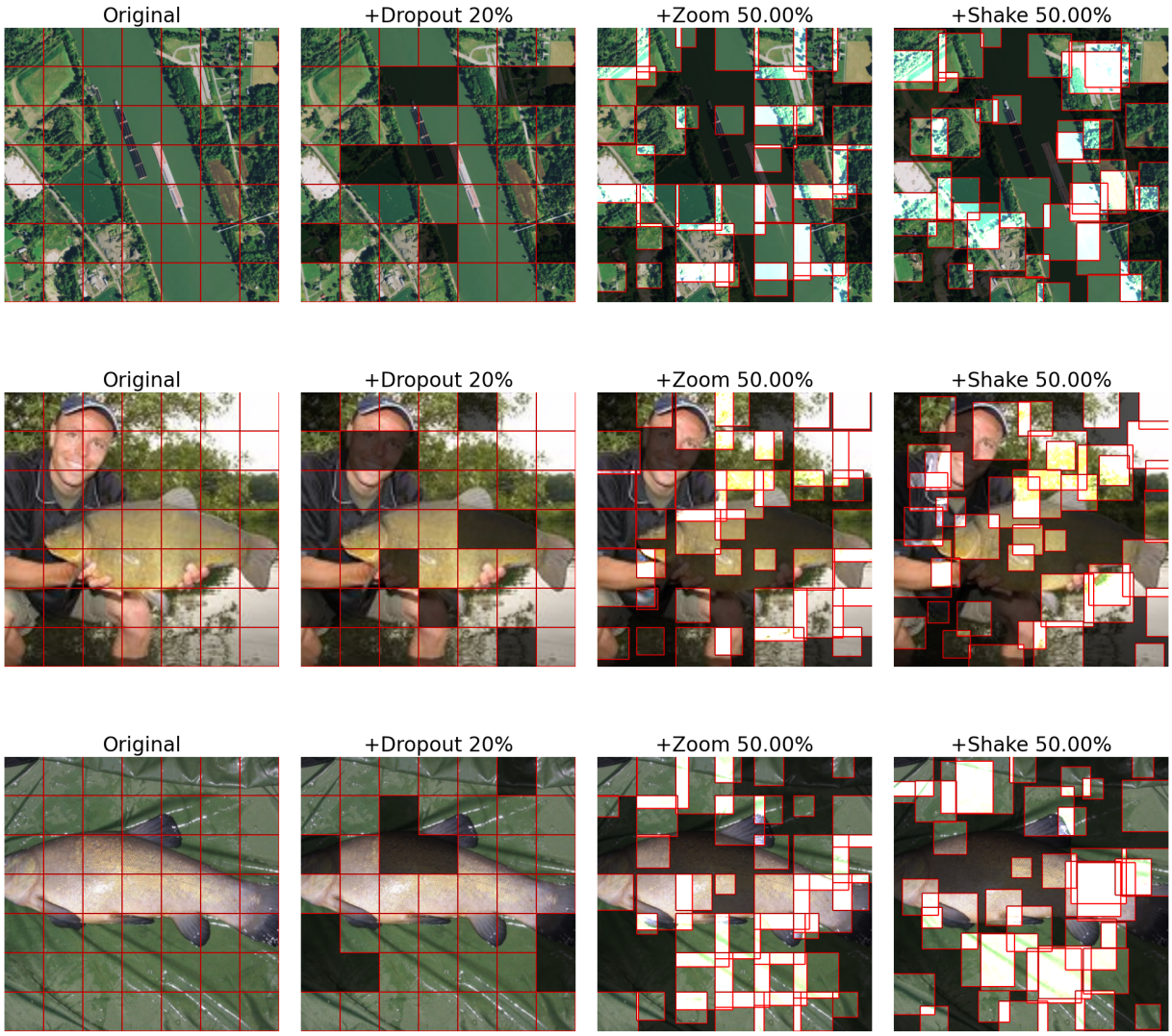

Figure 9. **Elasticity pipeline visualization:** In this figure, we present visualizations of successive input data perturbations. For clarity, patch borders are marked in red, and patch overlap is highlighted.