Hiera:A Hierarchial Vision Transformer without the Bells-and-Whistles

Presented By - Rittik Panda(MT2022090)

Introduction:

- In 2021, after the introduction of Vision Transformers, they have dominated several tasks in computer vision for their self attention based simple architecture, their accuracy in large-scale datasets,and ability to scale.Moreover, their simplicity unlocks the use of powerful pretraining strategies such as MAE which make ViTs computationally and data efficient to train.

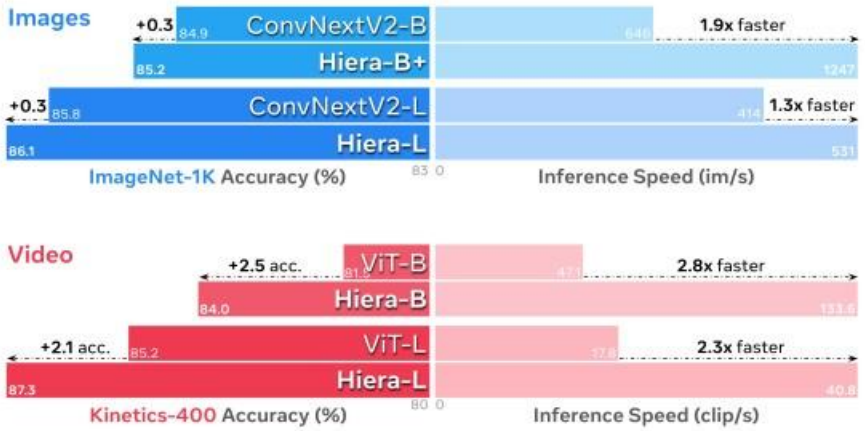
- **Yet many practitioners use convolution-based models for projects. Why?**

- One of the main reasons is that **CNNs can capture local spatial information effectively through convolutional layers and hierarchically increases the receptive field in later layers by max pooling (similar to human visual cortex).**On the other hand ViTs use their parameters inefficiently as they use the same spatial resolution and the same number of channels (Global Attention) in the network i.e. **non availability of strong inductive bias.**
- For Solving this issue many hiearchical VITs have been proposed over the years like-**SWIN Transformer, MVIT, MVIT-V2** etc, which use fewer channels but higher spatial resolution in early stages with simple features and more channels but lower spatial resolution later in the model with more complex features.
- These modern hierarchical vision transformers have added **several vision-specific components in the pursuit of supervised classification performance**.While these components lead to effective accuracies and attractive FLOP counts, the added complexity actually makes these transformers slower than their vanilla ViT counterparts.

- **Here, Hiera comes into the picture.**

Key Contributions:

- The authors have shown that this added complexity (bells-and-whistles) is unnecessary and can be eliminated from these hierarchical vision transformers through pretraining with a strong visual pretext task (MAE).
- The resulting model, Hiera, retains the accuracy of state-of-the-art multi-stage vision transformers but is significantly faster both during training and inference on variety of tasks for image and video recognition.



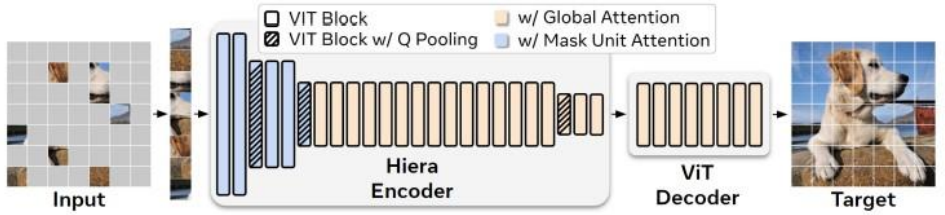


Figure 2. **Hiera Setup.** Modern hierarchical transformers like Swin (Liu et al., 2021) or MViT (Li et al., 2022c) are more parameter efficient than vanilla ViTs (Dosovitskiy et al., 2021), but end up slower due to overhead from adding spatial bias through vision-specific modules like shifted windows or convs. In contrast, we design Hiera to be as simple as possible. To add spatial bias, we opt to *teach* it to the model using a strong pretext task like MAE (pictured here) instead. Hiera consists entirely of standard ViT blocks. For efficiency, we use local attention within "mask units" (Fig. 4, 5) for the first two stages and global attention for the rest. At each stage transition, Q and the skip connection have their features doubled by a linear layer and spatial dimension pooled by a 2×2 maxpool. Hiera-B is shown here (see Tab. 2 for other configs).

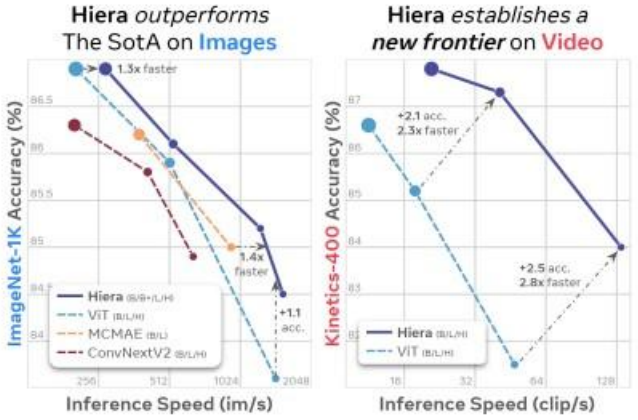


Figure 3. **Performance vs. prior work.** Hiera compared to B, L, and H variants of SotA models that use MAE-like pretraining. On images, Hiera is faster and more accurate than even the most recent SotA (He et al., 2022; Gao et al., 2022; Woo et al., 2023), offering 30-40% speed-up compared to the best model at every scale. On video, Hiera represents a new class of performance, significantly improving accuracy, while *being over 2× faster* than popular ViT models. Marker size is proportional to FLOP count.

Prior Work:

- Over the years many hierarchical vision transformers have been proposed.
- For this presentation, I mainly focused on the following papers.
    1. MVIT
    2. MVIT-V2

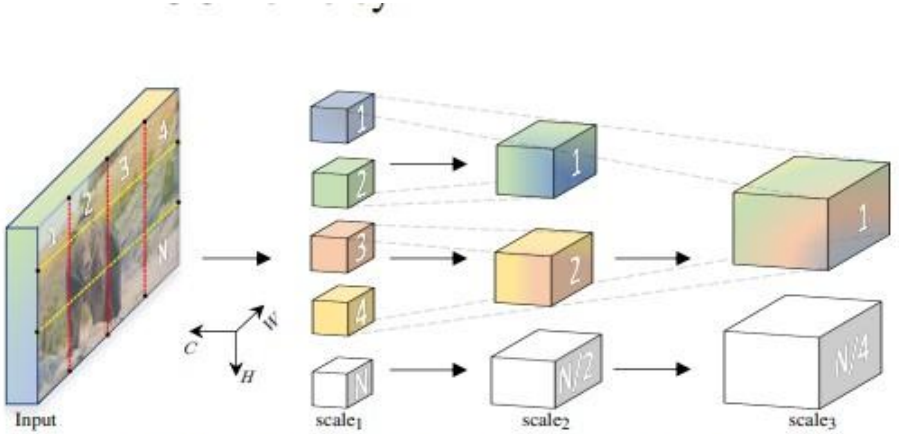


Figure 1. **Multiscale Vision Transformers** learn a hierarchy from *dense* (in space) and *simple* (in channels) to *coarse* and *complex* features. Several resolution-channel *scale* stages progressively *increase* the channel capacity of the intermediate latent sequence while *reducing* its length and thereby spatial resolution.

Multiscale Vision Transformers (MVIT)

- Multiscale Transformers have several channel-resolution‘scale’ stages. Starting from the high image resolution and a small channel dimension(focusing on simple low-level visual information), the stages hierarchically expand the channel capacity while reducing the spatial resolution(focusing on complex high-level features).
- They introduced MHPA instead of standard MHA.(which maintain a constant channel capacity and resolution throughout the network).
- MHPA changes input tensor of dimensions s given by, L = T × H × W to L˜ given $\tilde{L} = \left\lfloor \frac{L + 2p - k}{s} \right\rfloor + 1$ with the equation applying coordinate-wise.

where the reduced sequence length L˜ = T˜ × H˜ × W˜ .

- By default they used overlapping kernels k with shapepreserving padding p in pooling attention operators, so that L˜ experiences an overall reduction by a factor of sT sH sW .
- Attention is now computed on these shortened vectors.

$$\mathrm{PA}(\cdot) = \mathrm{Softmax}(\mathcal{P}(Q;\Theta_Q)\mathcal{P}(K;\Theta_K)^T/\sqrt{d})\mathcal{P}(V;\Theta_V),$$

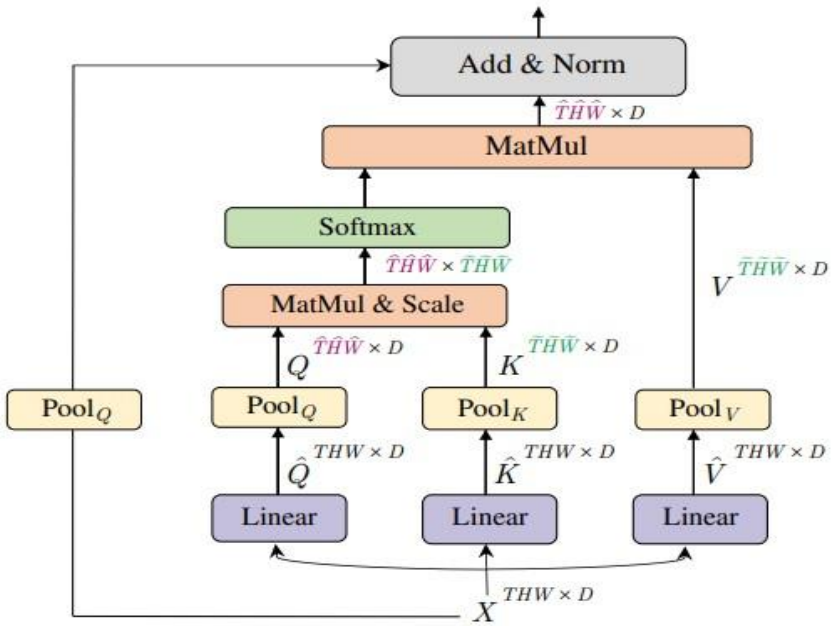


Figure 3. **Pooling Attention** is a flexible attention mechanism that (i) allows obtaining the reduced space-time resolution ($\hat{T}\hat{H}\hat{W}$) of the input (THW) by pooling the query, $Q = \mathcal{P}(\hat{Q};\Theta_Q)$, and/or (ii) computes attention on a reduced length ($\tilde{T}\tilde{H}\tilde{W}$) by pooling the key, $K = \mathcal{P}(\hat{K};\Theta_K)$, and value, $V = \mathcal{P}(\hat{V};\Theta_V)$, sequences.

Key Points Of MVIT:

- A scale stage is defined as a set of N transformer blocks that operate on the same scale with identical resolution across channels and space-time dimensionsD×T×H×W.
- At a stage transition,the channel dimension is upsampled(by increasing the output of the final MLP layer in the previous stageby a factor that is relative to the resolution change introduced at the stage) while the length of the sequence is down-sampled.
- In Query Pooling, Since, our intention is to decrease resolution at the beginning of a stage and then preserve this resolution throughout the stage, only the first pooling attention operatorof each stage operates at non-degenerate query stride s ,Q > 1, with all other operators constrained to sQ ≡ (1, 1, 1).
- Unlike Query pooling, changing the sequence length of key K and value V tensors, does not change the output sequence length and, hence, the space-time resolution. However, they play a key role in overall computational requirements of the pooling attention operator.
- We decouple the usage of K, V and Q pooling, with Q pooling being used in the first layer of each stage and K, V pooling being employed in all other layers.
- The pooling stride used on K and value V tensors is identical within a stage but varies adaptively w.r.t. to the scale across the stages.
- Since the channel dimension and sequence length change inside a residual block, we pool the skip connection to adapt to the dimension mismatch between its two ends.

stages	operators	output sizes
data layer	stride $\tau \times 1 \times 1$	$D \times T \times H \times W$
cube_1	$c_T \times c_H \times c_W$, D stride $s_T \times 4 \times 4$	$D \times \frac{T}{s_T} \times \frac{H}{4} \times \frac{W}{4}$
scale_2	$\begin{bmatrix} \text{MHPA}(D) \\ \text{MLP}(4D) \end{bmatrix} \times N_2$	$D \times \frac{T}{s_T} \times \frac{H}{4} \times \frac{W}{4}$
scale_3	$\begin{bmatrix} \text{MHPA}(2D) \\ \text{MLP}(8D) \end{bmatrix} \times N_3$	$2D \times \frac{T}{s_T} \times \frac{H}{8} \times \frac{W}{8}$
scale_4	$\begin{bmatrix} \text{MHPA}(4D) \\ \text{MLP}(16D) \end{bmatrix} \times N_4$	$4D \times \frac{T}{s_T} \times \frac{H}{16} \times \frac{W}{16}$
scale_5	$\begin{bmatrix} \text{MHPA}(8D) \\ \text{MLP}(32D) \end{bmatrix} \times N_5$	$8D \times \frac{T}{s_T} \times \frac{H}{32} \times \frac{W}{32}$

Table 2. **Multiscale Vision Transformers (MViT)** base model. Layer cube_1, projects *dense* space-time cubes (of shape $c_t \times c_y \times c_w$) to D channels to reduce spatio-temporal resolution to $\frac{T}{s_T} \times \frac{H}{4} \times \frac{W}{4}$. The subsequent stages progressively down-sample this resolution (at beginning of a stage) with MHPA while simultaneously increasing the channel dimension, in MLP layers, (at the end of a stage). Each stage consists of N_* transformer blocks, denoted in [brackets].

MVIT-V2

- Here they introduced an improved version of MHPA.
- They incorporated mainly two things in the new version.

1.Decomposed Relative position embedding.

- In MViT, the interaction between two patches will change depending on their absolute position in images even if their relative positions stay unchanged (ignoring the fundamental principle of shift-invariance)
- To address this issue, we incorporate relative positional embeddings, which only depend on the relative location distance between tokens into the pooled self-attention computation.

$$\mathrm{Attn}(Q,K,V) = \mathrm{Softmax}\left((QK^{\top} + E^{(\mathrm{rel})})/\sqrt{d}\right)V,$$

$$\text{where} \quad E_{ij}^{(\mathrm{rel})} = Q_i \cdot R_{p(i),p(j)}. \qquad (3)$$

where p(i) and p(j) denote the spatial (or spatiotemporal) position of element i and j.

- However, the number of possible embeddings Rp(i),p(j) scale in O(TW H), which can be expensive to compute. To reduce complexity, we decompose the distance computation between element i and j along the spatiotemporal axes: $R_{p(i),p(j)} = R^{\mathrm{h}}_{h(i),h(j)} + R^{\mathrm{w}}_{w(i),w(j)} + R^{\mathrm{t}}_{t(i),t(j)},$
- This reduces the number of learned embeddings to O(T + W + H). Which has a large effect on early stage high resolution feature maps.

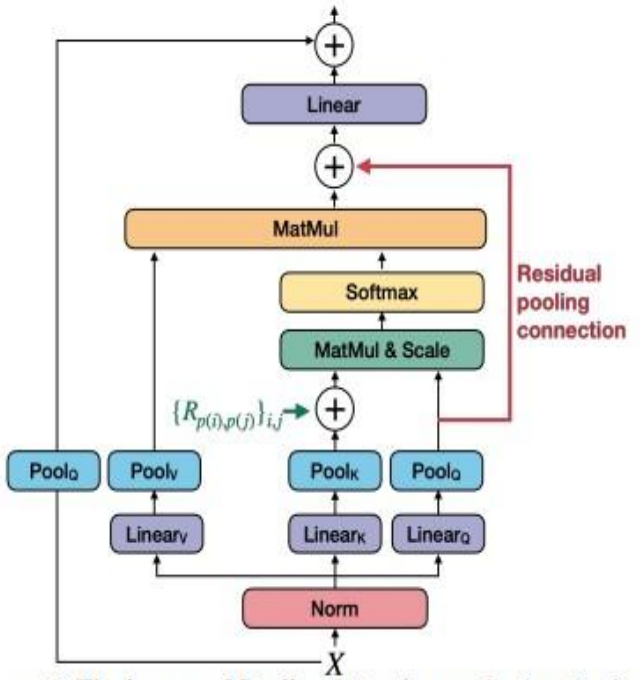


Figure 2. **The improved Pooling Attention** mechanism that incorporating decomposed relative position embedding, $R_{p(i),p(j)}$, and residual pooling connection modules in the attention block.

MVIT-V2

2.Residual pooling connection

- We saw that,MViTv1 has larger strides on K and V tensors than the stride of the Q tensors which is only downsampled if the resolution of the output sequence changes across stages. This motivates us to add the residual pooling connection with the (pooled) Q tensor to increase information flow and facilitate the training of pooling attention blocks.
- We introduce a new residual pooling connection insidethe attention blocks .Specifically, we add the pooled query tensor to the output sequence Z. So Eq. is reformulated as: $Z := \text{Attn}(Q, K, V) + Q.$
- **MVIT as Object Detector:**

They propose a simple Hybrid window attention (Hwin) design to add cross-window connections. Hwin computes local attention within a window in all but the last blocks of the last three stages that feed into FPN, so the input feature maps to FPN contain global information.

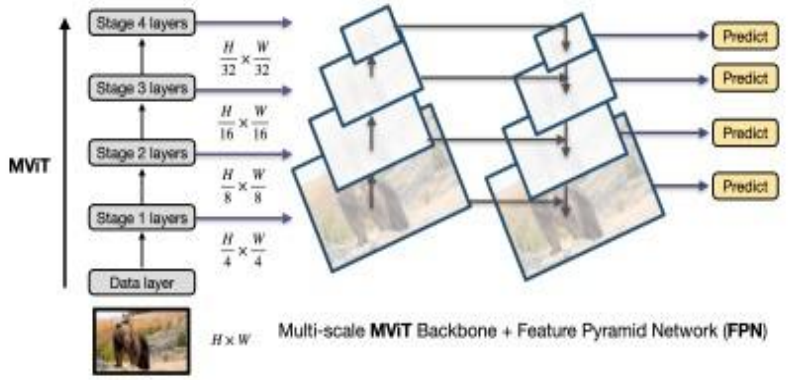


Figure 3. **MViT backbone used with FPN for object detection.** The multiscale transformer features naturally integrate with standard feature pyramid networks (FPN).

Other Minor Changes:
They conduct the channel dimension expansion in the attention computation of the first transformer block of each stage,instead of performing it in the last MLP block of the prior stage as in MViTv1.

Limitation: Large & Complex Model, slow in training and inference.

HIERA Approach:

- The authors aim to design a simplified, efficient multiscale vision transformer that achieves high accuracy on vision tasks without the need for specialized modules like convolution, shifted windows, or attention bias.
- They suggest that the spatial biases provided by these modules, which are absent in vanilla transformers, can be learned through a strong pretext task instead of complicated architectural modifications.
- For this purpose, they use Masked Autoencoding (MAE) as their pretext task (on MVIT-V2 after removing its complex modules) which has proven effective in teaching ViTs localization abilities for downstream tasks.
- MAE pretraining involves deleting masked tokens, unlike other masked image modeling approaches (which usually overwrite them). This method, while efficient, presents a challenge for hierarchical models since it disrupts the 2D grid they depend on.
- To overcome this, the authors introduce a distinction between tokens and “mask units”, where mask units represent the resolution of MAE masking and tokens are the internal resolution of the model. They mask 32×32 pixel regions, which equates to 8 × 8 tokens at the network’s start.
- Then authors evaluated hierarchical models by treating

  mask units as contiguous and separate from other tokens(considering each

  mask unit as a separate image)This allows them to successfully use MAE

  with an existing hierarchical vision transformer.

Creation Of HIERA

- The authors select MViTv2 as the base architecture for their experiments, citing its small 3x3 kernels are least affected by their unique masking technique.
- MVIT-V2 learns multi-scale representations across four stages.In the first two stages, K and V are pooled to decrease computation, and Q is pooled to transition from one stage to the next by reducing spatial resolution.By default, pooling attention in MViTv2 contain convs with stride 1 even if no downsampling is required.
- **MAE Pre-training :**

1.When applying Masked Autoencoders (MAE), the authors note that MViTv2 downsamples by 2x2 three times and uses a token size of 4x4 pixels, hence, they use a mask unit size of 32x32. This ensures each mask unit corresponds to 64,16,4,1 numbers of tokens across different stages,allowing each mask unit to cover at least one distinct token in each stage.

2.The mask units are then manipulated to ensure convolution kernels do not interfere with deleted tokens, treating each mask unit as an individual “image,” while ensuring that self-attention remains global.

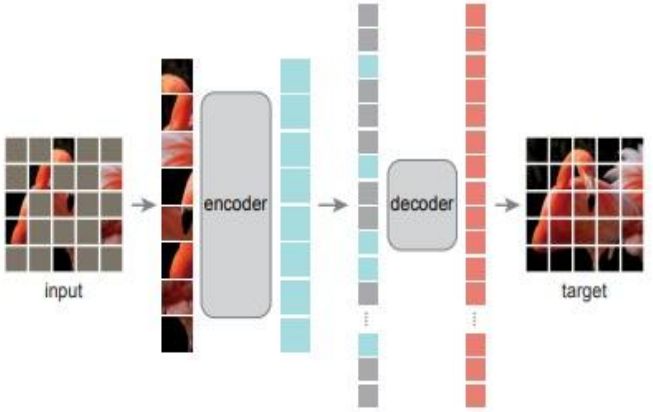


Figure 1. **Our MAE architecture**. During pre-training, a large random subset of image patches (*e.g.*, 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images (full sets of patches) for recognition tasks.

From ‘Masked Autoencoders Are Scalable Vision Learners’ paper

Simplifying MVIT-V2

- The authors experiment with removing non-essential components of MViTv2 while training with Masked Autoencoders (MAE). Their findings show that these components can be removed or simplified while maintaining high image classification accuracy on ImageNet-1K.
- Firstly, they replaced the relative position embeddings in MViTv2 with absolute position embeddings. This simplification resulted in no significant loss of accuracy when training with MAE and was much faster.
- Next, they replaced every convolution layer, a vision-specific module, with maxpools. Although this replacement initially dropped the accuracy, deleting additional stride=1 maxpools returned the model close to its previous accuracy while making it much faster.
- They also removed the overlap in the remaining maxpool layers by setting the kernel size equal to stride for each maxpool. This action eliminated the need for a padding trick, resulting in a model that was faster and more accurate.
- MViTv2 adds a residual connection in the attention layer between Q and the output to assist in learning its pooling attention. They've removed it as they've minimized the number of layers, making attention easier to learn.

	Image		Video	
Setting	acc.	im/s	acc.	clip/s
MViTv2-L Supervised	85.3	219.8	80.5	20.5
Hiera-L MAE				
a. replace rel pos with absolute *	85.6	253.3	85.3	20.7
b. replace convs with maxpools *	84.4	99.9†	84.1	10.4†
c. delete stride=1 maxpools *	85.4	309.2	84.3	26.2
d. set kernel size equal to stride	**85.7**	369.8	**85.5**	29.4
e. delete q attention residuals	85.6	374.3	**85.5**	29.8
f. replace kv pooling with MU attn	85.6	**531.4**	**85.5**	**40.8**

Mask Unit Attention

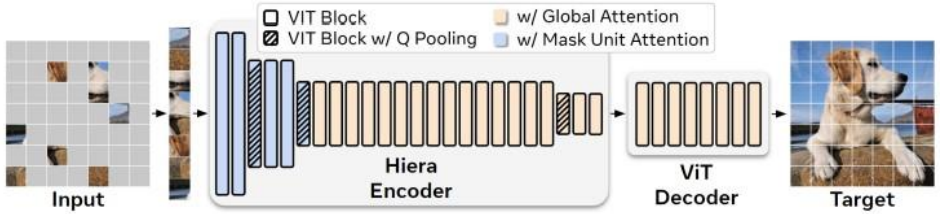
- Pooling Q is necessary to maintain a hierarchical model, but KV pooling is only there to reduce the size of the attention matrix in the first two stages. Directly deleting those would increase computational cost of the network.
- So they introduced an implementationally trivial alternative: local attention within a mask unit called “Mask Unit attention,” replacing KV pooling in the first two stages.
- While this “Mask Unit attention” is local instead of global like pooling attention K and V were only pooled in the first two stages,where global attention isn’t as useful. So this change resulted in no accuracy loss and increased throughput significantly — up to 32% on video.
- Finally, the authors clarify that mask unit attention is distinct from window attention because it adapts the window size to the size of mask units at the current resolution, avoiding potential issues with deleted tokens after downsampling.

Figure 6. **Mask Unit Attn vs. Window Attn.** Window attention (a) performs local attention within a *fixed* size window. Doing so would potentially overlap with deleted tokens during sparse MAE pretraining. In contrast, Mask Unit attention (b) performs local attention within individual mask units, no matter their size.

HIERA

- The culmination of the described changes is a model named “Hiera”. Hiera is 2.4 times faster on images and 5.1 times faster on video than the original MViTv2. In addition Hiera actually boasts improved accuracy due to the implementation of MAE.
- Hiera-L is three times faster to train than a supervised MViTv2-L in image related tasks.Comparing with a modified version of MViTv2 used in video tasks, Hiera-L achieves 85.5% accuracy in 800 pretrain epochs, making it 2.1 times faster to train.
- All benchmarks in the study are performed on an A100 GPU with fp16 precision, as this setup is deemed the most practical.

Figure 7. **Training time.** Measured in half precision A100 days. Our Hiera is *significantly* faster to train than MViTv2 due to being more efficient and benefiting from sparse pretraining (as opposed to MaskFeat). Here, supervised uses 300 epochs for ImageNet-1K and 200 for Kinetics-400, while MaskFeat and MAE use 400 for pretraining on images and 800 on video followed by 50 epochs of finetuning for both. Note that Hiera-L at 200 epochs of pretraining (81.8) already outperforms MViTv2-L supervised (80.5) on video, making it 5.6× faster to obtain higher accuracy.



HIERA Variants:

model	#Channels	#Blocks	#Heads	FLOPs	Param
Hiera-T	[96-192-384-768]	[1-2-7-2]	[1-2-4-8]	5G	28M
Hiera-S	[96-192-384-768]	[1-2-11-2]	[1-2-4-8]	6G	35M
Hiera-B	[96-192-384-768]	[2-3-16-3]	[1-2-4-8]	9G	52M
Hiera-B+	[112-224-448-896]	[2-3-16-3]	[2-4-8-16]	13G	70M
Hiera-L	[144-288-576-1152]	[2-6-36-4]	[2-4-8-16]	40G	214M
Hiera-H	[256-512-1024-2048]	[2-6-36-4]	[4-8-16-32]	125G	673M

Table 2. **Configuration for Hiera variants**. #Channels, #Blocks and #Heads specify the channel width, number of Hierablocks and heads in each block for the four stages, respectively. FLOPs are measured for image classification with 224×224 input. The stage resolutions are $[56^2, 28^2, 14^2, 7^2]$. We introduce B+ for more direct comparison against prior work with slower B models.

MAE Ablations

multi-scale	image	video
✗	85.0	83.8
✔	**85.6**	**85.5**

(a) **Multi-Scale Decoder.** Hiera being *hierarchical*, using multi-scale information for decoding brings significant gains.

mask	image	mask	video
0.5	85.5	0.75	84.9
0.6	**85.6**	0.9	**85.5**
0.7	85.3	0.95	84.4

(b) **Mask ratio.** High masking ratios lead to good performance, with video benefiting from higher masking than image modality.

target	image	video
pixel	85.6	85.5
HOG	**85.7**	**86.1**

(c) **Reconstruction target.** Both pixel and HOG targets result in strong performance.

dpr	image	video
0.0	85.2	84.5
0.1	85.6	85.4
0.2	**85.6**	**85.5**
0.3	85.5	85.2

(d) **Drop path rate.** Surprisingly, we find drop path important during MAE pretraining, especially for video, unlike in He et al. (2022); Feichtenhofer et al. (2022).

depth	image	video
4	85.5	84.8
8	**85.6**	**85.5**
12	85.5	85.4

(e) **Decoder depth.** We find that a deeper decoder than in Feichtenhofer et al. (2022) works better for video.

epochs	image	video
400	85.6	84.0
800	85.8	85.5
1600	**86.1**	86.4
3200	**86.1**	**87.3**

(f) **Pretraining schedule.** Our pretraining follows the same trend as He et al. (2022), benefiting significantly from longer training.

Table 3. **Ablating MAE pretraining with Hiera-L.** For each ablation, we use 400 (800) epochs of sparse MAE pretraining for IN1K (K400) and 50 epochs of dense finetuning unless otherwise noted. Our default† settings are marked in gray. For design choices not ablated here, we find the defaults in (He et al., 2022; Feichtenhofer et al., 2022) to be appropriate. † default pretraining length for the rest of the paper is 1600 (3200) epochs, unless otherwise noted.

Video Results

backbone	pretrain	acc.	FLOPs (G)	Param
ViT-B	MAE	81.5	180×3×5	87M
Hiera-B	MAE	84.0	**102**×3×5	**51M**
Hiera-B+	MAE	**85.0**	133×3×5	69M
MViTv2-L	-	80.5	**377**×1×10	218M
MViTv2-L	MaskFeat	84.3	**377**×1×10	218M
ViT-L	MAE	85.2	597×3×5	305M
Hiera-L	MAE	**87.3**	413×3×5	**213M**
ViT-H	MAE	86.6	1192×3×5	**633M**
Hiera-H	MAE	**87.8**	**1159**×3×5	672M

Table 4. **K400 results**. Hiera improves on previous SotA by a large amount, while being lighter and faster. FLOPs are reported as inference FLOPs × spatial crops × temporal clips.

backbone	pretrain	acc.	FLOPs (G)	Param
MViTv2-L	Sup, IN-21K	85.8	377×1×10	218M
MViTv2-L	MaskFeat	86.4	**377**×1×10	218M
Hiera-L	MAE	**88.3**	413×3×5	**213M**
Hiera-H	MAE	**88.8**	1159×3×5	672M

(a) **Kinetics-600** video classification

backbone	pretrain	acc.	FLOPs (G)	Param
MViTv2-L	Sup, IN-21K	76.7	377×1×10	218M
MViTv2-L	MaskFeat	77.5	**377**×1×10	218M
Hiera-L	MAE	**80.3**	413×3×5	**213M**
Hiera-H	MAE	**81.1**	1159×3×5	672M

(b) **Kinetics-700** video classification

Table 5. **K600 and K700 results**. Hiera improves over SotA by a large margin. FLOPs reported as inference FLOPs × spatial crops × temporal clips. Approaches using extra data are de-emphasized.

backbone	pretrain	acc.	FLOPs (G)	Param
K400 pretrain				
ViT-L	supervised	55.7	598×3×1	304M
MViTv2-$L_{40,312}$	MaskFeat	74.4	2828×3×1	218M
ViT-L	MAE	74.0	597×3×2	305M
Hiera-L	MAE	74.7	**413**×3×1	**213M**
Hiera-L	MAE	**75.0**	**413**×3×2	**213M**
SSv2 pretrain				
ViT-L	MAE	74.3	597×3×2	305M
Hiera-L	MAE	74.9	**413**×3×1	**213M**
Hiera-L	MAE	**75.1**	**413**×3×2	**213M**
ViT-L_{32}	MAE	75.4	1436×3×1	305M
Hiera-L_{32}	MAE	**76.5**	**1029**×3×1	**213M**

Table 6. **SSv2 results** pretrained on Kinetics-400 and SSv2. Hiera improves over SotA by a large margin. We report inference FLOPs × spatial crops × temporal clips.

Image Results

backbone	pretrain	mAP	FLOPs (G)	Param
K400 pretrain				
ViT-L	supervised	22.2	598	304M
MViTv2-$L_{40,312}$	MaskFeat	38.5	2828	218M
ViT-L	MAE	37.0	597	305M
Hiera-L	MAE	**39.8**	**413**	**213M**
ViT-H	MAE	39.5	1192	**633M**
Hiera-H	MAE	**42.5**	**1158**	672M
K600 pretrain				
ViT-L	MAE	38.4	598	304M
MViTv2-$L_{40,312}$	MaskFeat	39.8	2828	218M
Hiera-L	MAE	**40.7**	**413**	**213M**
ViT-H	MAE	40.3	1193	**632M**
Hiera-H	MAE	**42.8**	**1158**	672M
K700 pretrain				
ViT-L	MAE	39.5	598	304M
Hiera-L	MAE	**41.7**	**413**	**213M**
ViT-H	MAE	40.1	1193	**632M**
Hiera-H	MAE	**43.3**	**1158**	672M

Table 7. **AVA v2.2 results** pretrained on Kinetics. Hiera improves over SotA by a large margin. All inference FLOPs reported with a center crop strategy following Fan et al. (2021).

backbone	pretrain	acc.	FLOPs (G)	Param
Swin-T		81.3	**5**	29M
MViTv2-T		82.3	**5**	**24M**
Hiera-T	MAE	**82.8**	**5**	28M
Swin-S		83.0	9	50M
MViTv2-S		83.6	7	**35M**
Hiera-S	MAE	**83.8**	**6**	**35M**
ViT-B		82.3	18	87M
Swin-B		83.3	15	88M
MViTv2-B		84.4	10	**52M**
ViT-B	BEiT, DALLE	83.2	18	87M
ViT-B	MAE	83.6	18	87M
ViT-B	MaskFeat	84.0	18	87M
Swin-B	SimMIM	83.8	15	88M
MCMAE-B	MCMAE	85.0	28	88M
Hiera-B	MAE	84.5	**9**	**52M**
Hiera-B+	MAE	**85.2**	13	70M
ViT-L		82.6	62	304M
MViTv2-L		85.3	42	218M
ViT-L	BEiT, DALLE	85.2	62	304M
ViT-L	MAE	85.9	62	304M
ViT-L	MaskFeat	85.7	62	304M
Swin-L	SimMIM	85.4	**36**	**197M**
MCMAE-L	MCMAE	**86.2**	94	323M
Hiera-L	MAE	86.1	40	214M
ViT-H		83.1	167	**632M**
ViT-H	MAE	**86.9**	167	**632M**
Hiera-H	MAE	**86.9**	**125**	673M

Table 8. **ImageNet-1K** comparison to previous MIM approaches. We de-emphasize approaches using extra data and indicate the source of extra data.

backbone	iNat17	iNat18	iNat19	Places365
ViT-B	70.5	75.4	80.5	57.9
Hiera-B	73.3	77.9	83.0	58.9
Hiera-B+	**74.7**	**79.9**	**83.1**	**59.2**
ViT-L	75.7	80.1	83.4	59.4
Hiera-L	**76.8**	**80.9**	**84.3**	**59.6**
ViT-H	79.3	83.0	**85.7**	59.8
Hiera-H	**79.6**	**83.5**	**85.7**	**60.0**
ViT-H_{448}	83.4	86.8	88.3	60.3
Hiera-H_{448}	**83.8**	**87.3**	**88.5**	**60.6**

Table 9. **Transfer learning** on iNaturalists and Places datasets.

backbone	pretrain	AP^{box}	AP^{mask}	FLOPs	params	time
Swin-B	Sup, 21K	51.4	45.4	0.7T	109M	164ms
MViTv2-B	Sup, 21K	53.1	47.4	0.6T	73M	208ms
Swin-B	Sup	50.1	44.5	0.7T	109M	**164ms**
MViTv2-B	Sup	52.4	46.7	**0.6T**	**73M**	208ms
ViTDet-B	MAE	51.6	45.9	0.8T	111M	201ms
Hiera-B	MAE	52.2	46.3	**0.6T**	**73M**	173ms
Hiera-B+	MAE	**53.5**	**47.3**	**0.6T**	92M	192ms
Swin-L	Sup, 21K	52.4	46.2	1.1T	218M	243ms
MViTv2-L	Sup, 21K	53.6	47.5	1.3T	239M	447ms
MViTv2-L	Sup	53.2	47.1	1.3T	239M	447ms
ViTDet-L	MAE	**55.6**	**49.2**	1.9T	331M	396ms
Hiera-L	MAE	55.0	48.6	**1.2T**	**236M**	**340ms**

Table 10. **COCO object detection and segmentation** using Mask-RCNN. All methods are following the training recipe from Li et al. (2022b) and pretrained on ImageNet-1K by default. Methods using ImageNet-21K pretraining are de-emphasized. Test time is measured on a single V100 GPU with full precision.

Conclusion

- The authors create a simple hierarchical vision transformer by taking an existing one and removing all its bells and-whistles while supplying the model with spatial bias through MAE pretraining.
- The resulting architecture, Hiera is more effective than current work on image recognition tasks and surpasses the state-of-the-art on video tasks.
- However, the authors do not claim that these modules are unnecessary in general. In fact, here intend to show the opposite:the reason these spatial biases were necessary in the first place is because they are required when training a vision transformer from scratch with classification.
- As expected, we see the opposite trend as we did when training with a strong pretext task: the bells-and-whistles are necessary when training in a classical supervised setting.This reiterates the fact that, by training with MAE, we are replacing the need to explicitly build spatial biases into the network's architecture itself.

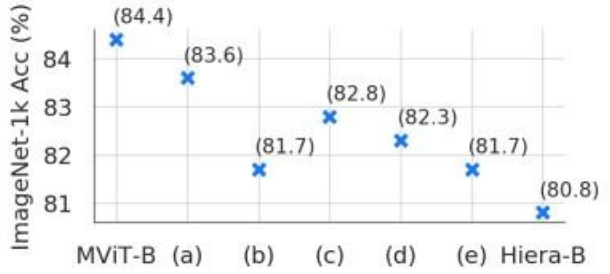


Figure 8. **Training on classification *from scratch*.** Here we repeat the experiment in Tab. 1 but without MAE pretraining, using MViTv2's supervised recipe instead. As expected, the bells-and-whistles that Hiera removes are actually *necessary* when training from scratch—hence their introduction in prior work in the first place. Hiera *learns* spatial biases instead.

Reference Papers

- AN IMAGE IS WORTH 16X16 WORDS:TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE
- Swin Transformer: Hierarchical Vision Transformer using Shifted Windows
- Masked Autoencoders Are Scalable Vision Learners
- Multiscale Vision Transformers
- MViTv2: Improved Multiscale Vision Transformers for Classification and Detection

THANK YOU