NARAIM: Native Aspect Ratio Autoregressive Image Models

Daniel Gallo Fernández[*]
University of Amsterdam
daniel.gallo.fernandez@student.uva.nl

Robert van der Klis[*]
University of Amsterdam
robert.van.der.klis@student.uva.nl

Răzvan-Andrei Matişan[*]
University of Amsterdam
razvan.matisan@student.uva.nl

Janusz Partyka[*]
University of Amsterdam
janusz.partyka@student.uva.nl

Efstratios Gavves
University of Amsterdam
e.gavves@uva.nl

Samuele Papa
University of Amsterdam
s.papa@uva.nl

Phillip Lippe
University of Amsterdam
p.lippe@uva.nl

Abstract

While vision transformers are able to solve a wide variety of computer vision tasks, no pre-training method has yet demonstrated the same scaling laws as observed in language models. Autoregressive models show promising results, but are commonly trained on images that are cropped or transformed into square images, which distorts or destroys information present in the input. To overcome this limitation, we propose NARAIM, a vision model pre-trained with an autoregressive objective that uses images in their native aspect ratio. By maintaining the native aspect ratio, we preserve the original spatial context, thereby enhancing the model's ability to interpret visual information. In our experiments, we show that maintaining the aspect ratio improves performance on a downstream classification task.

1 Introduction

Recent research has shown that pre-training large transformer models on vast datasets produces highly capable models, which can be fine-tuned for various tasks [4, 19]. In language modelling, this is achieved by training models with billions of parameters on next-token prediction using datasets with trillions of tokens, as this prediction task scales well with data, compute, and model size [4, 9, 13, 14, 16, 18]. However, in computer vision, no pre-training task has shown similarly favourable scaling laws [10]. For example, reconstruction tasks, e.g., masked autoencoders, have been found lacking when transferred to downstream tasks [1, 11]. Inspired by the success of autoregressive objectives in language, El-Nouby et al. [10] introduce Autoregressive Image Models (AIM), a class of vision transformers (ViTs) [8] trained with next patch prediction. These models achieve promising results, demonstrating similar scaling behaviours to autoregressive language models [10].

While ViTs can efficiently train on images with varying aspect ratios, the common practice remains to resize images to a square resolution for training, typically using random resized cropping, which can distort image information while providing regularization [2, 5, 10, 17]. However, Dehghani et al. [6] suggest that maintaining the native aspect ratio improves classification performance. We further hypothesize that this distortion is even more critical for generative pre-training objectives, where resizing could disrupt patterns or structures the model needs to reproduce.

[*] Equal contribution

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

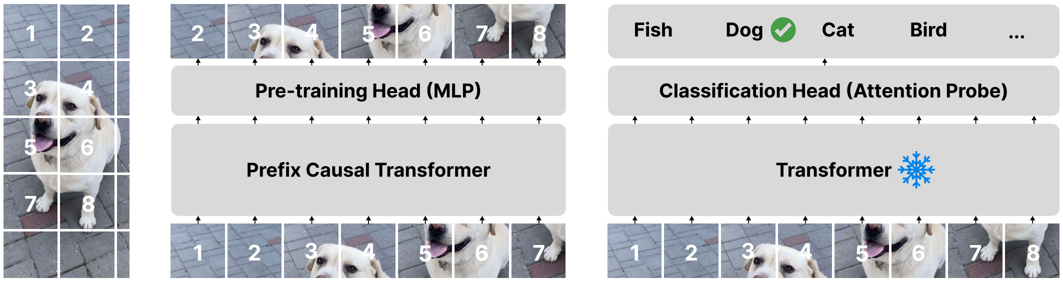

1
2
3
4
5
6
7
8
Pre-training Head (MLP)
Prefix Causal Transformer
Fish
Dog
Cat
Bird
...
Classification Head (Attention Probe)
Transformer

Figure 1: **NARAIM approach.** The input is divided into patches in row-major order, which are then processed by a vision transformer. The pre-training head utilizes the transformer's output to predict the next token based on the preceding ones. Meanwhile, the classification head, implemented with an attention probe, uses the transformer's output to predict a class.

Thus, in this paper, we propose Native Aspect Ratio Autoregressive Image Models (NARAIM), an AIM model that utilizes native aspect ratio inputs [6, 7, 10]. By preserving the aspect ratio during pre-training, our approach improves downstream classification accuracy without increasing computational costs, as the total number of input tokens remains constant. Additionally, since images with varying aspect ratios require positional embeddings that adapt to different image layouts, we highlight the advantages of fractional positional embeddings in the pre-training process [6]. Finally, we demonstrate that random cropping while preserving the native aspect ratio serves as an effective regularizer, outperforming the standard random resized crop method in the original AIM models, particularly on inputs with highly non-square aspect ratios.

2 Autoregressive Image Modeling with Native Aspect Ratios

In this section, we introduce NARAIM: Native Aspect Ratio Autoregressive Image Models, a variant of Autoregressive Image Models (AIM) that maintains the native aspect ratio of input images. AIM divides images into patches using a raster (row-major) ordering, which are subsequently processed by a ViT [8]. Using a causal attention mask, the objective of the ViT is to predict the next image patch in pixel space, which results in a strong representation model which can be fine-tuned for downstream applications. For a consistent shape, AIM resizes and crops images changing their aspect ratio, thus distorting potentially crucial image structures (see Figure 2). In NARAIM, we remove these distortions by using an aspect ratio preserving resize, as detailed below.

Native Aspect Ratio Resize. Instead of forcing an image into a square shape of exactly 224^2 pixels, we rescale the input image to an overall pixel number of approximately 224^2, while keeping the aspect ratio unchanged. Next, we take a crop from the top-left such that the height and the width are some integer multiple of the patch size. After cropping, we split the image into square patches of size 14 using a raster (row-major) ordering, which results in at most 256 patches. The patches are then processed by the ViT and used to predict the next patch in pre-training, as shown in Figure 1, or other objectives like classification in downstream tasks later. Figure 2 illustrates the difference between the traditional pre-processing of inputs in square-sized images and the native aspect ratio resize. As images of different shapes can still produce small variations in the token number, we add padding tokens where needed, and mask them in the loss calculation.

Downstream Adaptation. When the pre-training is completed, the model generates a feature vector for each patch. In order to evaluate the quality of these features, we freeze the ViT backbone and replace the pre-training head with a trainable attentive classification probe. We chose to use an attentive probe instead of a linear probe, as El-Nouby et al. [10] found that the attentive probe performs substantially better. Moreover, we keep the input format to the models consistent, i.e., a model that was pre-trained on native aspect ratio inputs will also be used to classify native aspect ratio images. To more easily adapt the model to downstream tasks when causal attention is not needed anymore we randomly sample a prefix causal attention mask during pre-training, similar to El-Nouby et al. [10]. We refer to Appendix D for more details.

Augmentations. The original AIM applied a random resized crop augmentation during training, which crops the image and changes its aspect ratio. While this augmentation helps to reduce

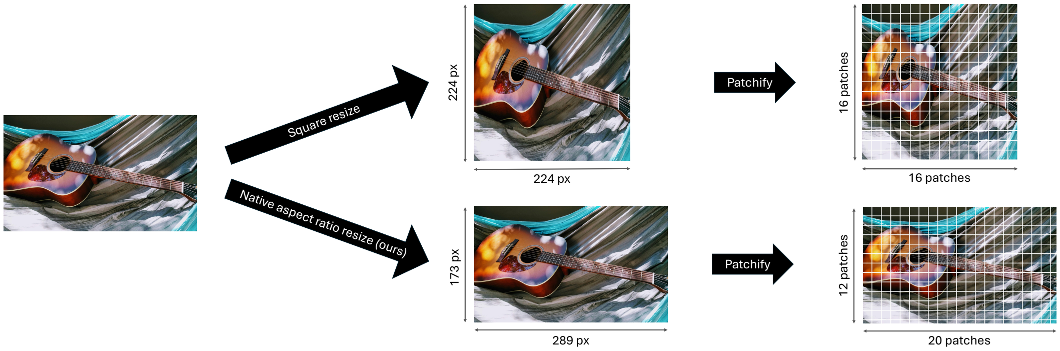


Figure 2: **Native aspect ratio resize.** Given a crop, it is common to resize it to a fixed-sized square. Since the image is going to be patchified and fed to a transformer, and the transformer itself is agnostic to the spatial organization of the patches, we propose keeping the native aspect ratio. First, we reshape the image keeping the aspect ratio fixed, ensuring the total number of pixels does not exceed 224^2. Then, we patchify the image, obtaining at most 256 patches.

overfitting, the distortions negatively impact the quality of the representations learned, as we show in Section 3. To maintain the regularization benefit, we adapt this augmentation to NARAIM by applying a random crop with the same aspect ratio as the original model, as explained in Figure 2. In addition, we impose the constraint that the crop needs to contain at least 224^2 pixels to prevent crops that would require severe upsampling. Additionally, we apply a random horizontal flip during training. During inference and downstream evaluation, we just take a native aspect ratio crop for NARAIM. For AIM, we follow El-Nouby et al. [10] and first resize the image so that the shortest side is 256 pixels, and then we take a center crop with a side length of 224 pixels.

Positional Embeddings. Positional embeddings play a significant role in helping the model understand the spatial location of the patches. We experiment with two different ways of calculating the positional embeddings: absolute and fractional [6]. The key difference is that absolute positional embeddings use the *index* of the patch along the horizontal and vertical axes, whereas fractional positional embeddings use the *proportion*, that is, the index divided by the number of patches along that direction. For the absolute positional embedding, we take the height and width *indices*, encode them using the fixed transformation from Vaswani et al. [19], and concatenate both representations. For the fractional variant, we get the height and width *proportions*, encode them using a learned dense layer, and sum them as in Dehghani et al. [6]. A mathematical formulation of how to compute these two positional embeddings can be found in Appendix C.

We anticipate that absolute positional embeddings may cause problems when tested on images whose aspect ratio is out-of-distribution, whereas fractional positional embeddings are likely to generalize more easily. Moreover, fractional positional embeddings enable the model to recognize when a patch is near the edge (since the proportion will approach 1), whereas absolute positional embeddings do not provide this information, as the total number of patches is unknown.

Loss Functions. We use two different loss functions for the pre-training head: with and without patch normalization. Let $\mathbf{x}_i \in \mathbb{R}^{3P^2}$ be the i^{th} patch, where P is the height and width of a patch. Then, with normalization, as proposed by He et al. [11], the i^{th} prediction target $\mathbf{y}_i \in \mathbb{R}^{3P^2}$ is $\mathbf{y}_i = (\mathbf{x}_{i+1} - \mu(\mathbf{x}_{i+1}))/\sigma(\mathbf{x}_{i+1})$. Without normalization, the prediction target is simply $\mathbf{y}_i = \mathbf{x}_{i+1}$.

The loss function is the mean-squared error (MSE) between the predicted and ground-truth input: we sum up the squared differences per subpixel and divide by the total number of subpixels in the patch. For fine-tuning on the classification task, we use a standard cross-entropy loss function.

3 Results and Discussion

To demonstrate the benefits of native aspect ratios in NARAIM, we compare NARAIM to AIM on ImageNet-1k [7], used for both the pre-training and the downstream tasks as in the small-scale studies of El-Nouby et al. [10]. Our model is implemented in JAX/Flax [3, 12] and trained on a single

Table 1: Validation MSE and accuracy of the pre-training and classification heads on ImageNet-1k with a ViT-B/14 backbone. The fine-tuning was repeated with four different seeds, and we report the mean and standard deviation. Using native aspect ratios in NARAIM improves the downstream classification accuracy over the original AIM [10] model. The results also highlight the importance of fractional embeddings, random crop augmentations, and normalization. In the first column, plus and minus signs are relative to the original NARAIM model (Appendix B).

Model	**Next-Token MSE**	**Class Accuracy**
AIM	0.340	54.7 ± 0.6
NARAIM (ours)	0.357	55.4 ± 0.1
+ Fractional embedding	0.354	56.0 ± 0.1
+ RandomCrop	0.354	$\mathbf{56.8} \pm 0.1$
- Normalization	0.010	52.6 ± 0.1

40GB Nvidia A100 GPU. Due to limited resources, our backbone is a ViT-B/14 with roughly 86M parameters and hence a scaled-down implementation of the original AIM model which had more than 600M parameters. Prior research [6] has shown that scaling models trained on native aspect ratio results in improved performance. We plan to verify that this scaling law extends also to our results in future work. Appendix B details the hyperparameters, and we make our code publicly available[2].

The results in Table 1 demonstrate that NARAIM outperforms AIM in downstream performance. The best NARAIM model achieves a validation accuracy of 56.8 ± 0.1, surpassing AIM by more than two percentage points. While the pre-training validation MSE is lower for the AIM model, the ultimate goal of these models is to create useful representations for downstream tasks. To measure the usefulness of representations for downstream tasks, the downstream validation accuracy is a more useful proxy than the pre-training validation MSE. In Appendix E, we visualize the MSE across different patch locations in the image. Both methods show similar MSE patterns, with the borders being most difficult due to the patch discontinuity.

Ablations. As for the ablation studies, we find that using fractional embeddings provides a small accuracy increase over absolute embeddings, likely due to the model with fractional embeddings being slightly better at classifying images whose aspect ratio is out-of-distribution. Using random crops substantially increases the downstream validation accuracy, which is explained by the fact that the base NARAIM model overfits during pre-training: thus, random crops alleviate this issue. Finally, looking at patch-wise normalization, we note that the pre-training MSE is on a different scale, and therefore cannot be compared. As for the downstream validation accuracy, we note that patch-wise normalization is very important. One intuitive explanation is this: if patches are not normalized, the pre-trained model will be geared towards detecting global patterns. Hence, the classifier may develop shortcuts such as predicting "boat" whenever it detects a blue patch, decreasing the accuracy.

Effect of Aspect Ratio. To gain further insights into the improved performance, we group images by their aspect ratios and visualize the classification accuracy per group in Figure 3. First, we note that results on the bins with aspect ratio $(0, 1/2)$ and $(2, \inf)$ are slightly noisy due to only containing several hundred images (see Appendix A for more information). Nonetheless, we observe that the accuracies for the NARAIM models are higher than that of the AIM model for every bin. We also note that the performance of the AIM model decreases when the model is applied to inputs with non-square aspect ratios, showing its focus on square-shaped images. The "NARAIM" and "NARAIM + Fractional" models showcase slightly noisy behavior over the bins, which may be attributed to the earlier observation that they overfit on the training set due to a lack of data augmentation.

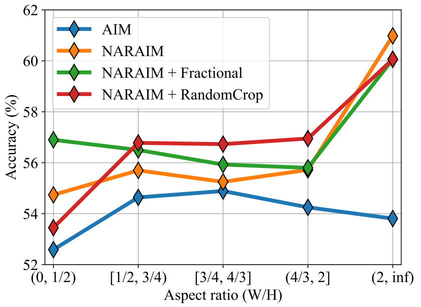


Figure 3: The classification accuracy over image aspect ratios. NARAIM improves across all aspect ratios.

In comparison, the performance for "NARAIM + RandomCrop" model is remarkably stable in the middle three bins. Furthermore, we find a significant performance improvement for strongly

[2] https://github.com/daniel-gallo/naraim

horizontal images, though the precise gap may be noisy due to limited samples in this bin. Overall, we conclude that our “NARAIM + RandomCrop" model is better at dealing with images with non-square aspect ratios than the “AIM" model.

Conclusion. In this paper, we have introduced NARAIM, a vision transformer model pre-trained with an autoregressive objective that uses images in their native aspect ratio. We showed that using native aspect ratios improves the downstream classification performance, in particular for non-square images, emphasizing the importance of modeling images in their original aspect ratio in autoregressive image models. In future work, we plan to scale our model to 600 million parameters to benchmark it against the original AIM and validate its benefits at scale, as well as extending the downstream evaluation to more tasks.

References

[1] Randall Balestriero and Yann LeCun. 2024. How Learning by Reconstruction Produces Uninformative Features For Perception. In *Forty-first International Conference on Machine Learning*.

[2] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. 2024. Revisiting Feature Prediction for Learning Visual Representations from Video. *Transactions on Machine Learning Research*. Featured Certification.

[3] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs. 2018.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models Are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

[5] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. 2023. Scaling Vision Transformers to 22 Billion Parameters. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7480–7512. PMLR.

[6] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey Gritsenko, Mario Lucic, and Neil Houlsby. 2023. Patch n’ Pack: NaViT, a Vision Transformer for any Aspect Ratio and Resolution. In *Advances in Neural Information Processing Systems*, volume 36, pages 2252–2274. Curran Associates, Inc.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The LLaMA 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

[10] Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Ángel Bautista, Vaishaal Shankar, Alexander T Toshev, Joshua M. Susskind, and Armand Joulin. 2024. Scalable Pre-training of Large Autoregressive Image Models. In *Forty-first International Conference on Machine Learning*.

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked Autoencoders Are Scalable Vision Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988.

[12] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX. 2023.

[13] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

[14] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361*.

[15] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

[16] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking Multimodal Understanding Across Millions of Tokens of Context. *arXiv preprint arXiv:2403.05530*.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[18] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

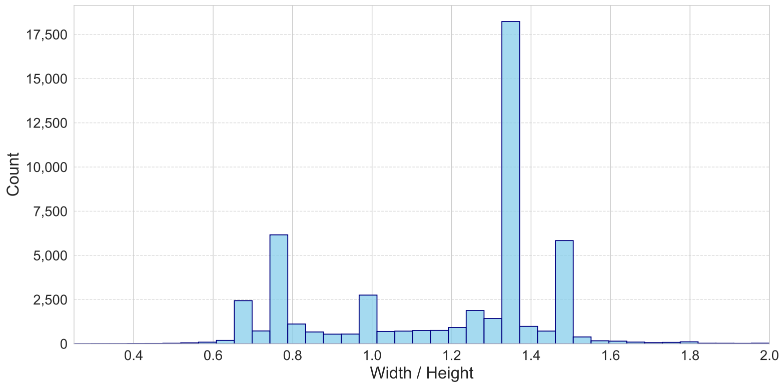


Figure 4: **Aspect ratio distribution.** This histogram shows the aspect ratios of the ImageNet-1k validation set. Most of the images are in landscape orientation, but we can observe three modes corresponding to portrait, square, and landscape images.

A Aspect ratio distribution

We use ImageNet-1k to train our models, both during pre-training and downstream classification. Figure 4 illustrates the aspect ratio of the images in the validation set.

B NARAIM specifications

The baseline NARAIM model uses absolute positional embeddings, patch-wise normalization, and no random cropping. In Table 2 we show the hyperparameters of the backbone and in Table 3 we show the training parameters.

Table 2: Hyperparameters of the backbone (ViT-B/14).

Name	**Value**
Patch size	14
Number of layers	12
Number of heads	12
Embedding dimension	768
Hidden dimension	3,072

Table 3: Training parameters (very similar to El-Nouby et al. [10]).

Parameter	**Pre-training value**	**Fine-tuning value**
Optimizer	AdamW	AdamW
Optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.98$	$\beta_1 = 0.9, \beta_2 = 0.999$
Peak learning rate	$1e^{-3}$	$1e^{-3}$
Minimum learning rate	0	$1e^{-5}$
Weight decay	0.01	0.1
Batch size	512	512
Patch size	14×14	14×14
Gradient clipping	1.0	3.0
Decay rate (for lr scheduler)	0.1	-
Warmup iterations	5,000	500
Cooldown iterations	10,000	-
Total iterations	500,000	50,000
Learning rate schedule	exponential	cosine decay

C Positional embeddings

To calculate the fractional embedding for a patch $\mathbf{x}_i$, let h_i, w_i be the vertical and horizontal indices of the patch $\mathbf{x}_i$ in the input, with H, W the total number of vertical and horizontal patches in the input. Then, the fractional embedding for the patch is $f(h_i/H) + g(w_i/W)$, where f and g are learnable one-layer perceptrons.

Regarding the absolute embeddings, let h_i, w_i again be the vertical and horizontal indices, and let ϕ be the function introduced in the original Transformer paper [19]:

$$\phi(\text{pos}, 2i) = \sin(\text{pos}/10000^{2i/d}),$$

and

$$\phi(\text{pos}, 2i+1) = \cos(\text{pos}/10000^{2i/d}),$$

where $d = \frac{d_{\text{model}}}{2}$. The complete positional embedding is then:

$$\phi\left(h_i, 1\colon d\right) \oplus \phi\left(w_i, 1\colon d\right),$$

where $\oplus$ denotes concatenation.

D Prefix causal attention

Adapting a model trained with causal attention to downstream tasks may cause issues: due to the attention mask, the model only learns to create representations using causal attention. This means that for a token $\mathbf{x}_i$, only the information from tokens $\{\mathbf{x}_1, ..., \mathbf{x}_{i-1}\}$ will be incorporated. However, in downstream tasks, every token should be able to attend to every other token.

Prefix causal attention addresses this issue by selecting a random integer n between 1 and $N-1$, where N is the number of tokens in an input [15]. Then, for tokens $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$, the causal attention mask is dropped, and every token in this set can attend to every other token in the set. For the remaining tokens, $\{\mathbf{x}_{n+1}, ..., \mathbf{x}_N\}$, we use the standard causal attention mask. Predictions for tokens $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ are not included in the loss calculation, as the tokens could be trivially predicted. Figure 5 displays the prefix causal attention for pre-training and fine-tuning.

El-Nouby et al. [10] trained networks with both prefix causal attention and regular causal attention, and found that prefix causal attention substantially improved the performance on downstream tasks.

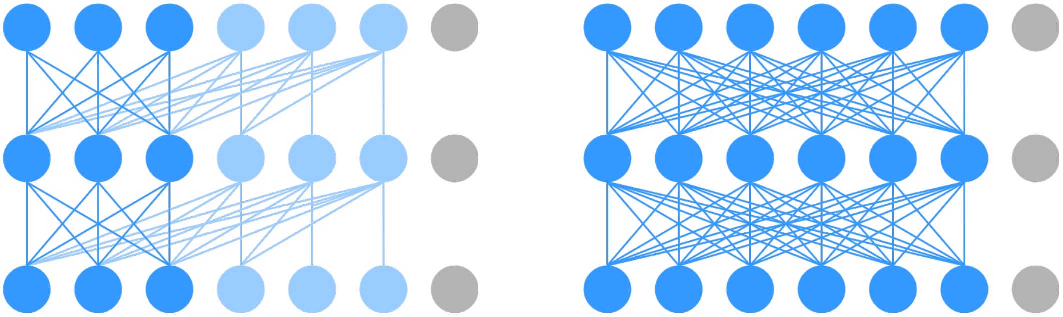
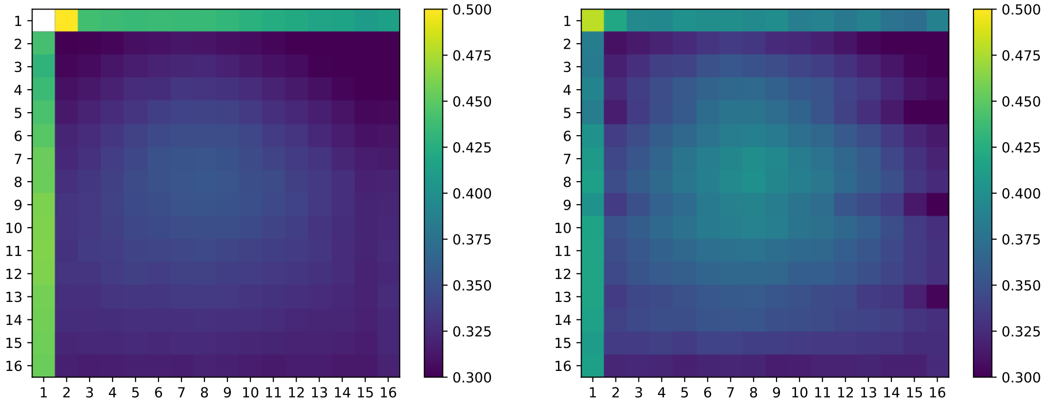


Figure 5: **Prefix causal attention.** For pre-training (left), we uniformly sample a prefix length n during pre-training (e.g., $n = 3$). The attention for the first n patches is set to be bidirectional and no loss will be computed for them. The rest of the patches adopt a causal mask and their loss is calculated. During fine-tuning to a downstream task (right), the mask is discarded. The gray patches represent the padding, which are added for reasons explained in Section 2.

E Raster patterns across patches

While El-Nouby et al. [10] reports the validation MSE per-row, we report it per-patch in Figure 6. This is easy to compute for AIM, that always has 256 patches. The first one is never predicted, which is why it is white on Figure 6a.



(a) AIM

(b) NARAIM with fractional embeddings

Figure 6: **Per-patch validation MSE.** In both models we see that predicting the first row and the first column is harder. Predicting the first row is hard because there is no information about the previous line. Predicting the first column is challenging because these patches are not correlated with the previous ones due to the carriage return.

For NARAIM, we first map the patch index to the interval $[0, 1]$, and then we discretize this into 16 bins. The first patch is never predicted, but the second patch can lie in the first bin if the image was horizontal. Hence, the white square of Figure 6a is not present in Figure 6b.