Auto-scaling Vision Transformers (ViTs) without Training

Wuyang Chen[1], Wei Huang[2], Xianzhi Du[3], Xiaodan Song[3], Zhangyang Wang[1], Denny Zhou[3]

[1]University of Texas, Austin

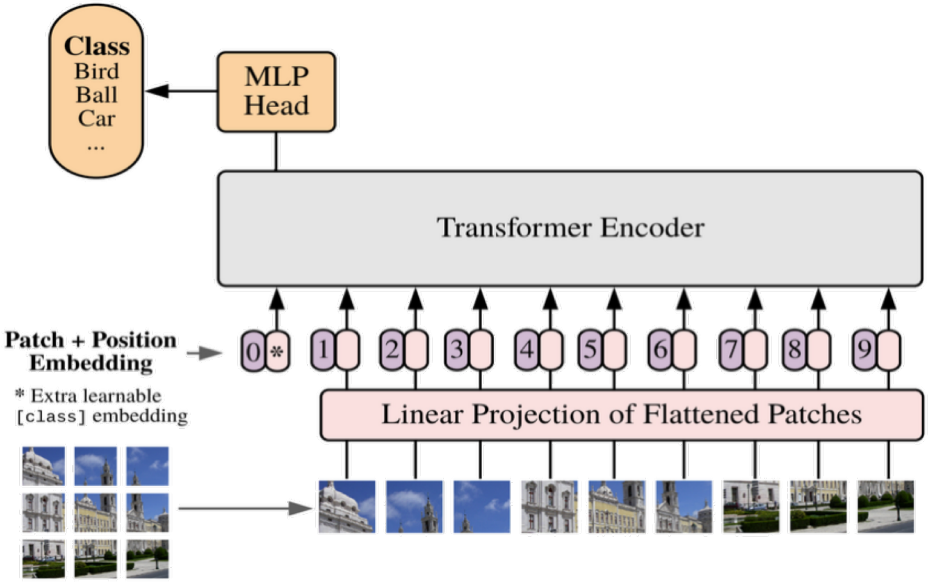
[2]University of Technology Sydney

[3]Google

https://github.com/VITA-Group/AsViT

How to Design & Scale-up ViTs in Principle?

- ViT (Vision Transformer): Tokens + Attentions + FFNs
- Principled designs?
  - Token size / FFN expansion ratio / #heads...
- Principled scaling rules?
  - More widths or more depths?
- Efficiency: find principles w/o heavy cost?



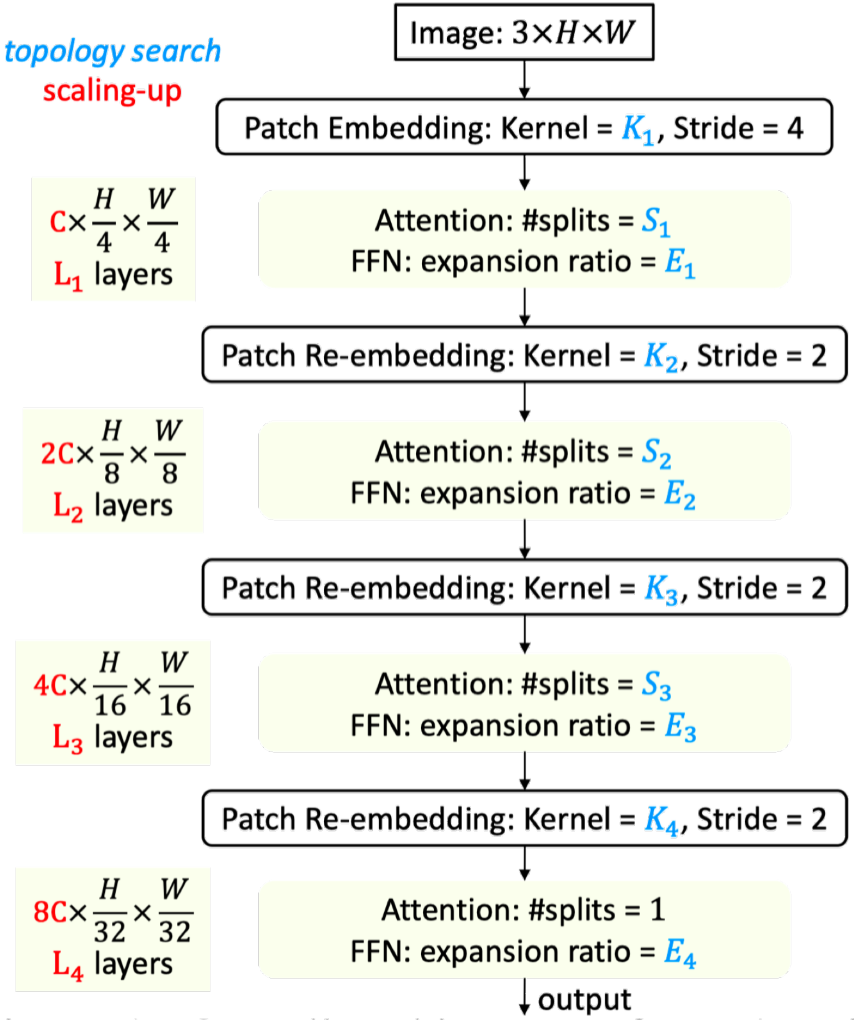
[1] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. ICLR 2021.

As-ViT: Auto-scaling Vision Transformers

- Automated architecture design (no training!)
- Automated scaling-up of ViT (no training!)
  - Meet different constraints in one job.
- Efficient ViT training via progressive re-tokenization.
  - Saves both training FLOPs and time cost.
- State-of-the-art performance on ImageNet and COCO.

Topology & Scaling Space

- **Topology Search**
  - Token (patch) size?
  - Attention Splits? (local vs. global attention)
  - Channel expansion ratio?
- **Model Scaling**
  - Channel Dim for each layer?
  - #Layers for each stage?



Training-free Topology Search via Generalized Complexity Measure of ViTs

- ViT is not a piece-wise linear function (GeLU, Self-attention).
- Measure the complexity of ViTs in a more general way. Input: $\mathbf{h}(\theta) = \sqrt{N}\left[\mathbf{u}^0 \cos(\theta) + \mathbf{u}^1 \sin(\theta)\right]$

ViT network as $\mathcal{N}$, its input-output Jacobian $\boldsymbol{v}(\theta) = \partial_\theta \mathcal{N}(\mathbf{h}(\theta))$ at the input θ, and $\boldsymbol{a}(\theta) = \partial_\theta \boldsymbol{v}(\theta)$

1. Curvature $\kappa = \int (\boldsymbol{v}(\theta)\cdot\boldsymbol{v}(\theta))^{-3/2}\sqrt{(\boldsymbol{v}(\theta)\cdot\boldsymbol{v}(\theta))(\boldsymbol{a}(\theta)\cdot\boldsymbol{a}(\theta)) - (\boldsymbol{v}(\theta)\cdot\boldsymbol{a}(\theta))^2}\,d\theta$
2. Length Distortion $\mathcal{L}^E = \frac{\text{length}(\mathcal{N}(\theta))}{\text{length}(\theta)} = \int \sqrt{\|\boldsymbol{v}(\theta)\|_2}\,d\theta$
3. "Length Distortion + Curvature" $\mathcal{L}^E_\kappa = \int \sqrt{\|\partial_\theta \hat{\mathbf{v}}(\theta)\|_2}\,d\theta \qquad \hat{\mathbf{v}}(\theta) = \mathbf{v}(\theta)/\sqrt{\mathbf{v}(\theta)\cdot\mathbf{v}(\theta)}$

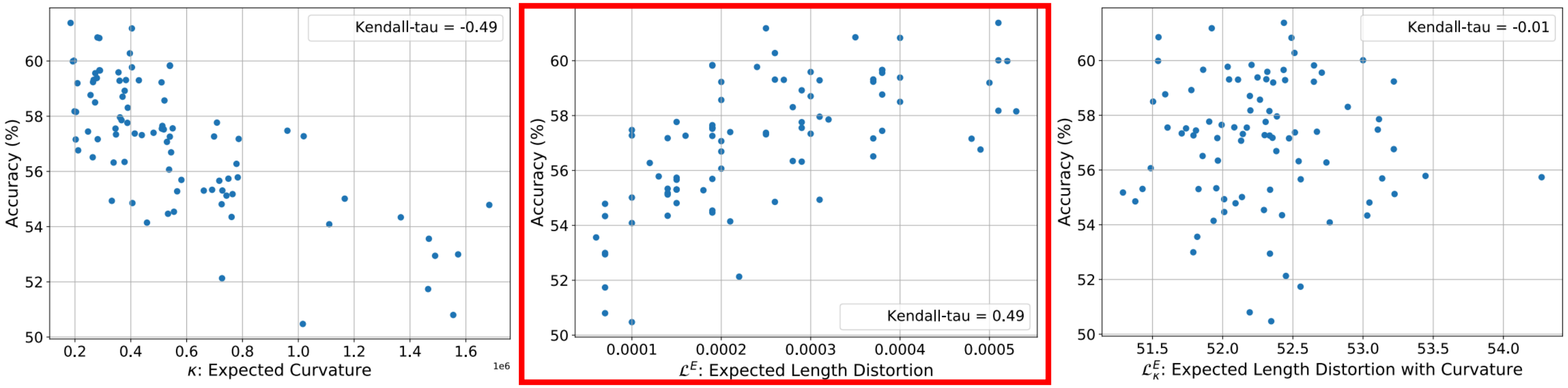


Figure 2: Correlations between κ, $\mathcal{L}^E$, $\mathcal{L}^E_\kappa$ and trained accuracies of ViT topologies from our search space.

Table 2: Complexity Study. τ: Kendall-tau correlation. Time: per ViT topology on average on 1 V100 GPU.

Complexity	τ	Time
κ	-0.49	38.3s
$\mathcal{L}^E$	0.49	12.8s
$\mathcal{L}^E_\kappa$	-0.01	48.2s

https://github.com/VITA-Group/AsViT

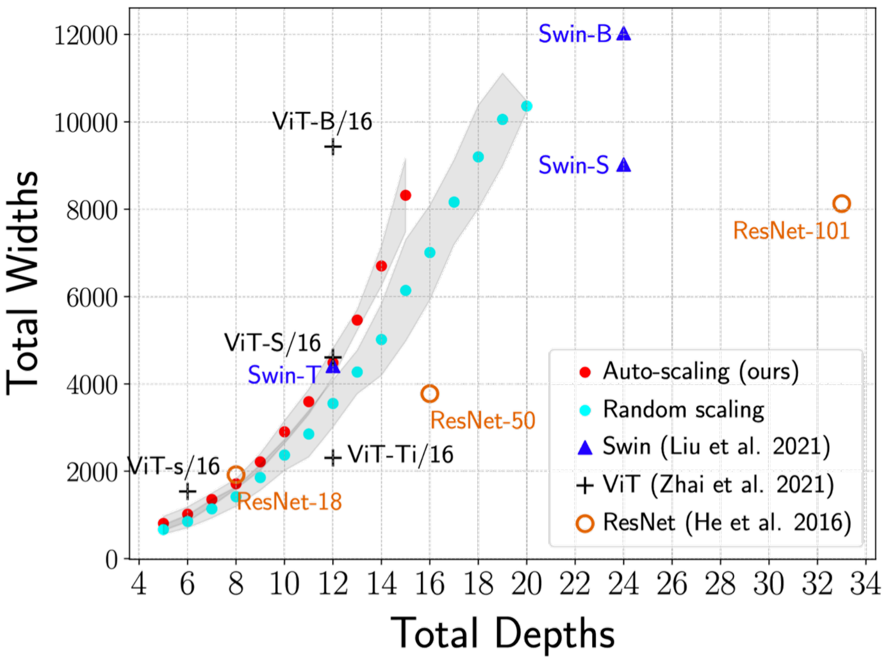
Training-free Model Scaling

- Grow a seed topology into different sizes in a single run.
  - Progressively allocate width & depth.

Algorithm 2: Training-free Auto-scaling ViTs.

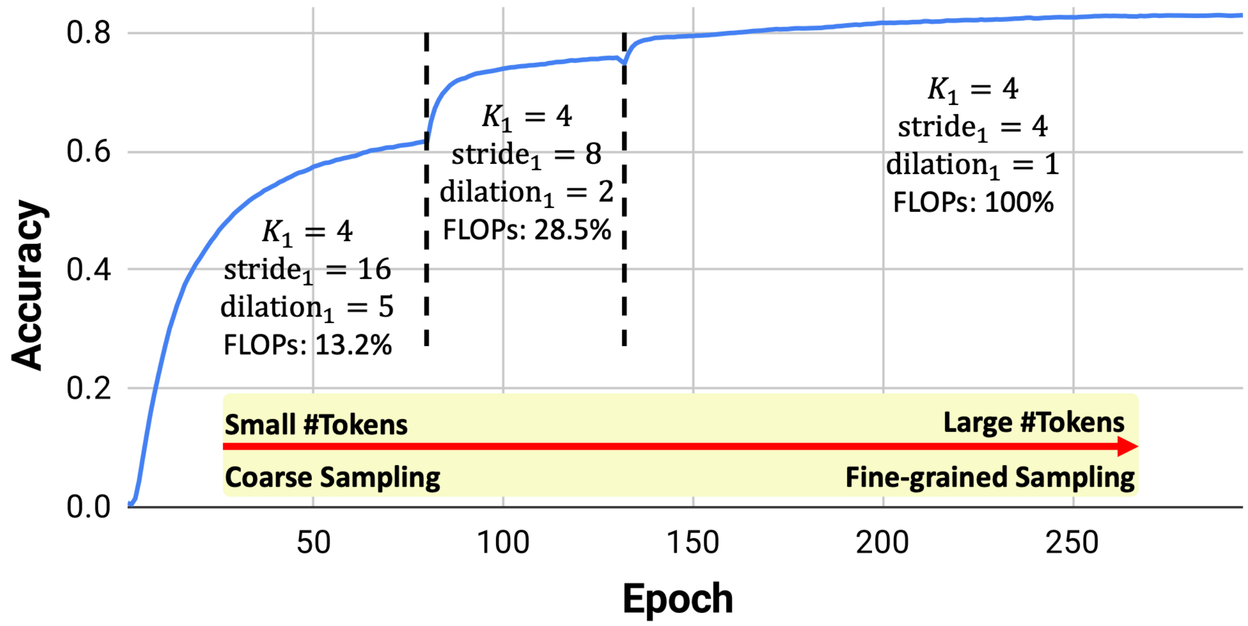
Input: seed As-ViT topology $\boldsymbol{a}_0$, stop criterion (#parameters) P, $t = 0$,
channel expansion ratio choices $\mathcal{C} = \{1.05\times, 1.1\times, 1.15\times, 1.2\times\}$,
depth choices $\mathcal{D} = \{(+1, 0, 0, 0), (0, +1, 0, 0), (0, 0, +1, 0), (0, 0, 0, +1)\}$.
while $|\boldsymbol{a}_t| \leq P$ **do**
  for *each scaling choice* $g_i \in \mathcal{C} \times \mathcal{D}$ **do**
    Scale-up: $\boldsymbol{a}_{t,i} = \boldsymbol{a}_t \leftarrow g_i$. ▷ which stage to deepen, to what extent to widen.
    Calculate $\mathcal{L}_i^E$ and $\kappa_{\Theta,i}$ for $\boldsymbol{a}_{t,i}$.
  Get ranking of each scaling choice $r_{\mathcal{L},i}$ by descendingly sort $\mathcal{L}_i^E$, $i = 1, \cdots, |\mathcal{C} \times \mathcal{D}|$.
  Get ranking of each scaling choice $r_{\kappa_\Theta,i}$ by ascendingly sort $\kappa_{\Theta,i}$, $i = 1, \cdots, |\mathcal{C} \times \mathcal{D}|$.
  Ascendingly sort each scaling choice g_i by $r_{\mathcal{L}^E,i} + r_{\kappa_\Theta,i}$.
  Select the scaling choice g_i^* with the top (smallest) ranking.
  $\boldsymbol{a}_{t+1} = \boldsymbol{a}_t \leftarrow g_i^*$.
  $t = t + 1$.
return *Growed ViT architectures* $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_t$.

Shallower & Wider ViTs



Efficient ViT Training: Progressive Re-tokenization

- Different sampling granularities in the first linear projection layer.
- Update the number of tokens during training.
  - Large stride & dilation => small stride & no dilation.



UTS Google

https://github.com/VITA-Group/AsViT

ImageNet accuracy

State-of-the-art Performance

COCO detection

Table 8: Two-stage object detection and instance segmentation results. We compare employing different backbones with Cascade Mask R-CNN on single model without test-time augmentation.

Backbone	Resolution	FLOPs	Params.	AP_{val}	AP_{val}^{mask}
ResNeXt-101	400∼800×1333	972 B	140 M	48.3	41.6
Swin-B (Liu et al., 2021)	400∼800×1333	982 B	145 M	51.9	45
SpineNet-190 (Du et al., 2020)	1536×1536	2076.8 B	176.2 M	52.2	46.1
As-ViT Large (ours)	1024×1024	1094.2 B	138.8 M	52.7	45.2

ImageNet

Table 5: Image Classification on ImageNet-1k (224 × 224).

Method	Params.	FLOPs	Top-1
RegNetY-4GF (Radosavovic et al., 2020)	21.0 M	4.0 B	80.0%
ViT-S (Dosovitskiy et al., 2020)	22.1 M	9.2 B	81.2%
DeiT-S (Touvron et al., 2020)	22.0 M	4.6 B	79.8%
T2T-ViT-14 (Yuan et al., 2021b)	21.5 M	6.1 B	81.7%
TNT-S (Han et al., 2021)	23.8 M	5.2 B	81.5%
PVT-Small (Wang et al., 2021)	24.5 M	3.8 B	79.8%
CaiT XS-24 (Touvron et al., 2021)	26.6 M	5.4 B	81.8%
DeepVit-S (Zhou et al., 2021)	27 M	6.2 B	82.3%
ConViT-S (d'Ascoli et al., 2021)	27 M	5.4 B	81.3%
CvT-13 (Wu et al., 2021)	20 M	4.5 B	81.6%
CvT-21 (Wu et al., 2021)	32 M	7.1 B	82.5%
Swin-T (Liu et al., 2021)	29.0 M	4.5 B	81.3%
BossNet-T0 (Li et al., 2021)	-	3.4 B	80.8%
AutoFormer-s (Chen et al., 2021c)	22.9 M	5.1 B	81.7%
GLiT-Small (Chen et al., 2021a)	24.6 M	4.4 B	80.5%
As-ViT Small (ours)	29.0 M	5.3 B	81.2%
RegNetY-8GF (Radosavovic et al., 2020)	39.0 M	8.0 B	81.7%
T2T-ViT-19 (Yuan et al., 2021b)	39.2 M	9.8 B	82.2%
CaiT S-24 (Touvron et al., 2021)	46.9 M	9.4 B	82.7%
ConViT-S+ (d'Ascoli et al., 2021)	48 M	10 B	82.2%
ViT-S/16 (Dosovitskiy et al., 2020)	48.6 M	20.2 B	78.1%
Swin-S (Liu et al., 2021)	50.0 M	8.7 B	83.0%
DeepViT-L (Zhou et al., 2021)	55 M	12.5 B	82.2%
PVT-Medium (Wang et al., 2021)	44.2 M	6.7 B	81.2%
PVT-Large (Wang et al., 2021)	61.4 M	9.8 B	81.7%
T2T-ViT-24 (Yuan et al., 2021b)	64.1 M	15.0 B	82.6%
TNT-B (Han et al., 2021)	65.6 M	14.1 B	82.8%
BossNet-T1 (Li et al., 2021)	-	7.9 B	82.2%
AutoFormer-b (Chen et al., 2021c)	54 M	11 B	82.4%
ViT-ResNAS-t (Liao et al., 2021)	41 M	1.8 B	80.8%
ViT-ResNAS-s (Liao et al., 2021)	65 M	2.8 B	81.4%
As-ViT Base (ours)	52.6 M	8.9 B	82.5%
RegNetY-16GF (Radosavovic et al., 2020)	84.0 M	16.0 B	82.9%
ViT-B/16 (Dosovitskiy et al., 2020)†	86.0 M	55.4 B	77.9%
DeiT-B (Touvron et al., 2020)	86.0 M	17.5 B	81.8%
ConViT-B (d'Ascoli et al., 2021)	86 M	17 B	82.4%
Swin-B (Liu et al., 2021)	88.0 M	15.4 B	83.3%
GLiT-Base (Chen et al., 2021a)	96.1 M	17.0 B	82.3%
ViT-ResNAS-m (Liao et al., 2021)	97 M	4.5 B	82.4%
CaiT S-48 (Touvron et al., 2021)	89.5 M	18.6 B	83.5%
As-ViT Large (ours)	88.1 M	22.6 B	83.5%

https://github.com/VITA-Group/AsViT

Thank you!

Code

Paper

https://github.com/VITA-Group/AsViT