xT: Nested Tokenization for Larger Context in Large Images

Ritwik Gupta [*1] **Shufan Li** [*2] **Tyler Zhu** [*13] **Jitendra Malik** [1] **Trevor Darrell** [†1] **Karttikeya Mangalam** [†1]

Abstract

Modern computer vision pipelines handle large images in one of two sub-optimal ways: down-sampling or cropping. These two methods incur significant losses in the amount of information and context present in an image. There are many downstream applications in which global context matters as much as high frequency details, such as in real-world satellite imagery; in such cases researchers have to make the uncomfortable choice of which information to discard. We introduce *xT*, a simple framework for vision transformers which effectively aggregates global context with local details and can model large images end-to-end on contemporary GPUs. We select a set of benchmark datasets across classic vision tasks which accurately reflect a vision model's ability to understand truly large images and incorporate fine details over large scales and assess our method's improvement on them. *xT* is a streaming, two-stage architecture that adapts existing vision backbones and long sequence language models to effectively model large images without quadratic memory growth. We are able to increase accuracy by up to 8.6% on challenging classification tasks and F_1 score by 11.6 on context-dependent segmentation on images as large as 29,000 x 29,000 pixels.

Code and pre-trained weights are available at https://github.com/bair-climate-initiative/xT.

1. Introduction

As camera technology has advanced, images have been getting increasingly larger over the past decade. Images captured by sensors on smartphones now capture images at 4K resolution (roughly 8.3M pixels) while professional DSLR cameras capture images at 8K resolution. Elsewhere, sensors on satellites and microscopes capture images with over a billion pixels.

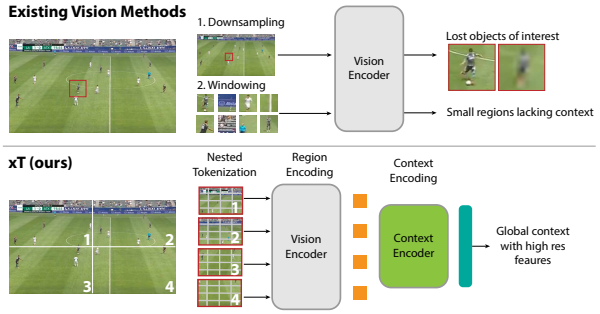

Figure 1: *xT* allows large images to be modeled end-to-end on contemporary GPUs **without compromising on high frequency features or global context.**

Modern computer vision pipelines are limited by the memory in the systems they are trained upon, resulting in the creation of models that only operate on small images. Computer vision practitioners limit the size of images in two less-than-ideal ways: down-sampling or cropping. While these simple operations produce powerful models when measured against typical computer vision benchmarks, the loss of high frequency information or global context is limited for many real-world tasks.

Consider a video feed of a football game. Captured natively in 8K resolution, a model attempting to answer the question of where a player on the left side of the screen will pass the ball to on the right side of screen will not be able to reason over the entire image in one pass. The image, the downstream model, and all intermediate tensors cannot fit in the memory of modern, large VRAM GPUs. A common approach is to process the image by treating it as individual "windows", each fed through the model without sharing context, resulting in sub-optimal performance.

We introduce *xT*, a streaming, two-stage framework by which myopic vision backbones can effectively integrate local and global context over large images without incurring quadratic memory growth. In particular, we tackle both issues of increasing GPU memory utilization and the integration of context across very large images. We achieve this by introducing token hierarchies to state-of-the-art vision backbones (Liu et al., 2021; Ryali et al., 2023) and imbuing

[*]Equal contribution [1]University of California, Berkeley [2]University of California, Los Angeles [3]Princeton University. Correspondence to: Ritwik Gupta <ritwikgupta@berkeley.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

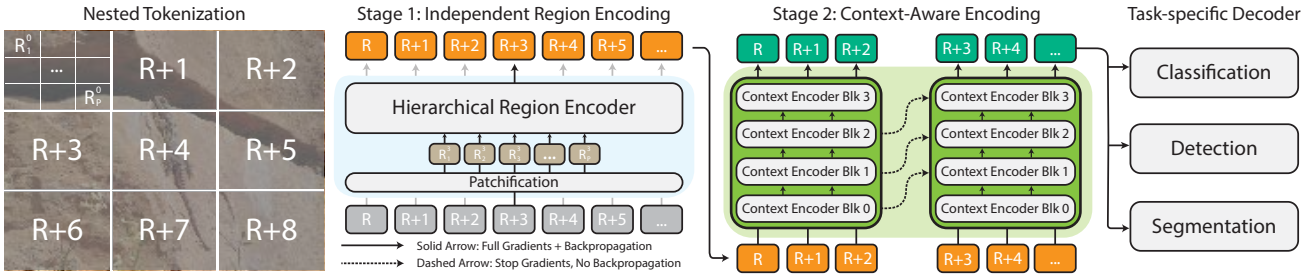

Figure 2: **Overview of our methodology.** *xT* provides a way for existing vision backbones trained on small images to work effectively with large images. The key is our **nested tokenization** of large images at multiple levels: at the region level as input $R^0, \ldots, R^8$ ($R, \ldots, R+8$ for readability) for the region encoders, and then at the patch level $R^i_0, \ldots, R^i_{P-1}$ inside the encoders to understand local details. The image regions then undergo **independent, hierarchical encoding**, by passing through a vision backbone that serves as a region encoder. Hierarchical region encoders result in down-sampled features which, when combined with **context encoders**, allows us to process more regions at once than typically possible. One such context encoder, Transformer-XL, is illustrated in Stage 2. It recurrently processes previous prior sequence tokens using cross attention, extending its context range significantly with depth. The resulting sequence has assimilated both local and global context and is finally fed to a **task-specific decoder.**

the resulting local features with global context through the use of long-sequence models, such as Transformer-XL and Mamba (Dai et al., 2019; Gu & Dao, 2023), obtained from the field of natural language processing. *xT* matches and beats the performance of competitive large image architectures on multiple downstream tasks that require large visual contexts such as segmentation, detection, and classification. We demonstrate results on a variety of downstream tasks and achieve up to an 8.6% gain in accuracy on classification tasks and an 11.6 increase in F_1 score on context-dependent segmentation.

2. Related Works

Modeling large images Many prior works have attempted to model large images. These approaches fall into one of two buckets: 1) multi-pass hierarchical or cascading approaches and 2) sliding windows combined with some suppression mechanism. Dalal and Triggs famously utilized a sliding window approach for robust object recognition in their work on histograms of oriented gradients (Dalal & Triggs, 2005). R-CNN (Girshick et al., 2014) classified proposals generated by selective search in a cascade, resulting in a slow, albeit effective detector that worked for large images. Gadermayr, et. al. (Gadermayr et al., 2019) proposed a cascading convolutional neural network (CNN) where regions of interest (RoI) are first segmented using low resolution imagery. These RoI are used to refine the segmentation mask using high resolution imagery. (Zhang et al., 2021) implement a multi-scale vision transformer architecture which outputs a feature pyramid combined with a linear attention mechanism.

As demonstrated by prior work, there are inherent benefits to using hierarchical backbones when attempting to represent multi-scale features, so all of our experiments are done with them. The *xT* framework is agnostic to the style of vision backbone used for feature extraction.

Assessing global understanding Methods that aim to improve the handling of high-resolution or large images rarely benchmark their methods on real-world, large image datasets. For example, both (Zhang et al., 2021) and (Yang et al., 2022) are ViT architectures with non-global attention to better model large images. However, both works experiment on datasets such as ImageNet which is comprised of images of 224×224 pixels large.

Other evaluation methods rely on naïve data processing techniques such as cropping medium-sized images from Cityscapes into smaller chunks and independently stitching the outputs together (Gu et al., 2021), or center-cropping a 224×224 patch from a larger (yet objectively small) ImageNet image (Wang et al., 2021). Ultimately, these approaches are sensitive to changes in test-time resolution, though representation learning methods exist to rectify this issue (Reed et al., 2023).

Real-world classification and segmentation datasets that are heavily dependent on their surroundings, or in cases where the object that needs to be modeled is a small fraction of the overall image, would benefit from better integration of global context. Myopic vision encoders which utilize a windowing approach may fail at large objects in detection tasks as objects will span multiple windows.

We evaluate on real-world datasets such as xView3-SAR (Paolo et al., 2022), a dataset where the average image is $29,400 \times 24,400$ pixels large, and iNaturalist

2018 (Van Horn et al., 2018), where the entire 800×800 image is utilized at once for evaluation. We also evaluate on the standard Cityscapes dataset for direct comparison on a common vision benchmark.

Learning global context Convolutional neural networks operate over images in sliding windows defined by the size of the kernels used for convolution. However, convolutions do not inherently aggregate information from surrounding windows. Following seminal work by Hubel and Wiesel (Hubel & Wiesel, 1962) and Koenderink and Van Doorn (Koenderink & Van Doorn, 1999) on feature pooling in the visual cortex, LeCun, et. al. (LeCun et al., 1989) introduced "average pooling" into CNNs as a context aggregation mechanism. Dilated convolutions as implemented by Yu and Kolun (Yu & Koltun, 2016) further increased the receptive field of CNNs.

Some models treat memory as an explicit, external component of the model. Neural Turing machines and memory networks (Graves et al., 2014; Weston et al., 2015) learn an access mechanism to read/write from a fixed-size memory matrix that is external to the backbone itself. More recently, MeMViT (Wu et al., 2022) extends this concept to sequences of images by caching and compressing activations from prior sequences with attention as a learned access mechanism.

Transformers and vision transformers (Vaswani et al., 2017; Dosovitskiy et al., 2020), consisting of stacked attention layers (Bahdanau et al., 2016), are able to maintain context over a fixed number of tokens. Prior works attempt to address limitations in attention via approaches such as factorization and adaptive masking (Child et al., 2019; Sukhbaatar et al., 2019). Notably, the Transformer-XL (Dai et al., 2019) efficiently passes context over a large number of tokens by recycling state over recurring input segments.

Recurrent neural networks (RNNs) (Hopfield, 1982; Rumelhart & McClelland, 1987) addressed context over long sequences by recurrently passing hidden states over a fixed set of time steps but suffered from catastrophic foregetting (Hochreiter, 1991). Subsequent work on long short-term memory (Hochreiter & Schmidhuber, 1997) tackled this issue via gating. Structured state space models (SSMs) (Gu et al., 2022) have re-emerged recently as a viable successor to RNNs. Of the SSM family of models, the Mamba (Gu & Dao, 2023) architecture presents a selection mechanisms for state spaces allowing it to generalize to large sequence lengths.

xT is a streaming, two-stage architecture in which a powerful vision backbones extracts features from regions of a large image in batches. These features are streamed to a shallow context encoder which integrates global context across local features, effectively increasing the receptive field of the vision backbone across the entire image while staying within memory and parameter limits.

3. Background

In this section, we briefly summarize the needed background for methods used in our work.

3.1. Long-Context Models as Context Encoders

xT utilizes long-context models originally designed for text in order to mix information across large images. These methods extend the context length beyond the typical limit of transformers. Below we briefly review two techniques which we build upon as our context encoders: Transformer-XL (Dai et al., 2019) and Mamba (Gu & Dao, 2023).

Transformer-XL uses recurrence to pass prior information to future windows via prior hidden states. This effect propagates through depth, so an N-layer transformer capable of taking a length L sequence can be easily extended to handle a sequence of length NL.

Each hidden state $\mathbf{h}_\tau^n$ of layer n for sequence τ is computed from the previous layer hidden states $\mathbf{h}_{\tau-1}^{n-1}$ and $\mathbf{h}_\tau^{n-1}$ as

$$\begin{aligned} \tilde{\mathbf{h}}_\tau^n &= [\mathrm{SG}(\mathbf{h}_{\tau-1}^{n-1}) \circ \mathbf{h}_\tau^{n-1}] \\ \mathbf{q}_\tau^n, \mathbf{k}_\tau^n, \mathbf{v}_\tau^n &= \mathbf{h}_\tau^{n-1} W_q^\top, \tilde{\mathbf{h}}_\tau^n W_k^\top, \tilde{\mathbf{h}}_\tau^n W_v^\top \\ \mathbf{h}_\tau^n &= \mathrm{Transformer}(\mathbf{q}_\tau^n, \mathbf{k}_\tau^n, \mathbf{v}_\tau^n) \end{aligned} \tag{1}$$

where SG stands for a stop gradient. This is the same as the original Transformer, except that the keys and values $\mathbf{k}_\tau^n, \mathbf{v}_\tau^n$ are computed using the previous sequence's hidden state $\mathbf{h}_{\tau-1}^{n-1}$ in addition to the current sequence's hidden state $\mathbf{h}_\tau^{n-1}$ using cross attention. This mechanism allows for the recurrence of the hidden states $\mathbf{h}_\tau^n$ across layers. The application of a stop gradient between sequences lets information be propagated without suffering the memory costs incurred with full sequence backpropagation.

State Space Models State space models (Gu et al., 2022; Nguyen et al., 2022) have been re-discovered recently as a potential replacement for transformers in long-sequence modeling. These models can be formulated as ordinary differential equations of the form

$$\begin{aligned} \frac{dh(t)}{dt} &= Ah(t) + Bx(t) \\ y(t) &= Ch(t) + Dx(t) \end{aligned} \tag{2}$$

where $x(t) \in \mathbb{R}$ is the input signal and $y(t) \in \mathbb{R}$ is the output signal. Practically, this is computed through a dis-

cretization of the ODE via the zero-order hold (ZOH) rule:

$$\bar{A} = \exp(\Delta A)$$
$$\bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B$$
$$h(t) = \bar{A}h_{t-1} + \bar{B}x_t \quad (3)$$
$$y_t = Ch_t \quad (4)$$

Mamba (Gu & Dao, 2023) is a new state space model that introduces a selective scan mechanism that allows time-varying parameterizations. Mamba theoretically carries context across very long sequences without any loss in accuracy and is implemented efficiently using custom CUDA kernels.

3.2. Linear attention mechanism

A standard transformer block with multi-headed self attention requires quadratic memory with respect to sequence length L for fully global context. This is not ideal in the face of limited GPU memory. HyperAttention (Han et al., 2023) is an attention mechanism with near-linear complexity with respect to sequence length. It reduces the complexity of naive attention by first finding large entries of the attention matrix using Locality Sensitive Hashing (LSH). These dominant entries, combined with another randomly sampled subset from the matrix, are then used to approximate output of naive attention. This approach is particular helpful when the long range context correspondences are sparse.

4. Methodology

Our goal for *xT* is to demonstrate a simple framework for allowing existing methods to process large images in a memory efficient and context-preserving manner. We achieve this through a streaming, two-stage architecture. First, images are tokenized in a multi-stage hierarchy (Section 4.1) into regions. These regions are encoded by a powerful, but myopic, vision encoder in batches (at worst case, in serial) (Section 4.2), resulting in a set of local features for the entire image. A lightweight context encoder then integrates information across all local features, culminating in a feature vector which contains contextualized global semantics over the entire image (Section 4.3). This feature vector is then used for task-specific decoding. Our overall pipeline is illustrated in Figure 2.

4.1. Nested Tokenization

Given a large input image of shape $\alpha H \times \beta W$, we first subdivide the image into $H \times W$ regions so that our region encoder can adequately process them. Each region R^i is further patchified into P patches, $R^i_0, \ldots, R^i_{P-1}$, by the region encoder backbone in order to extract features for each region. The regions are non-overlapping and zero-padded in instances when the region size, $H \times W$, does not evenly divide the image size.

Typically our images and regions are square, so we use a simplified notation to denote our pipeline parameters. We refer to a pipeline which receives images of size $\alpha R \times \alpha R$ and subdivides them into $R \times R$ regions as an $\alpha R/R$ setup. Standard setups are $512/256$, or $4096/512$, in which we split our image into 2×2 and 8×8 tiles respectively.

4.2. Region Encoder

The region encoder is any vision model which has been previously trained on small images small images $H \times W$, usually 224×224 or 256×256. The region encoder independently generates feature maps for each region $R^i_{1\ldots P}$. In our experiments, we utilize vision transformers which output a shorter sequence length than which is input to them. These sequence lengths are less than the equivalent length produced by isotropic ViTs (Dosovitskiy et al., 2020). In this setup, we are able to effectively handle images with an increased number of regions, as our sequence length is reduced by $4\times$ or greater. However, we also demonstrate the effectiveness of *xT* with a CNN-based encoder in the Appendix.

We stream regions through the region encoder in batches when GPU memory allows. However, either when the image is too large such that all of its constituent regions cannot fit into GPU memory, or when the regions themselves are too large, we process the image sequentially. The features generated from this process contain information limited to each region and are concatenated in row-major order to form a sequence.

4.3. Context Encoder

The context encoder is a lightweight sequence-to-sequence model that is able to effectively attend to long sequences. In *xT*, the context encoder plays the key role of disaggregating global context across local features extracted using the region encoder. Critically, we constrain the context encoder to have a near-linear memory and time cost. This design allows *xT* to "see" many other regions of the large image which otherwise would not be feasible with the naive usage of a vision model. *xT*'s usage of a context encoder significantly extends the receptive field of existing vision models with a marginal increase in memory cost and the number of parameters.

Our method acts primarily in two settings: when our sequence of region features fits entirely within the context encoder's context length, and when it does not. In the first, we simply process everything at once. We use standard 2D positional embeddings which are added to the nested region features.

We experiment with three context encoders with linear "attention" mechanisms: a LLaMA-style (Touvron et al., 2023)

architecture using HyperAttention (referred to as *Hyper*), and Mamba. These two settings are called *⟨xT⟩ Hyper* and *⟨xT⟩ Mamba*, where ⟨xT⟩ is an operator joining the choice of region encoder with context encoder.

When our input sequence does not fit in the entire context length, we need to additionally compress our regions to maintain contextual information for future regions. We experiment with a derivative of Transformer-XL that utilizes HyperAttention, a form of linear attention, and absolute positional embeddings instead. We denote this setting as *⟨xT⟩ XL*. At the time of writing, Mamba does not have an efficient kernel for implementing an XL-style recurrence, so we cannot apply Mamba to this setting.

Effective Receptive Field Calculations Vision models used as-is are able to have an effective receptive field across the entire image but have zero reception on large images as they simply run out of memory. We aim to demonstrate the effective receptive field of *xT*'s two-stage architecture. In summary, the receptive field of *xT* is global but with minor aliasing artifacts due to an effective convolution over the image with the region encoder.

We provide calculations of the effective receptive field for the Swin ⟨xT⟩ XL setup. This is further ablated in Section 7.1.

In the context encoder, we concatenate the features of C regions into one "chunk" as our features are both smaller than the inputs from the hierarchical regional encoder. The Transformer-XL context encoder additionally has reduced attention memory requirements from using HyperAttention, further improving the number of regions that fit into one "chunk". Each region attends to all other regions in this chunk, and they also have access to the previous chunk's hidden states for more context flow through the model. Context scales as a function of depth in Transformer-XL.

Consequently, we can calculate the context enhancement we achieve beyond just a standard model query on a small image. If we use a region encoder with input size R (typically 256) and receive a large image of size $\alpha R \times \beta R$, then we will have a total of $\alpha\beta$ regions available for our context encoder. Then we increase our context from increased chunk sizes by a factor of C, and also increase it from our recurrent memory by a factor of N, the depth of the context model. These values are calculated in Table 1 and further visualized in Figure 4.

In total, our context is multiplied by $\alpha\beta NC$. However, note that there is a trade-off between $\alpha\beta$ and C, as increasing the size of our input image limits the chunk sizes that we can create from the region features.

Model	Input (px)	Region (px)	XL Layers	Context
Swin-B	256	256	-	65,536
Swin-B	512	512	-	65,536
Swin-B ⟨xT⟩ XL	512	256	1	131,072
Swin-B ⟨xT⟩ XL	512	256	2	196,608
Swin-B ⟨xT⟩ XL	4096	256	2	786,432

Table 1: The effective context length (in pixels) is calculated for Swin-B versus Swin-B ⟨xT⟩ XL. Context is fixed for Swin while it grows as a function of region size and the number of Transformer-XL layers for Swin ⟨xT⟩ XL.

5. Experiments

Settings We aim to demonstrate the efficacy of *xT* across the most common vision tasks: classification, detection, and segmentation. To do so, we pick a representative and challenging benchmark dataset in each domain for experimentation.

We focus on iNaturalist 2018 (Van Horn et al., 2018) for classification, xView3-SAR (Paolo et al., 2022) for segmentation, and Cityscapes (Cordts et al., 2016) for detection. Since iNaturalist 2018 is a massive dataset, we focus on the *Reptilia* super-class, the most challenging subset available in the benchmark (Van Horn et al., 2018). xView3-SAR is a difficult segmentation dataset for two reasons: it is comprised of extremely large images that are 29,400×24,400 pixels large on average, and the objects in the dataset are heavily influenced by their non-local surroundings. Lastly, Cityscapes is used for evaluation of detection and to serve as a common baseline across prior work.

Metrics We measure top-1 accuracy for classification. For detection, we measure mean average precision (mAP) along with $\text{mAP}_{\text{Large}}$. $\text{mAP}_{\text{Large}}$ is of specific interest since large objects in large images cross multiple region boundaries, resulting in a challenging training and evaluation problem. Segmentation is measured in two ways: aggregate F_1 score for segmentation of any object in the image, and close-to-shore F_1 score for objects that are "close to the shoreline". xView3-SAR is comprised of synthetic aperture radar images which demonstrate unique artifacts around busy areas not found in typical vision benchmarks. Particularly, objects within 2km of the shoreline are impacted heavily by a shoreline that may not be visible in crops around the object.

5.1. Classification

We utilize the SwinV2 (Liu et al., 2022) and Hiera (Ryali et al., 2023) families of hierarchical vision models as the region encoders for the classification experiments. All variants of Swin—tiny, small, base, and large—are pretrained on the ImageNet-1k (Russakovsky et al., 2015) dataset. Both Hiera-B and Hiera-B+ are initialized with MAE (He et al., 2022) pre-trained/ImageNet-1k fine-tuned weights. Two

Model	Top-1 Acc	Size(s)	Param	Mem (GB)
Swin-T	53.76	256	31M	0.30
Swin-T ⟨xT⟩ Hyper	52.93	256/256	47M	0.31
Swin-T ⟨xT⟩ Hyper	60.56	512/256	47M	0.29
Swin-T ⟨xT⟩ XL	58.92	512/256	47M	0.17
Swin-T ⟨xT⟩ Mamba	**61.97**	512/256	44M	0.29
Swin-S	58.45	256	52M	0.46
Swin-S ⟨xT⟩ Hyper	57.04	256/256	69M	0.46
Swin-S ⟨xT⟩ Hyper	**63.62**	512/256	69M	0.46
Swin-S ⟨xT⟩ XL	62.68	512/256	69M	0.23
Swin-S ⟨xT⟩ Mamba*	-	-	-	-
Hiera-B	48.60	224	54M	0.26
Hiera-B+	50.47	224	73M	0.33
Hiera-B ⟨xT⟩ Hyper	**57.20**	448/224	70M	0.21
Swin-B	58.57	256	92M	0.50
Swin-B ⟨xT⟩ Hyper	55.52	256/256	107M	0.61
Swin-B ⟨xT⟩ Hyper	**64.08**	512/256	107M	0.74
Swin-B ⟨xT⟩ XL	62.09	512/256	107M	0.39
Swin-B ⟨xT⟩ Mamba	63.73	512/256	103M	0.58
Swin-L	68.78	256	206M	0.84
Swin-L ⟨xT⟩ Hyper	67.84	256/256	215M	1.06
Swin-L ⟨xT⟩ Hyper	72.42	512/256	215M	1.03
Swin-L ⟨xT⟩ XL	**73.47**	512/256	215M	0.53
Swin-L ⟨xT⟩ Mamba	73.36	512/256	212M	1.03

Table 2: **Comparison to prior methods on iNaturalist Reptilia Classification.** Our methods improve on prior works significantly, showing that previous methods still fail to integrate global context. Memory is per base region size, which is fixed for each comparison. *Custom Mamba CUDA kernels are incompatible with Swin-S at this time.

layers of *Hyper* or four layers of Mamba are used as the context encoder, intialized randomly.

We train end-to-end on the Reptilia subset of iNaturalist 2018 for 100 epochs using the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$) using cosine learning rate decay schedule. Swin-T, Swin-S, Hiera-B/+, and their *xT* variants use a base learning rate of 1×10^{-4} while Swin-B, Swin-L, and their *xT* variants use a base learning rate of 1×10^{-5}. *xT*'s nested tokenization scheme is represented as two values, e.g. 512/256, where the first value is the size of the regions extracted from the input image and the second value is the input size expected by the region encoder.

Table 2 contains results for variants of Swin, Hiera, and their *xT* variants on iNaturalist-Reptilia. Particularly, we demonstrate *xT*'s results as a function of region/input size and type of context encoder. *xT* outperforms their comparable baselines by up to 8.6% in top-1 accuracy.

We show in Figure 3 that *xT* sets a new accuracy-parameter frontier when compared against existing methods.

5.2. Segmentation on xView3-SAR

We evaluate *xT* on the xView3-SAR (Paolo et al., 2022) dataset, a large, real-world dataset of satellite imagery for the task of detecting dark vessels. The average image size is

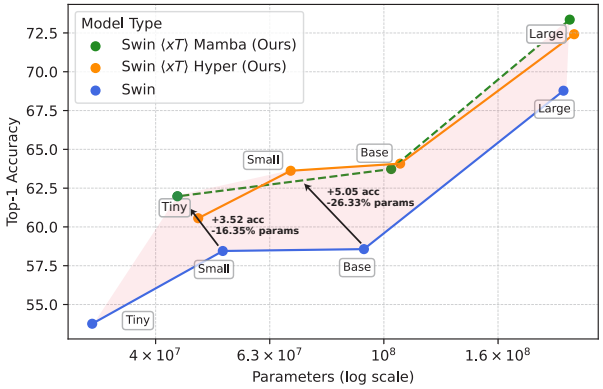

Figure 3: ***xT* improves upon the accuracy-parameter frontier of existing methods for high-resolution classification on iNaturalist** using nested tokenization on top of the origin architecture for larger context. See Table 2 for detailed comparisons.

$29,400 \times 24,400$ pixels large, so accurate detections require using long-range geospatial context to make decisions. We calculate two metrics on xView3-SAR: the aggregate and overall detection F_1 scores which reflect general task proficiency, and of most importance, the close-to-shore detection F_1, which requires detecting vessels close to the shore by using predominant shoreline information.

We adopt the same setup as prior methods (Seferbekov, 2022), tackling the problem as a segmentation and regression problem using a standard encoder-decoder architecture for dense prediction. In our case, we adopt Swin Transformer v2 (Liu et al., 2022) pretrained on ImageNet-1k as our region encoder and use a UNet-style decoder (Ronneberger et al., 2015). We regress the centers of objects as Gaussians and vessels as binary masks.

We test Swin-T, Swin-S, and Swin-B, and their *xT*-variants using Transformer-XL as a context encoder with $N = 4$ layers. The models are trained over encoder input sizes of 512 and 1024, which *xT* effectively boosts to 4096. At inference time, we take overlapping crops of the same size across the image and combine our predictions with post-processing. We sweep our models over learning rates of $\{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$ using AdamW (with the same hyperparameters as in Section 5.1) and report the validation numbers in Table 3.

While prior works tackle xView3 using smaller models like CNNs, ours opts to adapt transformers for the task, which is an under-unexplored space for the type of images present. Therefore, larger models are not always better, as observed in Table 3. However, *xT* always outperforms the corresponding non-context model, beating baselines by up to 13.4 points on the overall F_1 detection score and 6.0 points on the close-to-shore F_1 score, signaling *xT*'s ability to im-

Model	Shore↑	Agg↑	F_1 Score↑	Param	Mem↓	Input Size(s)
Swin-T	50.0	47.6	67.8	32.4	1.24	512
Swin-T	51.6	53.2	76.8	32.4	5.30	1024
Swin-T ⟨xT⟩ XL	47.5	49.4	**81.2**	36.9	**0.47**	4096/512
Swin-T ⟨xT⟩ XL	**56.0**	**54.8**	78.2	36.9	1.65	4096/1024
Swin-S	46.1	44.9	67.7	53.7	1.84	512
Swin-S	41.2	43.8	71.0	53.7	7.24	1024
Swin-S ⟨xT⟩ XL	50.2	48.1	75.3	58.3	**0.54**	4096/512
Swin-S ⟨xT⟩ XL	**52.8**	**55.1**	**78.8**	58.3	2.24	4096/1024
Swin-B	50.2	51.6	72.1	92.7	2.36	512
Swin-B	**54.4**	**54.7**	75.8	92.7	9.65	1024
Swin-B ⟨xT⟩ XL	52.4	51.5	76.4	97.4	**0.70**	4096/512
Swin-B ⟨xT⟩ XL	51.0	50.8	**77.2**	97.4	2.82	4096/1024

Table 3: **Comparison to prior methods on xView3-SAR detection.** We evaluate our methods on xView3-SAR, a dataset for dark vessel detection from satellite imagery. Our method improves on prior state of the art hierarchical transformers significantly while introducing few extra parameters and using less memory per region due to our efficient context encoder. This shows that prior works, expectedly, are unable to model long range visual contexts.

Model	mAP	Input Size(s)
Swin-B-DetINO	OOM	2048
Swin-B-DetINO ⟨xT⟩ XL	**43.0**	2048/512

Table 4: **Comparison with Swin-B on the Cityscapes object detection task.** Swin-B is unable to model Cityscapes images in their entirety within the memory of an 80GB A100.

prove models on large context tasks. Due to our streaming, two-stage architecture with nested tokenization, the memory utilized per region is much lower as we see multiple regions at once per pass while only introducing $5 - 10\%$ more parameters, .

5.3. Object Detection

We use SwinV2 (Liu et al., 2022) as the backbone and adopt DetINO (Zhang et al., 2022) for the detection head. We follow the standard COCO (Lin et al., 2015) 1x training schedule and train on Cityscapes for 12 epochs with the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and a learning rate of 1×10^{-3}. The learning rate for the SwinV2 backbone is scaled by a factor of 0.1. The input is first padded to the nearest multiple of chip size first, and then chipped in the same way as Section 5.1. Each region is then processed by the region encoder independently. The outputs from Swin encoder are concatenated to form a global feature map of the entire image, which is then processed by the context encoder. The resulting sequence is passed to the DetINO detection head.

We report mAP for our detection experiments. These experiments are relatively brief—naïve usage of the Swin backbone results in out-of-memory errors for the large 1,024 x 2,048 images found in Cityscapes. Conversely, *xT* is able to model the images with ease.

6. Ablations

In this section, we ablate our design choices based on iNaturalist classification, particularly around our context encoder. The settings that we keep constant are that *xT* is run on 512/256 inputs, where we chunk our four regions together to create one sequence into our context encoder. As this usually fits in memory, we have no need to perform cross-chunk contextualizing as we do in xView.

6.1. Context Encoder Architecture

We tested ViT, Hyper, and Mamba as context encoders in the course of our classification experiments. These results are detailed Table 5a. iNaturalist 2018 resizes their images to 800×800 pixels for training. These images can be modeled comfortably by ViT and benefit from full self-attention. However, both Hyper and Mamba both perform better than ViT as context encoders with the added benefit of having a much larger capacity for scale.

While Mamba has less parameters than both ViT and Hyper—up to 8% fewer parameters for Swin-T ⟨xT⟩ Mamba than for Swin-T ⟨xT⟩ Hyper—this difference disappears as the region encoder increases in size as seen in Table 2. Furthermore, the decrease in parameters ultimately has an insignificant impact on the peak memory usage of the overall *xT* model.

6.2. Context Encoder Depth

Crucial to our design is how deep our context encoder should be, as our goal is to keep it as lightweight as possible so its overhead is minimal. We find that an acceptable increase in parameters keeps the number of layers to either 1 or 2. As shown in Table 5b, larger region encoders generally benefit from having deeper context encoders. The accuracy is the greatest when the depth is 2 for the largest model, and the trade-off is acceptable for the smallest model, being within 1 accuracy point, so we choose depth 2 as our default.

6.3. Resolution Size Matters

We revisit the central assumption of our work, which is if high resolution is essential for understanding. In Table 5c, we emphasize our results on iNaturalist which prove precisely that models previously did not take advantage of such context, and now do with *xT*. Comparing the Swin-T/L 256 run with Swin-T/L 256/256, which is our method taking in no context, our model actually does worse with extra parameters, likely due to non-functional parameters interfering with the model's learning. However, once we increase our input image size by $4\times$ to $512/256$, our model is immedi-

context	acc	param	mem
ViT	63.26	107M	0.73
Hyper	**64.08**	107M	0.74
Mamba	63.73	103M	0.58

(a) **Context Encoder Backbone.** ViT and Hyper perform similarly without cross chunk context, while Mamba performs most efficiently.

model	depth	acc
Swin-T	1	**86.38**
	2	85.09
Swin-S	1	**88.26**
	2	88.03
Swin-B	1	89.08
	2	**90.73**
Swin-L	1	91.67
	2	**94.48**

(b) **Context Encoder Depth.** Larger region encoders benefit from deeper context encoders. We choose $N = 2$ for our default depth.

model	size	acc	params
Swin-T	256	53.76	31M
	256/256	52.93	47M
	512/256	**60.56**	47M
Swin-L	256	68.78	206M
	256/256	67.84	215M
	512/256	**72.42**	215M

(c) **Resolution Matters.** *xT* does worse than region encoder-only given no context, but performs much better with multiple regions while using the same # of parameters.

Table 5: **Ablating *xT* design choices.** We highlight our defaults in blue and **bold** the best numbers.

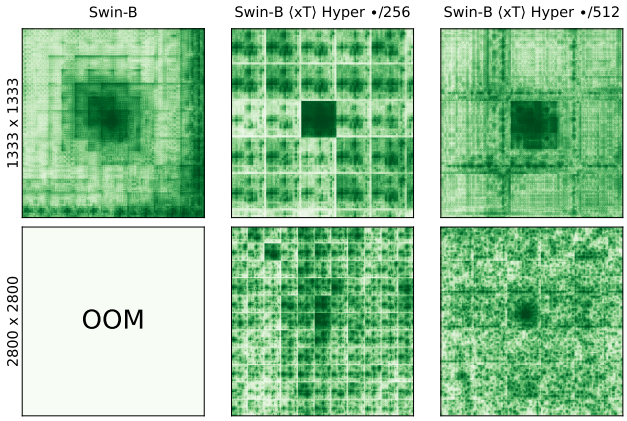

Figure 4: **Effective receptive fields of Swin-B and Swin-B ⟨xT⟩ Hyper**. The center feature from the last layer of region encoder is used to assess sensitivity to areas across the image. Darker green signifies greater sensitivity.

ately able to take advantage of the context with no increase in parameters, boosting accuracy up by 4.6%-7.6%.

7. Discussion

7.1. Effective Receptive Field

We empirically visualize the effective receptive field (ERF) (Luo et al., 2016) of *xT* in Figure 4. The effective receptive field is computed by setting the gradient of one pixel in the final feature map to 1 while keeping the gradient of all other pixels as zero, and then computing the gradient with respect to the input image through a backward pass. The visualized image measures the magnitude of gradient with darker green areas demonstrating greater "sensitivity" and lighter areas demonstrating lower. We show that Swin Transformer has an ERF similar to a skewed Gaussian which vanishes quickly over distance. Comparatively, *xT* retains a more uniform distribution across the entire image. Since our nested tokenization approach is effectively a convolution, we see convolution-like artifacts in the ERF that are mitigated as region sizes get larger. This highlights the *xT* framework's capacity at capturing and integrating long range context in large images.

7.2. Throughput

In lieu of FLOPS, which is difficult to standardize, hardware dependent, and an inconsistent measure of running time when using custom kernels, we instead report the throughput of our method and compare to prior works in Figure 5. For our comparisons, we use 40GB Nvidia A100 GPUs. As we work with multiple image resolutions at once, we calculate a throughput of *regions/second*, which is the number of encoder-sized regions we process per second.

On a per-model size comparison, our method drops in throughput slightly compared to prior methods (with the exception of the fastest size, Tiny). However, we are able to achieve large accuracy gains for this slight tradeoff which are much better than if we used a larger model from prior work instead. This demonstrates the benefit of our method in improving the frontier in modeling larger images efficiently.

7.3. Memory Growth

Modern vision backbone, such as Swin, rapidly run out-of-memory near-quadratically as the image size increases. In Figure 6, we show how the *xT* framework effectively removes this quadratic memory growth, allowing the model to reason over large images end-to-end. Thanks to our chunking design, our method incurs minimal extra memory when we increase our input size as we recurrently store previous regions as context for future context-aware encoding.

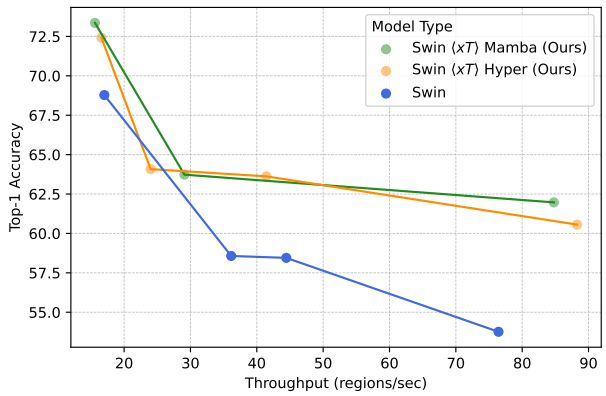

Figure 5: ***xT* offers greatly increased accuracy per throughput.** On iNaturalist classification, we find that our models only slightly diminished throughput (with the exception of Swin-T ⟨xT⟩ XL/Mamba) but achieved greater accuracies at each throughput threshold.

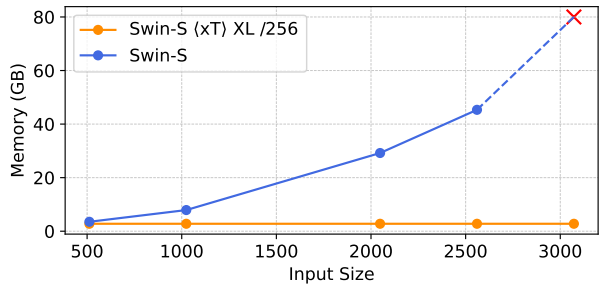

Figure 6: **Swin rapidly goes out of memory (indicated by the red X) as images grow in size whereas *xT* retains near-constant memory cost**. *xT* can scale to much larger images than the naïve usage of a vision backbone.

7.4. Locality of Features

Many vision backbones have adjustable receptive fields for the integration of local information. Larger receptive fields have memory and throughput implications, however. We ablate the Swin transformer for various region and window sizes and compare the impact on accuracy, throughput, and memory in Table 6 on the iNaturalist classification task.

We set the window size used by Swin appropriately as a function of region size. That is, an region size of 256 corresponds to a window size of 16, region size of 512 to a window size of 32, and so on. Swin-B, without the usage of *xT*, can achieve marginally higher accuracy at small image sizes. However, this trend is reversed when working with larger images. Of importance is the ability for *xT* to retain high throughput (measured as time per epoch in minutes) and low memory usage (memory per region in GB) while achieving equivalent accuracy downstream.

Model	Acc.	Size(s)	W. Size	Th.	Mem.
Swin-B	**67.02**	512	32	29.53	6.02
Swin-B ⟨xT⟩ XL	65.49	512/256	16	**12.77**	**0.49**
Swin-B	67.37	1024	64	242.92	25.1
Swin-B ⟨xT⟩ XL	**68.19**	1024/256	16	**64.7**	**2.03**

Table 6: **Comparison of top-1 accuracy (Acc.), time per epoch (Th.), and memory per region (Mem.) as a function of region size(s) and window sizes.** Swin-B and our method have approximately equivalent accuracies, but *xT* achieves much more desirable throughput and memory utilization.

8. Conclusion

Large images have been the norm for the past decade in consumer and professional domains. Yet, state-of-the-art methods in computer vision still limit themselves to modeling small images by throwing away valuable information via down-sampling and cropping. Large images from the real world contain a wealth of information that often require context across long distances to properly make sense of.

In this work, we introduce *xT*, a framework through which vision models trained to handle small images can cheaply integrate larger context over large images. *xT* achieves significant gains in downstream tasks such as classification, detection, and segmentation on real-world datasets. By offering a simple framework by which vision researchers can effectively model large images, we aim to widen the types of data and tasks vision researchers aim to solve.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

Trevor Darrell's group was supported in part by funding from the Department of Defense as well as BAIR's industrial alliance programs. This work was additionally supported by the DoD's High Performance Computing Modernization Program. Ritwik Gupta is supported by the National Science Foundation under Grant No. DGE-2125913 and funding from the UC Berkeley AI Policy Hub. We are grateful to Bryan Comstock for his timely technical support even when the authors were asking a lot of questions. Thank you to Suzie Petryk and William Yang for reviewing the transcript.

References

Bahdanau, D., Cho, K., and Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate, May 2016.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating Long Sequences with Sparse Transformers, April 2019.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.350.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285.

Dalal, N. and Triggs, B. Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, October 2020.

Gadermayr, M., Dombrowski, A.-K., Klinkhammer, B. M., Boor, P., and Merhof, D. CNN cascades for segmenting sparse objects in gigapixel whole slide images. *Computerized Medical Imaging and Graphics*, 71:40–48, January 2019. ISSN 0895-6111. doi: 10.1016/j.compmedimag.2018.11.002.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

Graves, A., Wayne, G., and Danihelka, I. Neural Turing Machines, December 2014.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations (ICLR)*, 2022.

Gu, J., Kwon, H., Wang, D., Ye, W., Li, M., Chen, Y.-H., Lai, L., Chandra, V., and Pan, D. Z. Multi-Scale High-Resolution Vision Transformer for Semantic Segmentation, November 2021.

Han, I., Jayaram, R., Karbasi, A., Mirrokni, V., Woodruff, D. P., and Zandieh, A. Hyperattention: Long-context attention in near-linear time. *arXiv preprint arXiv:2310.05869*, 2023.

He, K., Chen, X., Xie, S., Li, Y., Dollar, P., and Girshick, R. Masked Autoencoders Are Scalable Vision Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-66546-946-3. doi: 10.1109/CVPR52688.2022.01553.

Hochreiter, S. Untersuchungen zu dynamischen neuronalen Netzen. April 1991.

Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, April 1982. doi: 10.1073/pnas.79.8.2554.

Hubel, D. H. and Wiesel, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154, January 1962. ISSN 0022-3751. doi: 10.1113/jphysiol.1962.sp006837.

Koenderink, J. J. and Van Doorn, A. J. The Structure of Locally Orderless Images. *International Journal of Computer Vision*, 31(2):159–168, April 1999. ISSN 1573-1405. doi: 10.1023/A:1008065931878.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. Microsoft COCO: Common Objects in Context, February 2015.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin Transformer: Hierarchical

Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, Montreal, QC, Canada, October 2021. IEEE. ISBN 978-1-66542-812-5. doi: 10.1109/ICCV48922.2021.00986.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., and Guo, B. Swin transformer v2: Scaling up capacity and resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Luo, W., Li, Y., Urtasun, R., and Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

Nguyen, E., Goel, K., Gu, A., Downs, G. W., Shah, P., Dao, T., Baccus, S. A., and Ré, C. S4nd: Modeling images and videos as multidimensional signals using state spaces. *Advances in Neural Information Processing Systems*, 35, 2022.

Paolo, F., Lin, T.-t. T., Gupta, R., Goodman, B., Patel, N., Kuster, D., Kroodsma, D., and Dunnmon, J. xView3-SAR: Detecting Dark Fishing Activity Using Synthetic Aperture Radar Imagery. *Advances in Neural Information Processing Systems*, 35:37604–37616, December 2022.

Reed, C. J., Gupta, R., Li, S., Brockman, S., Funk, C., Clipp, B., Keutzer, K., Candido, S., Uyttendaele, M., and Darrell, T. Scale-MAE: A Scale-Aware Masked Autoencoder for Multiscale Geospatial Representation Learning. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4065–4076, October 2023. doi: 10.1109/ICCV51070.2023.00378.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Rumelhart, D. E. and McClelland, J. L. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, pp. 318–362. MIT Press, 1987. ISBN 978-0-262-29140-8.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.

Ryali, C., Hu, Y.-T., Bolya, D., Wei, C., Fan, H., Huang, P.-Y., Aggarwal, V., Chowdhury, A., Poursaeed, O., Hoffman, J., Malik, J., Li, Y., and Feichtenhofer, C. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML'23*, pp. 29441–29454, Honolulu, Hawaii, USA, July 2023. JMLR.org.

Seferbekov, S. xView3 2nd place solution. https://github.com/DIUx-xView/xView3_second_place, 2022.

Sukhbaatar, S., Grave, E., Bojanowski, P., and Joulin, A. Adaptive Attention Span in Transformers. In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 331–335, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1032.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. LLaMA: Open and Efficient Foundation Language Models, February 2023.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The iNaturalist Species Classification and Detection Dataset. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8769–8778, Salt Lake City, UT, June 2018. IEEE. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00914.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions, August 2021.

Weston, J., Chopra, S., and Bordes, A. Memory Networks, November 2015.

Wu, C.-Y., Li, Y., Mangalam, K., Fan, H., Xiong, B., Malik, J., and Feichtenhofer, C. MeMViT: Memory-Augmented Multiscale Vision Transformer for Efficient Long-Term Video Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13587–13597, 2022.

Yang, C., Xu, J., Mello, S. D., Crowley, E. J., and Wang, X. GPViT: A High Resolution Non-Hierarchical Vision Transformer with Group Propagation. In *The Eleventh International Conference on Learning Representations*, September 2022.

Yu, F. and Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions, April 2016.

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L. M., and Shum, H.-Y. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022.

Zhang, P., Dai, X., Yang, J., Xiao, B., Yuan, L., Zhang, L., and Gao, J. Multi-Scale Vision Longformer: A New Vision Transformer for High-Resolution Image Encoding, May 2021.