Early Convolutions Help Transformers See Better

TeteXiao, Mannat Singh, EricMintun, Trevor Darrell, Piotr Dollár, Ross Girshick

Traditional Computer Vision Architecture

Strengths of CNNs

- Able to effectively use SGD training approaches
- Standard hyper parameter values
- Basic data augmentations
- Known training recipes

Problem: Optimizability

Limitations of ViT

- Sensitive to **optimizer** choice (AdamW vs. SGD)
- Sensitive to **dataset specific** learning hyper params
- Sensitive to **training schedule** length
- Sensitive to **network depth**
- Struggle to use prior training recipes

Related Work

ViT

- Stand-alone self-attention without convolutions
- Use image patches and positional encodings as transformer input
- Self-attention with a non-local means, integrated with a ResNet
- Multi-scale networks, increasing transformer depth, locality priors
- Efficiency and accuracy, not optimizability.

Inductive Bias

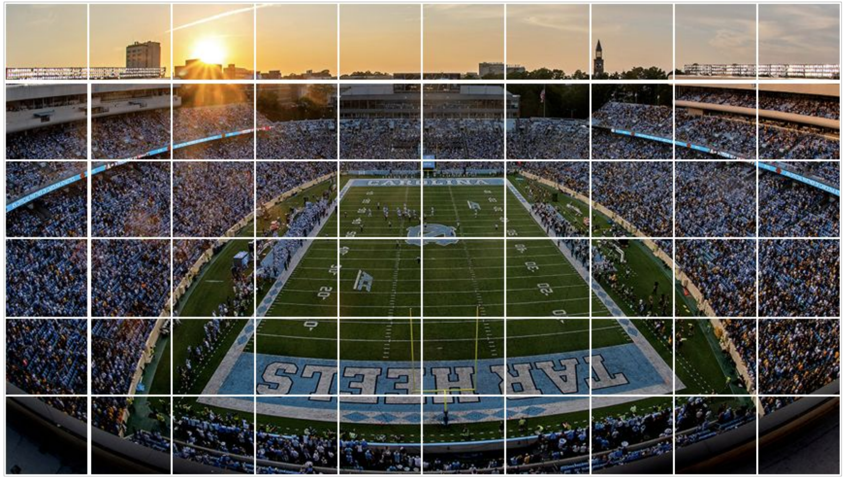
Pattern detection

Inductive bias: anything which makes the algorithm learn one pattern instead of another pattern

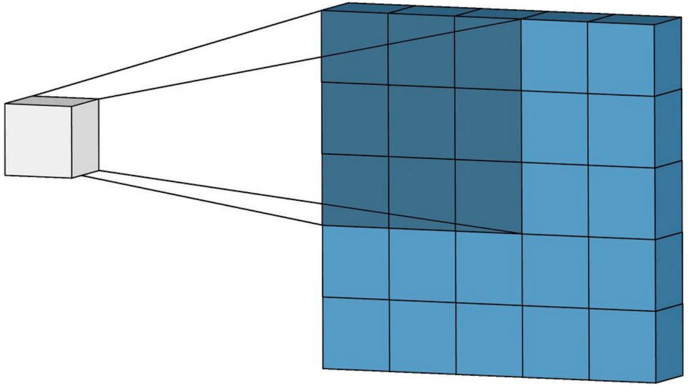
ViT	CNNs
global processing performed by multi-headed self-attention	bias towards local processing

Stems for ViT Models

Patchify Stem vs Convolutional Stem



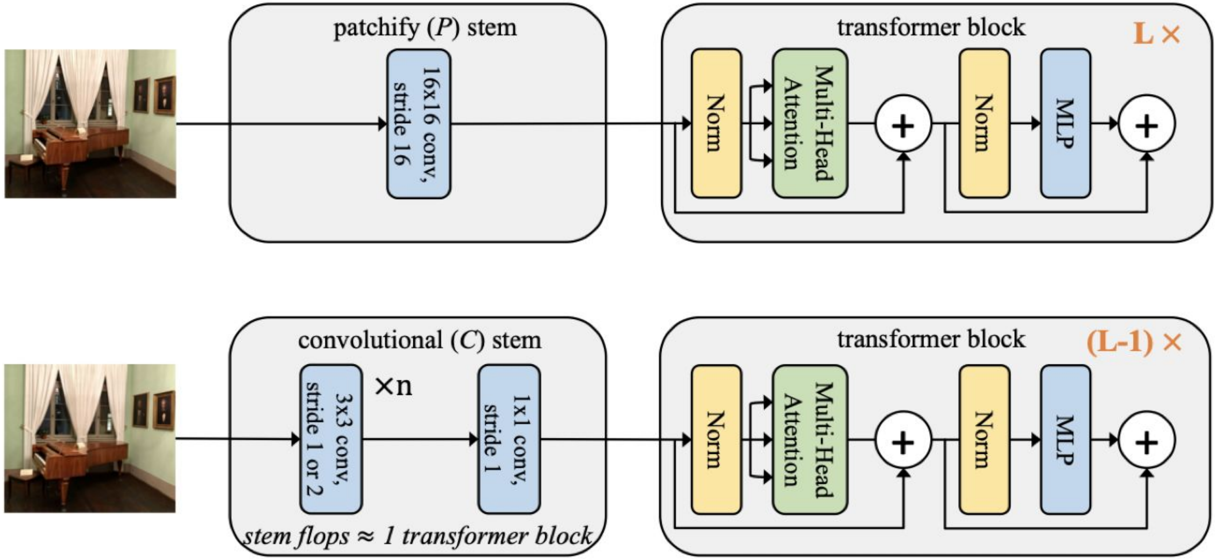
Patchify Stem



Convolutional Stem

Proposed Architecture

Modification to Stem



Original ViT (baseline, termed ViT_P):

- *Sensitive to lr and wd choice*
- *Converges slowly*
- *Works with AdamW, but not SGD*
- *Underperforms sota CNNs on ImageNet*

Ours (termed ViT_C, same runtime):

- ✓ *Robust to lr and wd choice*
- ✓ *Converges quickly*
- ✓ *Works with AdamW, and also SGD*
- ✓ *Outperforms sota CNNs on ImageNet*

AdamW (Adam with Weight Decay)

Optimization Algorithm for Neural Networks

Extension of Adam optimizer with a weight decay term to address overfitting.

Learning Rate Adaptation: Adjusts learning rates for each parameter individually.
Weight Decay: Penalizes large weights, acting as regularization.

Pros:

- **Adaptive Learning Rates:** Faster convergence with individually adapted rates.
- **Regularization:** Weight decay helps prevent overfitting.

Cons:

- **Computational Complexity:** More computationally expensive compared to SGD.
- **Hyperparameter Tuning:** Requires careful tuning despite adaptive features.

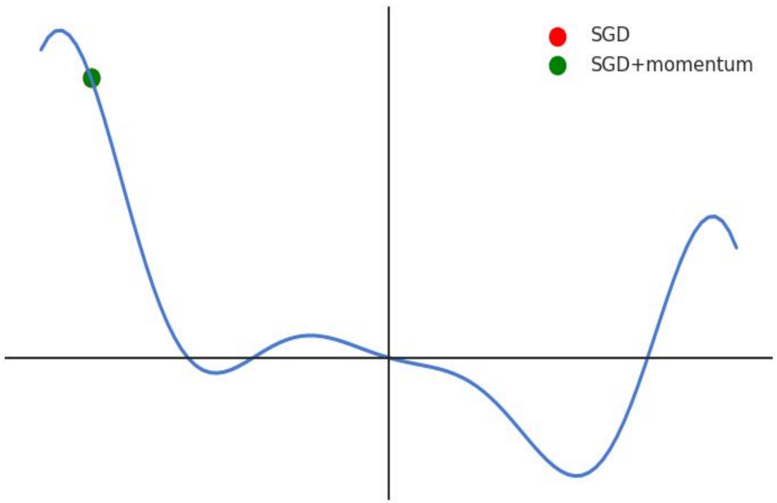
Stochastic Gradient Descent (SGD)

Optimization Algorithm for Neural Networks

Classic optimization algorithm minimizing the loss function during neural network training.

Learning Rate: Determines step size during parameter updates.
Momentum: Accelerates convergence, especially in high-curvature regions

Pros: Simplicity, Memory Efficiency
Cons: Hyperparameter Sensitivity, Noisy Updates



Model Size

Ensuring Parity

ViT_P Modifications:

- reduced the MLP multiplier from **4** to **3** for the **1GF** and **4GF** models
- reduce the number of transformer blocks from **24** to **14** for the **36GF** model

ViT_C Modifications:

- One fewer transformer Block

model	ref model	hidden size	MLP mult	num heads	num blocks	flops (B)	params (M)	acts (M)	time (min)
ViT_P-1GF	~ViT-T	192	3	3	12	1.1	4.8	5.5	2.6
ViT_P-4GF	~ViT-S	384	3	6	12	3.9	18.5	11.1	3.8
ViT_P-18GF	=ViT-B	768	4	12	12	17.5	86.7	24.0	11.5
ViT_P-36GF	$\frac{3}{5}$ViT-L	1024	4	16	14	35.9	178.4	37.3	18.8

model	hidden size	MLP mult	num heads	num blocks	flops (B)	params (M)	acts (M)	time (min)
ViT_C-1GF	192	3	3	11	1.1	4.6	5.7	2.7
ViT_C-4GF	384	3	6	11	4.0	17.8	11.3	3.9
ViT_C-18GF	768	4	12	11	17.7	81.6	24.1	11.4
ViT_C-36GF	1024	4	16	13	35.0	167.8	36.7	18.6

Measuring Optimizability

Establishing Metrics for Evaluation

Optimizability: The ability of a model to be effectively trained and optimized.

Metrics introduced:

- *training length stability:* the gap to asymptotic accuracy
- *optimizer stability:* accuracy gap between AdamW and SGD
- *hyperparameter stability:* comparing the error distribution functions (EDFs)
- *peak performance:* the result of a model at 400 epochs using its best-performing optimizer and parsimoniously tuned lr and wd values

Stability Experiments

Comparing 3 Types of Models

Compare ViT models with image patch stem to ViT with convolutional stem

Compare to RegNetY, a SOTA CNN that is easy to optimize, as a reference point for good stability

Use ImageNet-1k's standard training and validation sets

Report top-1 error

Data augmentations:

- AutoAugment
- Mixup
- CutMix
- Label smoothing

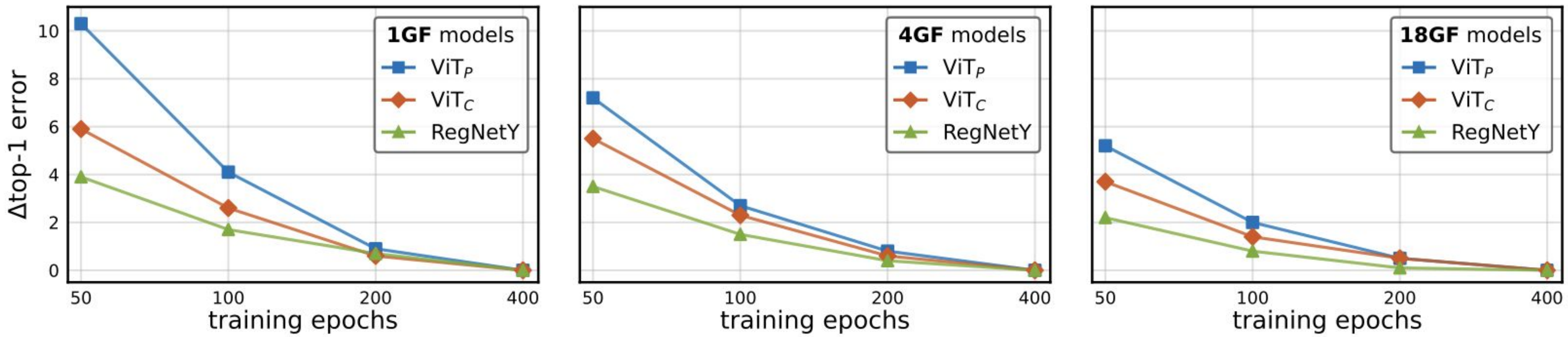
Training Length Stability

24 variations of **ViT**s with **AdamW** optimizer

- Stem (Patch or convolutional)
- Model size GF (1, 4, 18)
- Epochs (50, 100, 200, 400)

12 variations of **RegNetY** with **SGD**

- Model size GF (1, 4, 16)
- Epochs (50, 100, 200, 400)



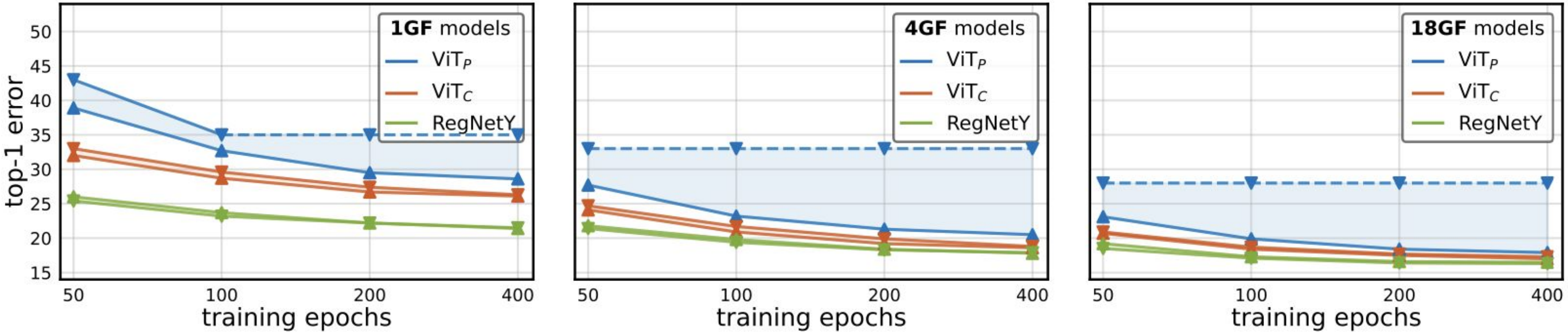
Optimizer Stability

48 variations of **ViTs**

- Stem (Patch or convolutional)
- Model size GF (1, 4, 18)
- Epochs (50, 100, 200, 400)
- Optimizer (AdamW, SGD)

24 variations of **RegNetY**

- Model size GF (1, 4, 16)
- Epochs (50, 100, 200, 400)
- Optimizer (AdamW, SGD)



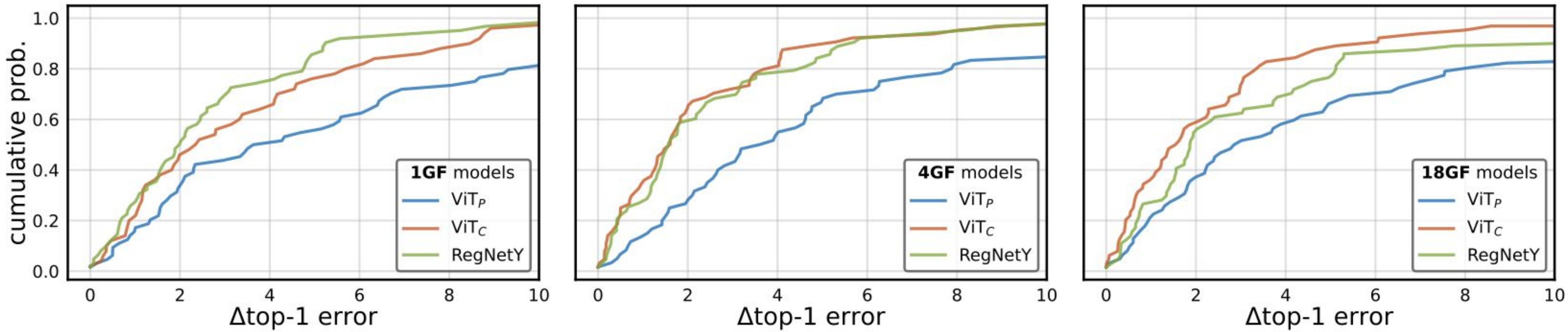
Learning Rate and Weight Decay Stability (AdamW)

3 model variations

- ViT with patch stem
- ViT with convolutional stem
- RegNetY

64 instances of each model

- 50 epochs
- Random lr and wd in interval around optimal lr and wd for each model



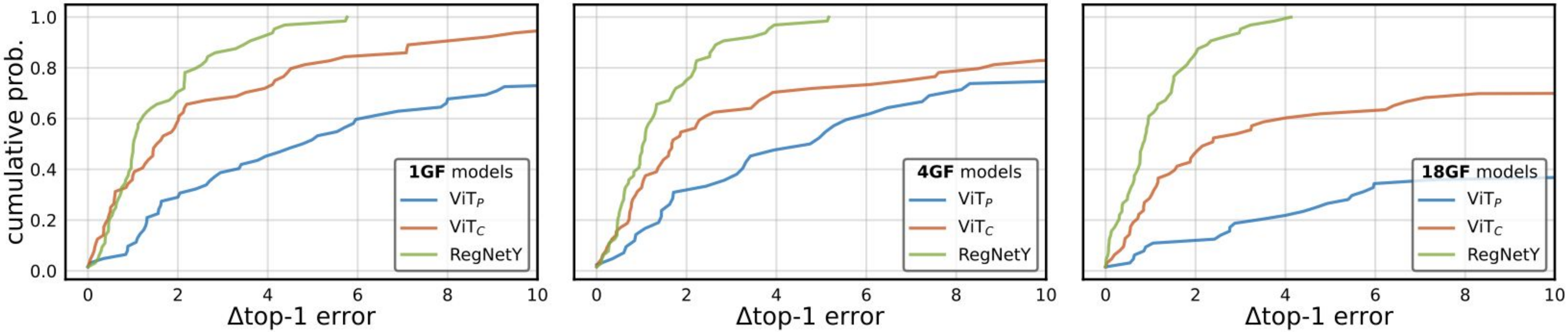
Learning Rate and Weight Decay Stability (SGD)

3 model variations

- ViT with patch stem
- ViT with convolutional stem
- RegNetY

64 instances of each model

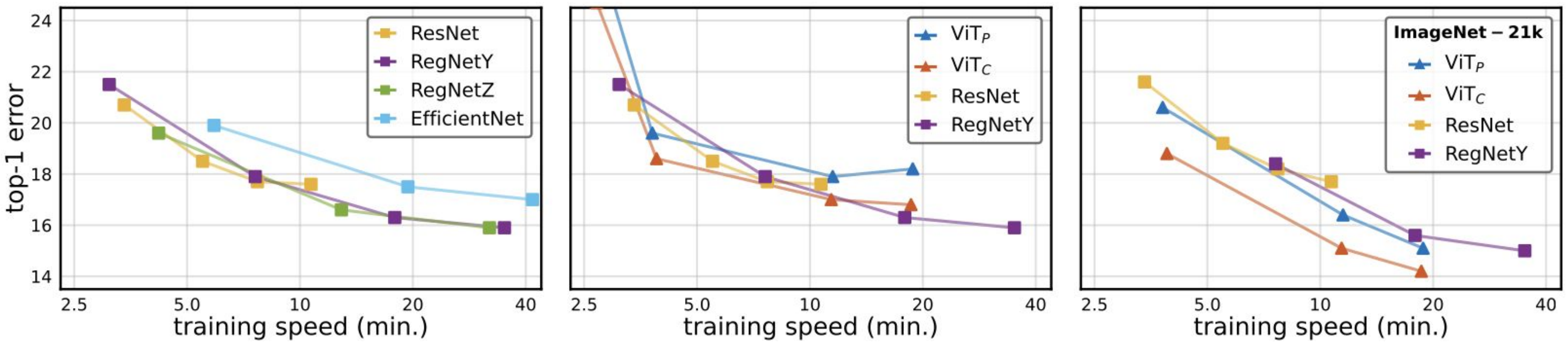
- 50 epochs
- Random lr and wd in interval around optimal lr and wd for each model



Peak Performance

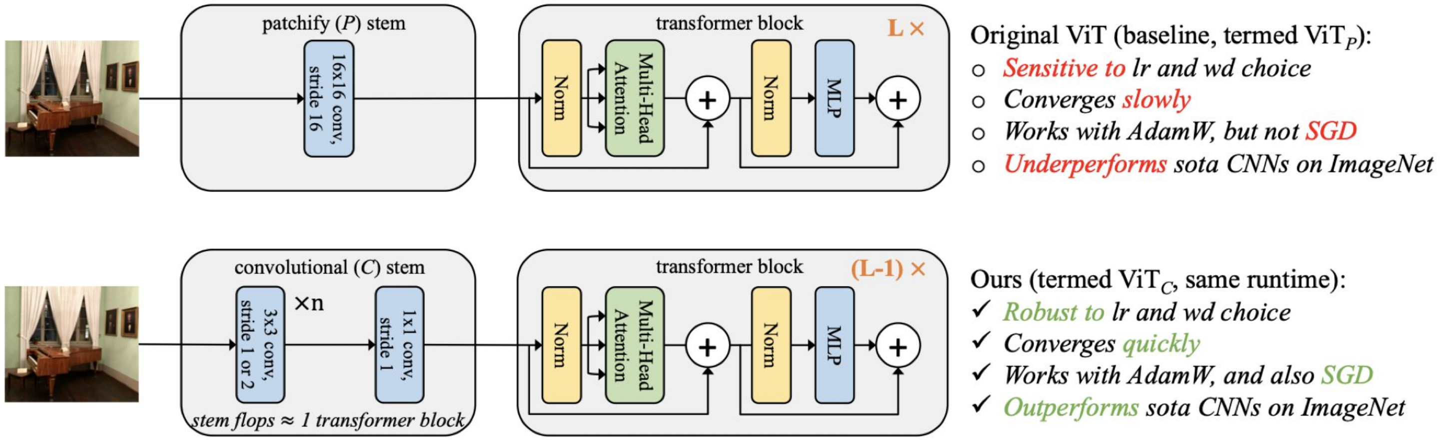
We see a boost in performance as data scales up

- With ImageNet-1k ViTc is not able to beat CNN
- On ImageNet-21k ViTc is able to beat CNN and ViTp



Summary

Injecting Convolutional Inductive Bias into ViTs



- Builds on the ViT and proposes seemingly trivial change to stem which greatly changes optimization behavior.
- Results are consistent across a wide spectrum of model complexities and dataset scales