Research progress briefing

Presenter: 林保羅 (Paulo Linares)

Advisor: 傅楸善(Chiou-Shann Fuh) 博士

Lightweight ResNet-based block for low-rank visual representation learning

E-mail: d12922028@csie.ntu.edu.tw

Foundation models and visual representation learning

- **Architectures:** foundation models, lightweight models
- **Training regimes:** pre-training + fine-tuning
- **Downstream tasks**
  - How? : linear probing, fine-tuning, PEFT
  - Visual tasks: image classification, pose estimation, semantic segmentation

Scaling issues: Very large models suffer from exploding gradients, making optimization and training stability increasingly difficult.

Self-supervised pre-training: Learn representations without manual labels. Pretext tasks: contrastive learning, masked image modeling, reconstruction

Supervised pre-training: Train on large labeled datasets (e.g. ImageNet 1K, ImageNet 21K). Provides strong initialization for downstream tasks.

Large Vision Models

- Since the introduction of **Vision Transformers (ViT)**, significant effort has focused on developing scalable **Large Vision Models (LVMs)**. Early ViTs performed worse than CNN backbones on **dense prediction tasks** (object segmentation and detection).

- To address this limitation, researchers proposed **CNN–ViT hybrid architectures**.

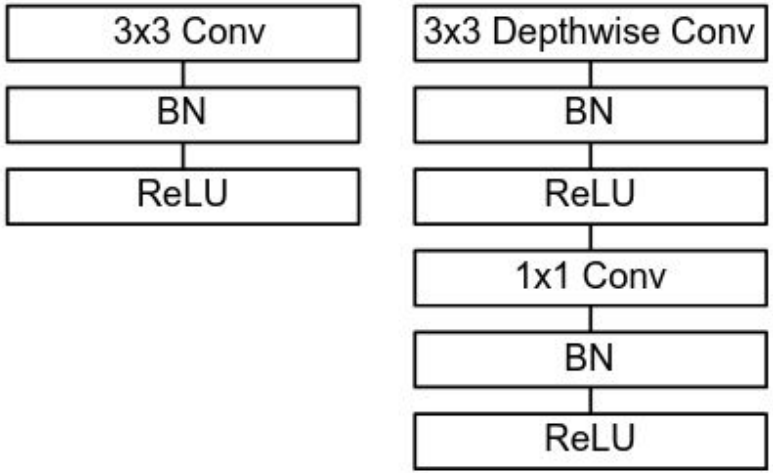
- **Strategy 1:** replaces **fully connected layers with convolutions**. Requires changes to token embedding, self-attention, and feed-forward modules.

- **Strategy 2: partially integrates convolutions**, using token/feature reshaping to apply convolution while preserving the core ViT structure.

Lightweight models

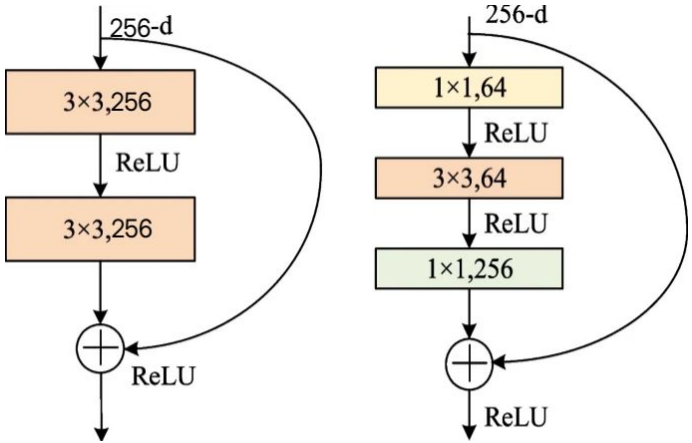
- Depthwise separable convolution (CNN-based)
- Bottleneck modules
- Knowledge Distillation
- Model compression

Lightweight models (Depthwise separable convolution)

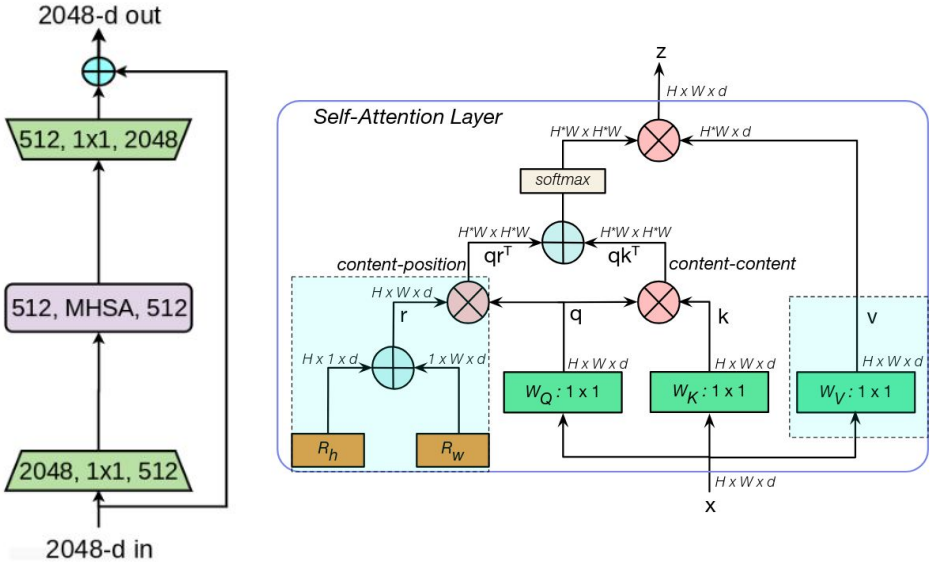


Reduce parameters by using depthwise convolutions per channel followed by 1×1 pointwise convolutions

Lightweight models (Bottleneck modules)

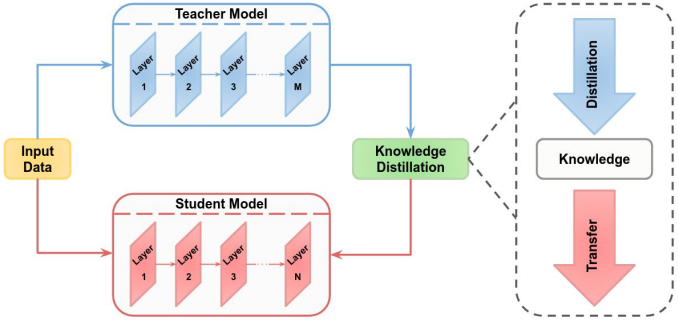


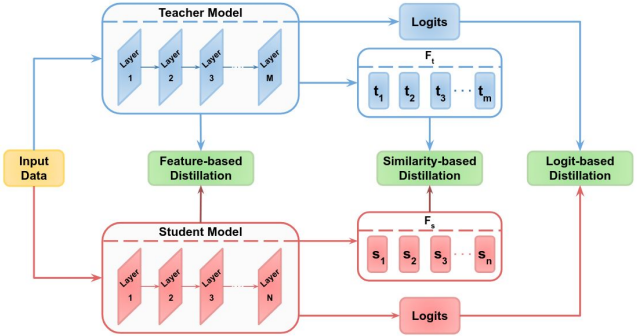
ResNet bottleneck



Bottleneck Transformer (BoTNET)

Lightweight models (Knowledge distillation - KD)





- During offline KD the teacher is frozen while the student is trained.
- **Feature-based KD**: minimizing the differences in features between the teacher and student
- **Similarity-based KD:** considers the pairwise similarities between features or instances
- **Logit-based KD**: force the student model to mimic the final predictions of the teacher.
- **Lightweight models with KD**: the teacher is orders of magnitude larger than the student.

[A Comprehensive Survey on Knowledge Distillation]

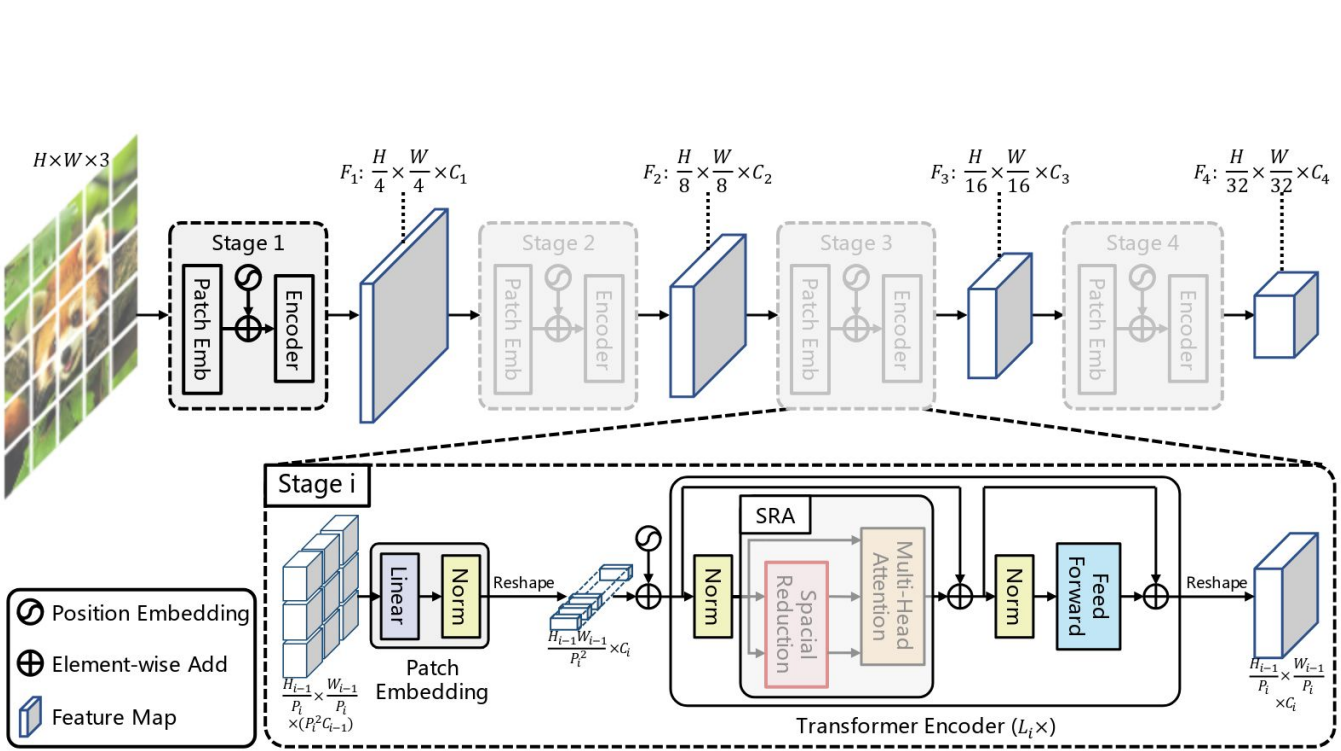
Pyramid Vision Transformer (PVT)

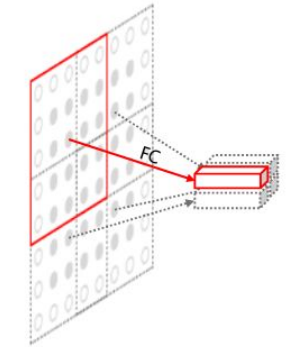
W. Wang et al. (ICCV ‘21)

- Hybrid CNN–Transformer architecture
- Supervised pre-training on ImageNet-2012 for image classification

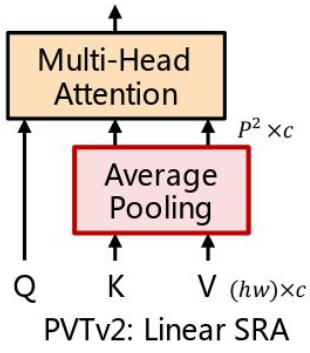
- Limitations of PVT-V1:
  - High computational cost for high-resolution images
  - Non-overlapping patches break local spatial continuity

- PVT-V2 improvements:
  - Introduces a hybrid CNN–Transformer design
  - Linear Spatial Attention Reduction to downsample token sequences
  - Overlapping patch embeddings using convolutions and a convolutional FFN





PVTv2: Overlapping Patch Embedding



PVTv2: Linear SRA

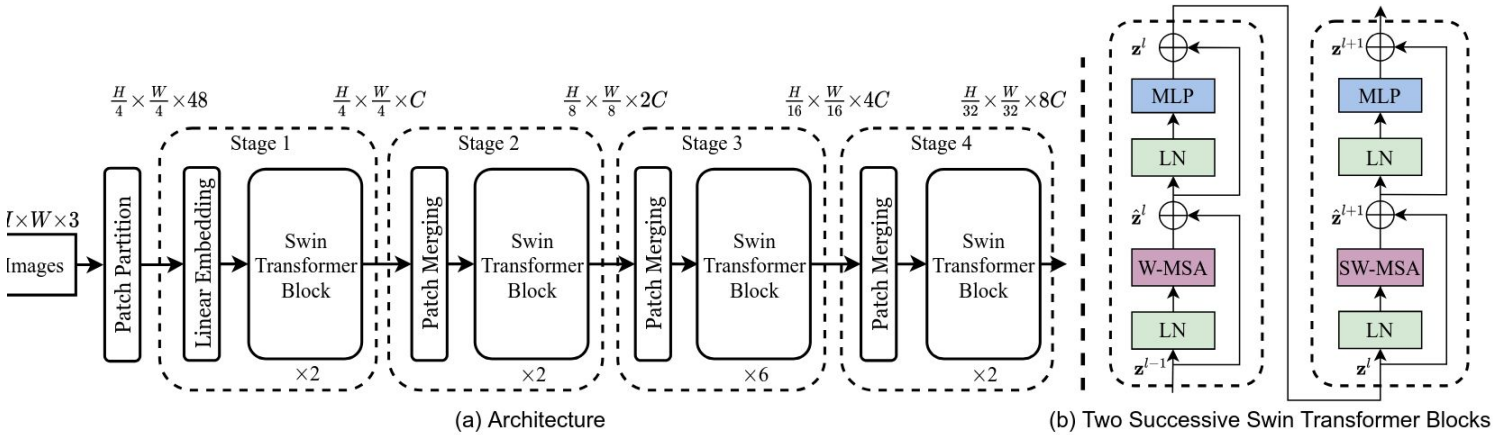
Patch embedding: Non-overlap (ViT) vs. Overlapping (Convolution)

SRA: Spatial Reduction Attention: Convolution vs. Average Pooling

Swin Transformer V2

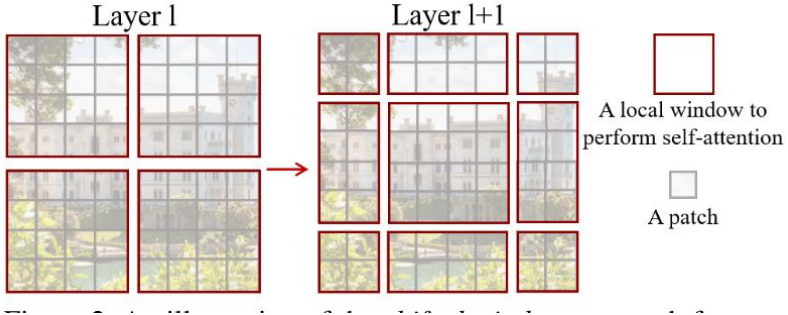
Z. Liu et al. (CVPR ‘22)

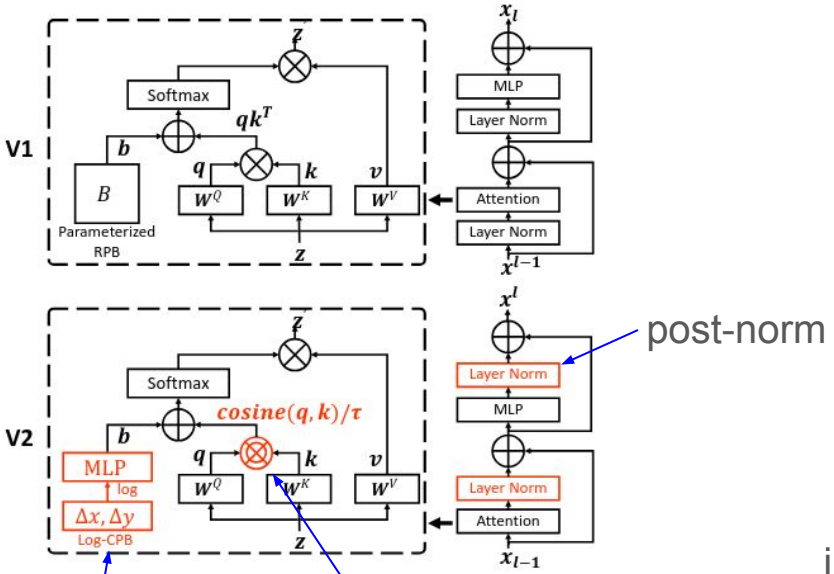
- Hybrid CNN–Transformer architecture
- Addresses activation instability when scaling Vision Transformers, proposing a post-norm residual design to improve training stability
- Emphasizes the importance of self-supervised pre-training for Large Vision Models (LVMs)
- Pre-training datasets:
  - Supervised: ImageNet-1K, ImageNet-21K
  - Self-supervised: subset of JFT-3B (~70M images)



Regular partition

Shifted partition





post-norm

Continuous relative position bias
+
Log-spaced coordinates

Cosine attention

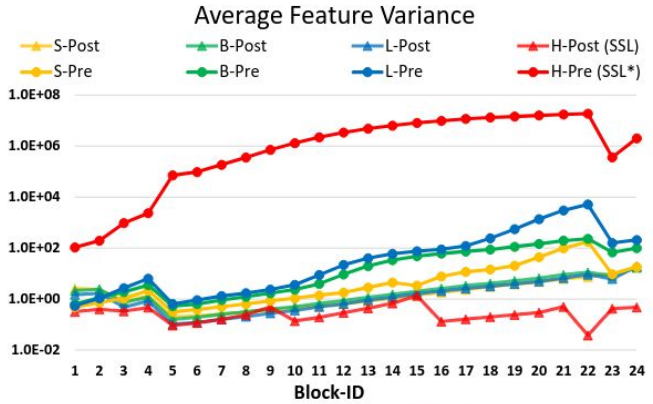


Figure 2. The Signal Propagation Plot [6, 76] for various model sizes. H-size models are trained at a self-supervised learning phase, and other sizes are trained by an image classification task. * indicates that we use a 40-epoch model before it crashes.

instability issue when scaling up model capacity (pre vs. post normalization)

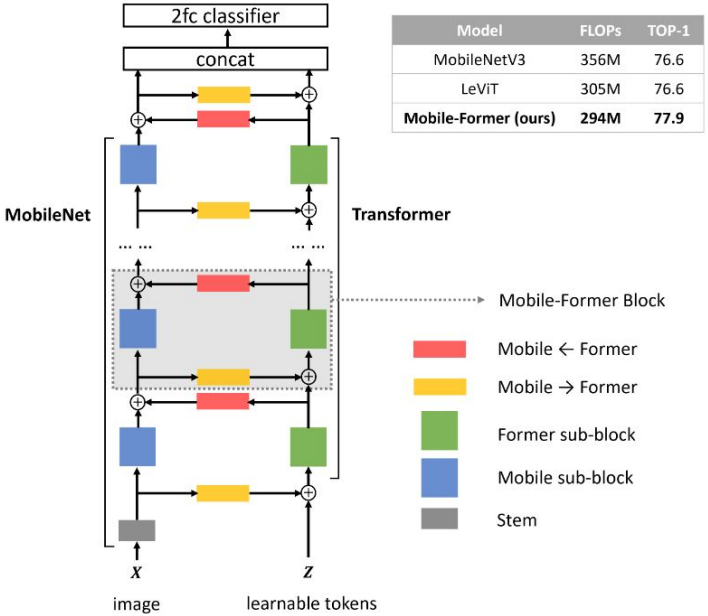
Signal Propagation Plot: statistics of the hidden activations at different points inside a network (conditioned on an input)

SimMIM: Simple Framework for Masked Image Modeling (CVPR '22)

Mobile-Former

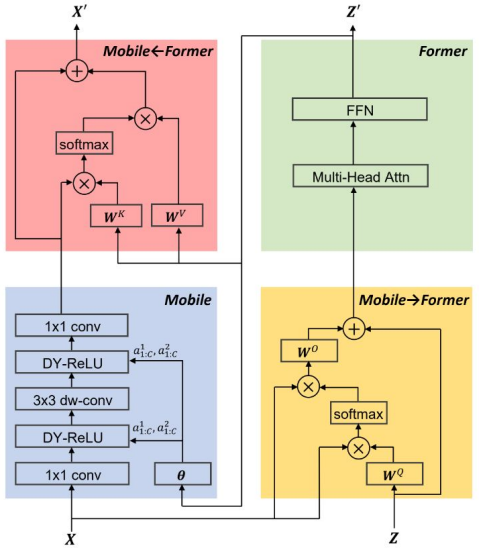
Y. Chen et al. (CVPR ‘22)

- **Hybrid CNN–Transformer architecture**
- Designed to capture **local features and global interactions efficiently**
- Introduces a **two-way bridge between MobileNet and Transformer branches via cross-attention**
- **Mobile-Former block components:**
  - Mobile module
  - Former module
  - Mobile → Former interaction
  - Former → Mobile interaction
- Uses **Dynamic ReLU** in the mobile sub-block for adaptive activation



Model	FLOPs	TOP-1
MobileNetV3	356M	76.6
LeViT	305M	76.6
Mobile-Former (ours)	**294M**	**77.9**

Mobile-Former framework
(two backbones: ViT + MobileNet)

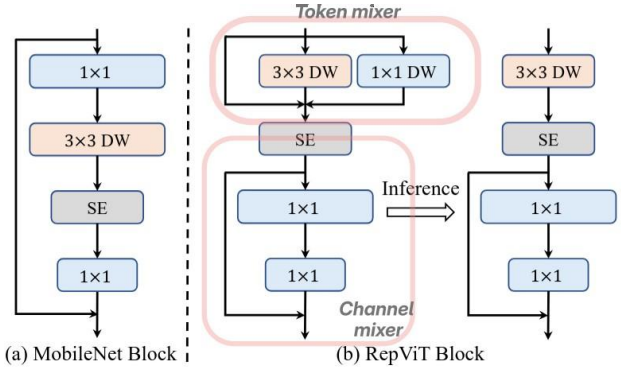


Cross-attention between
the two backbones

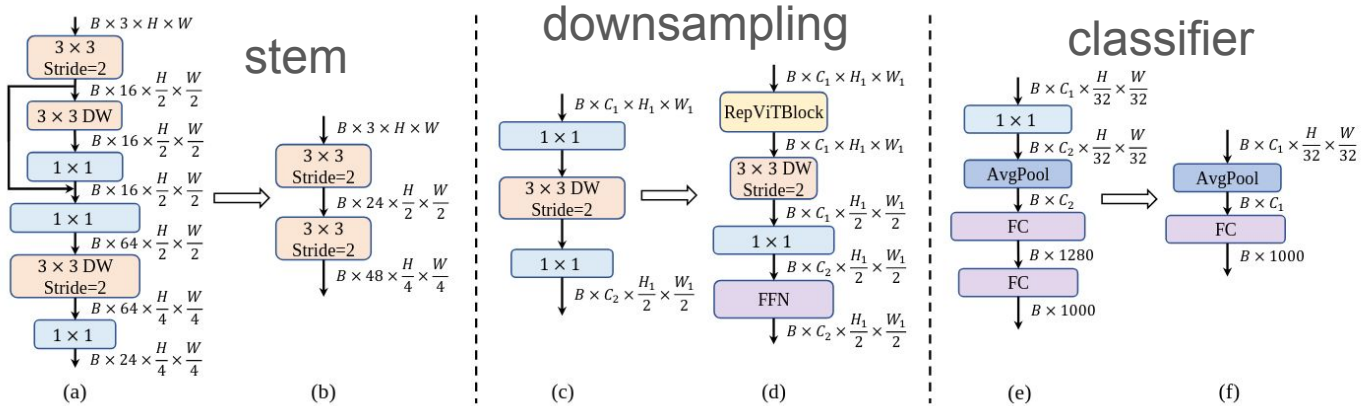
Revisiting MobileNet from a ViT perspective (RepViT)

A. Wang et al. (CVPR ‘24)

- Supervised pre-training on ImageNet-1K
- Applied to downstream tasks such as object detection and semantic segmentation
- Modernizes MobileNet design:
  - Token mixer / channel mixer block design
  - Improved macro architecture (RepViT block placement)
  - Refined micro design within RepViT blocks
- **SAM-RepViT:** replaces the **SAM (Segment Anything Model) image encoder** with RepViT, achieving **comparable performance to heavier SAM variants.**



Block design



Macro design

CNN-transformer hybridization

Pure Transformer blocks are suboptimal for vision because they lack the 2D inductive bias needed to model spatial structure, motivating CNN–ViT hybrid backbones that inject convolutional locality into Vision Transformers.

CNNs → strong spatial inductive bias

Transformers → strong scaling

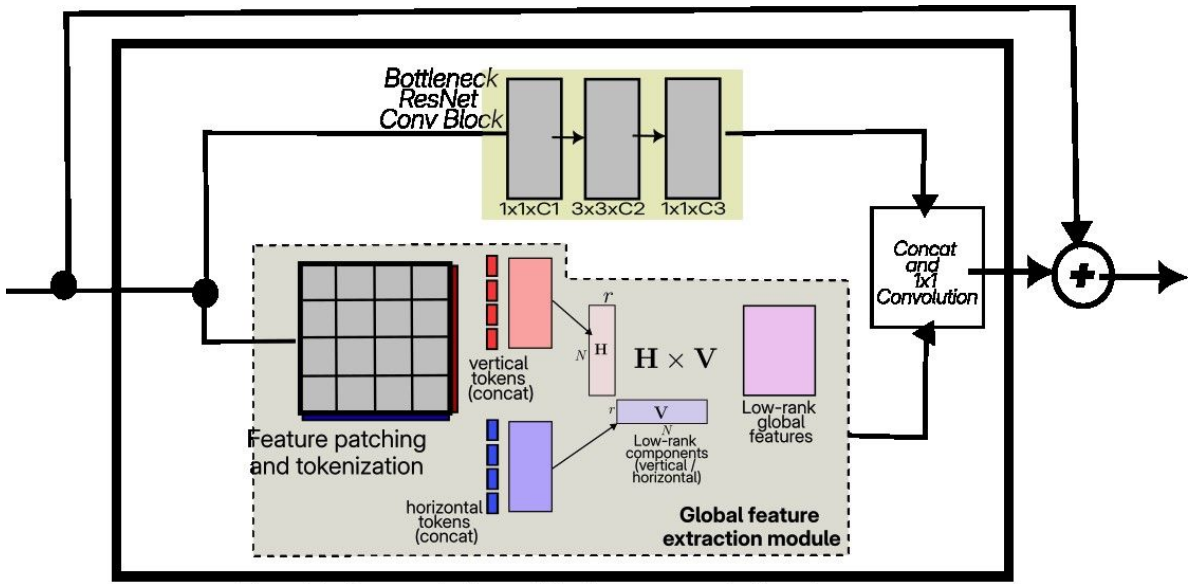
Hybrid type	Structure
Sequential	CNN → Transformer
Interleaved	Conv and attention blocks alternate
Parallel	CNN branch + ViT branch

How to design efficient networks to effectively encode both local processing and global interaction? - Mobile-Former

Combine convolution and vision transformer in series, either using convolution at the beginning or intertwining convolution into each transformer block

Proposed method

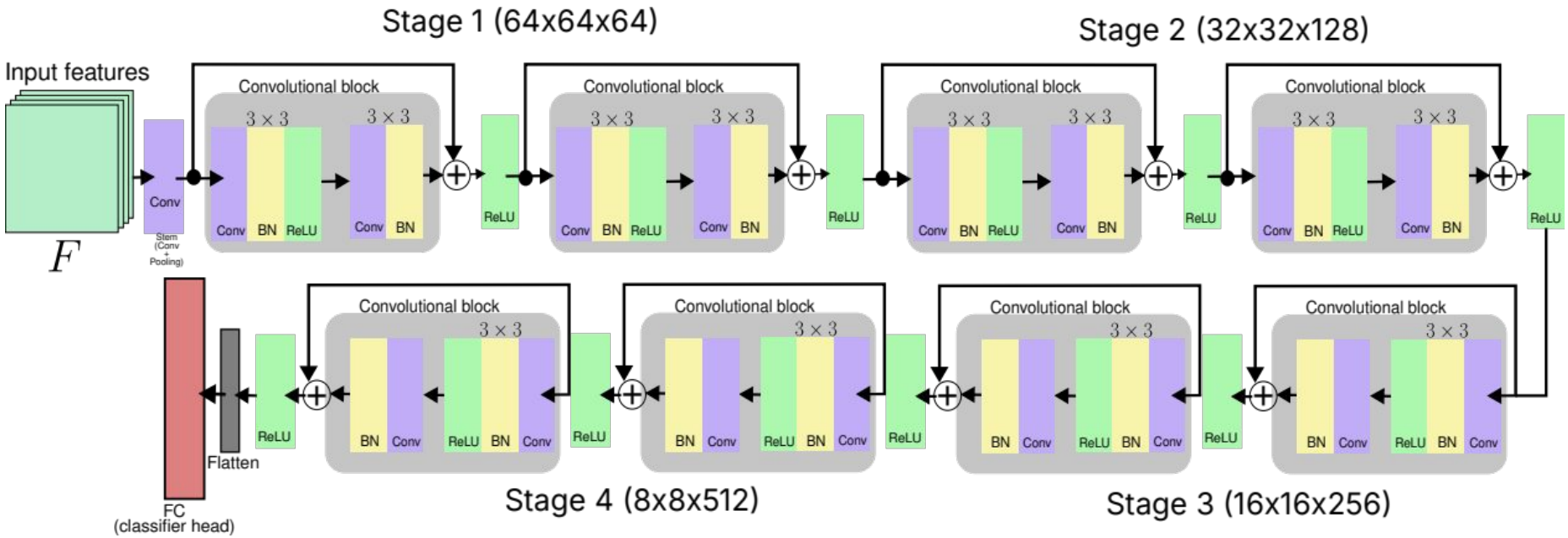
- Low-rank feature generator (tokenization, MLPs) + Local features (CNNs)
- Proposed setups: ResNet backbones with Low-rank blocks
- **Block functionality testing**: Backward incorporation within backbones



ResNet module with Low-rank features

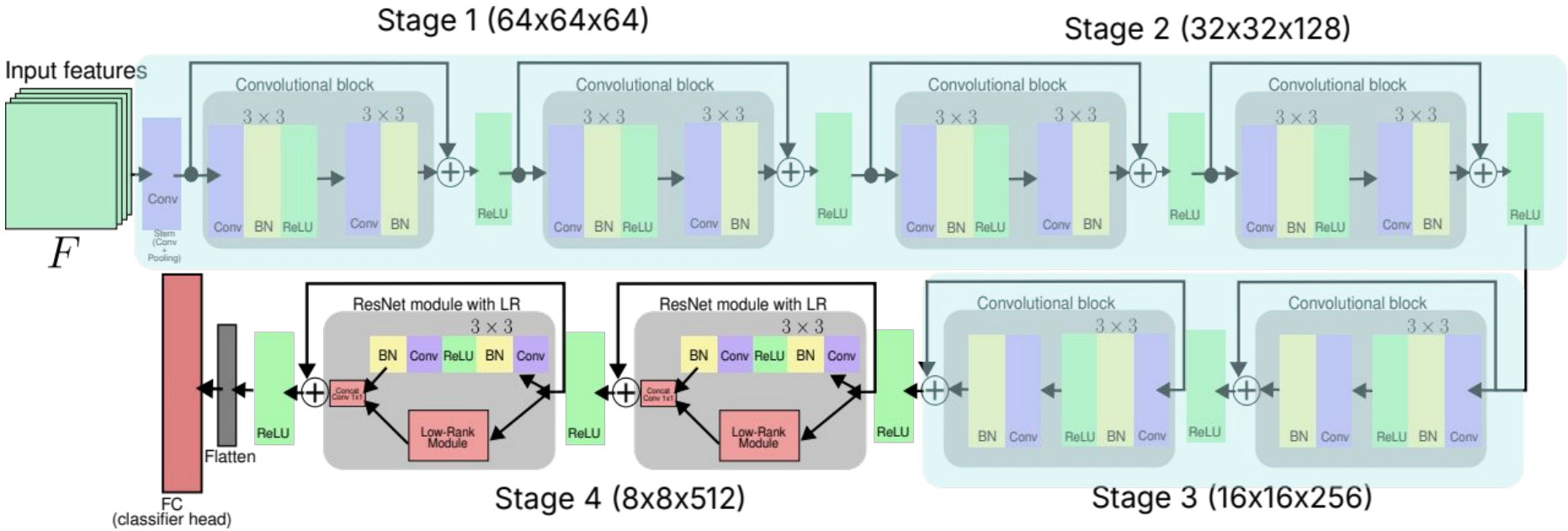
Proposed method

- Initial experiments: ResNet18 as the backbone (fast prototyping)
- Use the pre-trained ResNet18 from Torchvision (68.85% acc. on ImageNet1K)



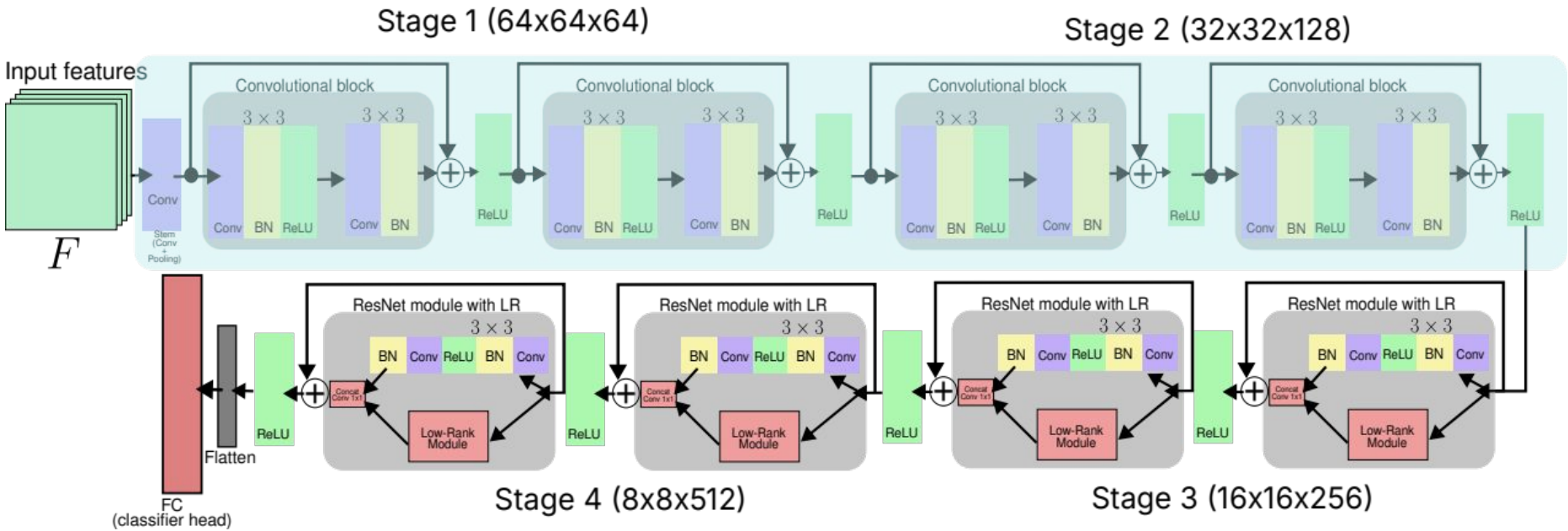
Proposed method (backward incorporation)

- Freeze the first 3 stages
- Replace the modules at the **4th stage** with the proposed ResNetLR module.
- Train on ImageNet 1K (supervised)



Proposed method (backward incorporation)

- Freeze the first 2 stages
- Replace the modules at the **3th, 4th stage** with the proposed ResNetLR module.
- Train on ImageNet 1K (supervised)



Proposed method (backward incorporation)

- Why **backward incorporation**?
  - Training from scratch on ImageNet takes around 4 days.
  - Inexpensive testing of the LowRank module (different variants)
  - The first stages act as the network stem: bigger features .
- Analyze the t**raining dynamics** during the first two experiments.
- If the accuracy does not outperform the baseline's (or attain a comparable performance) -> **modify the LowRank module**.
- Each **experimental trial** takes 2-3 hours (10 epochs) on a workstation with:
  - AMD Ryzen 5 5600 6-Core Processor
  - Memory: 32 GB DDR4 RAM
  - Graphics card: GeForce RTX 3090
  - Storage: 1TB Nvme
- The proposed module scales with feature size (spatial dimensions)
  - Another approach: Replace the ResNet modules (incorporation of ResNet + LowRank) in a forward manner.
  - Increasing the channel number does not affect the module's size significantly.

Models' parameter count

- C_{in}: input channels
- C_{out}: output channels
- N: spatial size at the module input
- p: patch size
- ~~d: head dimension~~
- R: low-rank factor
- ~~B: number of tokenizer blocks~~
- N_{eff}: effective spatial size inside LR path
- ~~$N_p = N_{\text{eff}}/p$: number of patches per axis~~
- ~~$T = 4d$: tokenizer width (`BTSIZE`)~~

Stage	Output size	Channels	Params (Regular)	Params (LowRank)	
Input	256×256	3	0	0	
Stem	64×64	64	9,536	9,536	
Layer1	64×64	64	147,968	62,208	↓42%
Layer2	32×32	128	525,568	247,808	↓47%
Layer3	16×16	256	2,099,712	991,232	↓47%
Stage4	8×8	512	8,393,728	3,960,448	↓47%
Head	1×1	512 → 1000	513,000	513,000	
Total			**11,689,512**	**5,783,232**	**↓~49%**

Convolutional block 1 $O(C_{in}C_{out} + C_{out}^2)$

Convolutional block 2 $O(C_{out}^2)$

Parameter reduction of almost 49%.

Low-rank block 1 $O(C_{in}T + C_{in}RN_{\text{eff}} + C_{in}C_{out})$

Low-rank block 2 $O(C_{out}T + C_{out}RN_{\text{eff}} + C_{out}^2)$

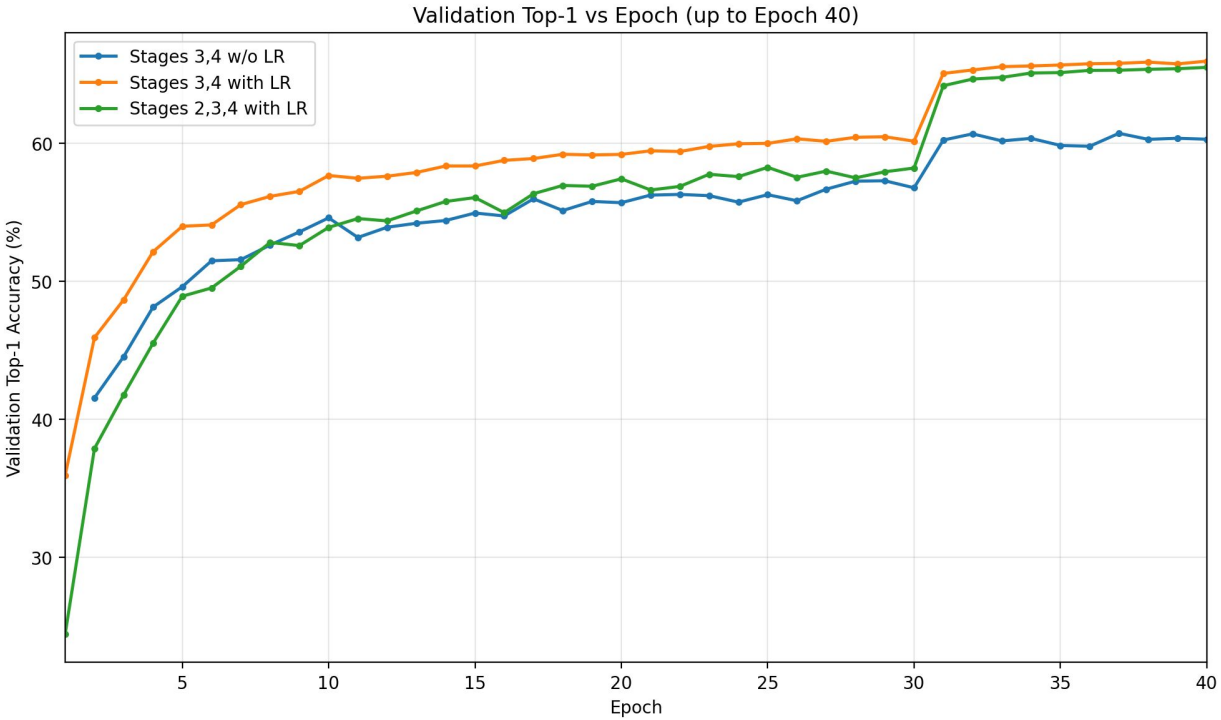
Low-rank blocks' parameter count scales with feature size and rank.

Results

- **Experiment set 1:** Ablations on using an auxiliary regularization loss for encouraging orthogonality between in the generated low-rank basis.

- **Experiment set 2**: Ablations on retraining ResNet stages with and without the proposed low-rank module.

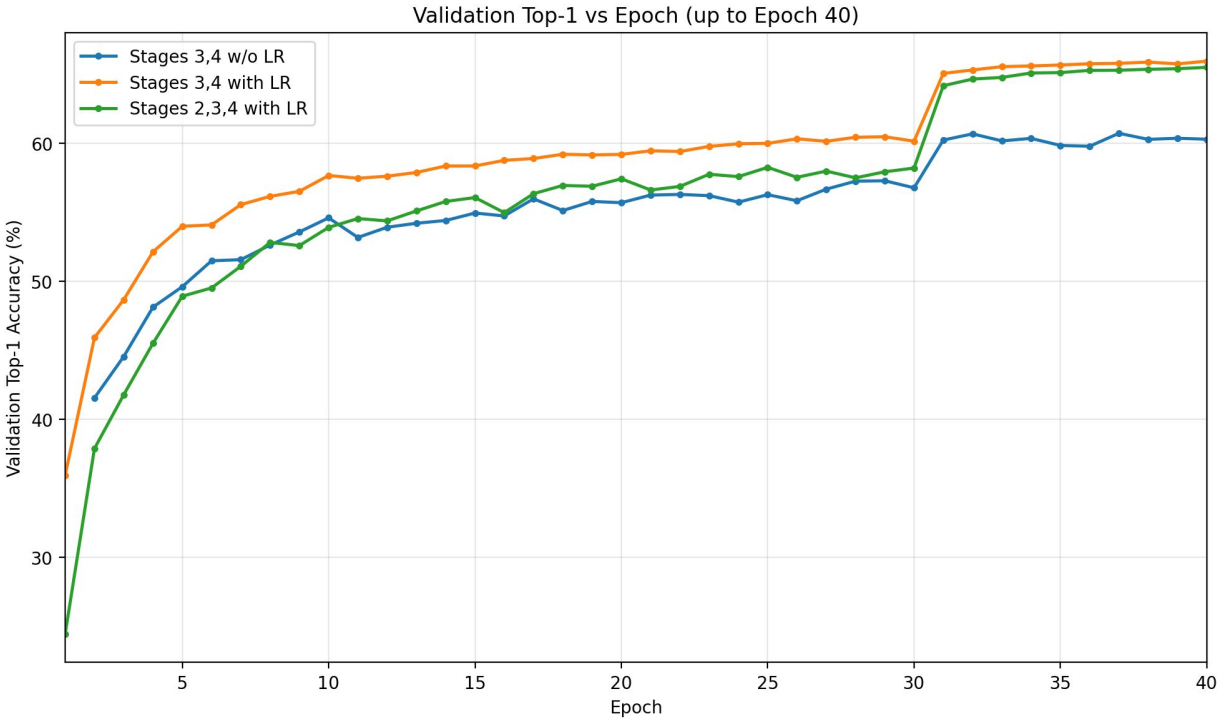
Results



Training validation curves on ImageNet 1K (40 epochs)

- The model **without the low-rank** module stagnates at **~60%** Top-1 Accuracy.
- There is a **boost of ~6.5%** when **using the low-rank** module.
- Using **LR at Stages 2,3,4**:
  - Fewer #parameters
  - Close to the best model.

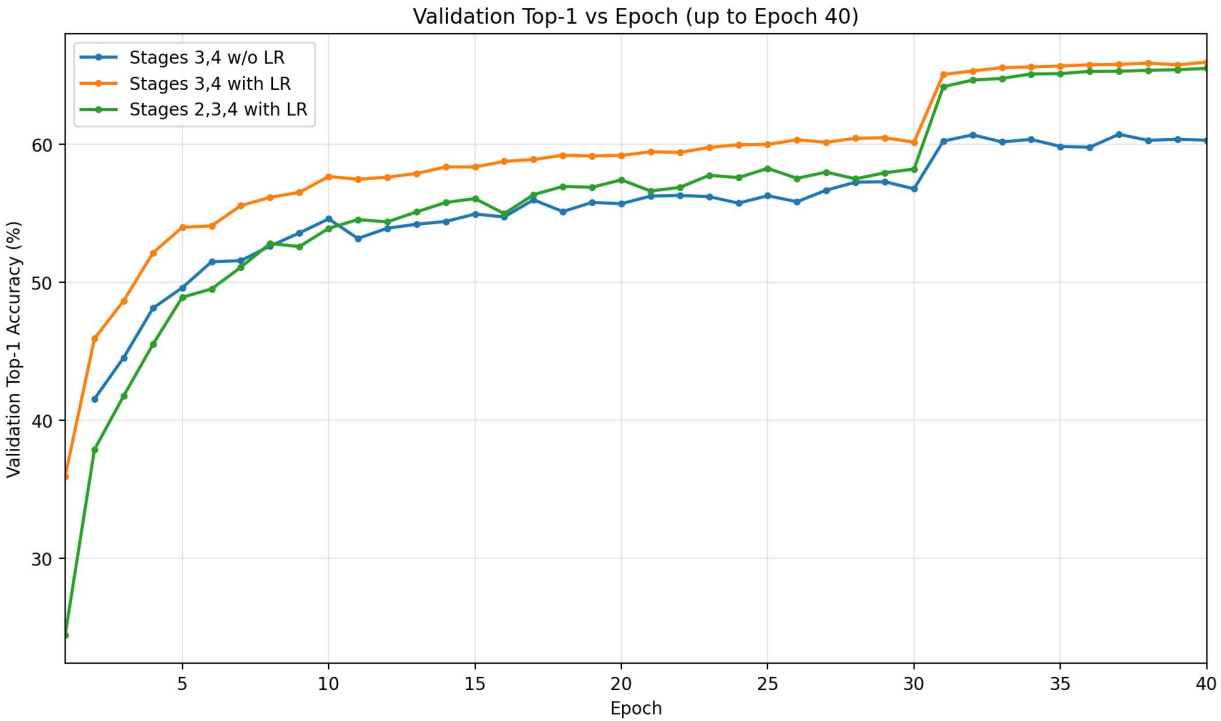
Results



Training validation curves on ImageNet 1K (40 epochs)

- The model **without the low-rank** module stagnates at **~60%** Top-1 Accuracy.
- There is a **boost of ~6.5%** when **using the low-rank** module.
- Using **LR at Stages 2,3,4**:
  - Fewer #parameters
  - Close to the best model.

Results



Training validation curves on ImageNet 1K (40 epochs)

- The model **without the low-rank** module stagnates at **~60%** Top-1 Accuracy.
- There is a **boost of ~6.5%** when **using the low-rank** module.
- Using **LR at Stages 2,3,4**:
  - Fewer #parameters
  - Close to the best model.

Results

- Replacing the **4th stage** of ResNet18 with the proposed method yields an Top-1 accuracy of 58.21% after **35 epochs**.
- Replacing the **3th and 4th** stages attains a **Top-1 accuracy of 65.69%** after 35 epochs.
- If the low-rank module is not included at the 3th and 4th stages (convolutional only) the accuracy is **60%**.
- Torch's implementation of ResNet18 reaches an accuracy of 68.85% Top-1 accuracy, while the accuracy of the proposed method is **3.16% lower** but using **~40% less parameters** than the original ResNet18.

Conclusion and next steps

- A new ResNet-based **basic feature extraction block** is introduced in this work:
  - Combines **local features** through convolutions with **global features** through MLPs.
  - Parameter count reduction: Specially for large features
- The **novel block** addresses efficient local processing and global interaction in **vision models**.
- ResNet18 is adopted as the **backbone** during the initial experimental trials.
- The module is incorporated into different stages of a pre-trained ResNet18 to r**educe the training time** (~4 days -> 3 hours [10 epochs]).
- **ImageNet 1K** is adopted as the dataset for pre-training.
- After **35 epochs** the proposed method attains **65.59% Top-1 accuracy** on ImageNet 1K. The parameter count is reduced by **~40%** than the original ResNet18.

- **After validating the proposed block** -> Incorporate the block into other backbones (ResNet50,MobileNet,CNN-ViT hybrids).
- Use the pre-trained backbones on **downstream tasks**: semantic segmentation, pose estimation.

Summary (*The content will be included in the midterm exam*)

- **Methodology:** Develop a novel ResNet-based block that fuses global low-rank features with local convolutional features to build a compact vision backbone.
- **Steps:**
  - Integrate the proposed block into **ResNet-18** during early experiments to refine the design.
  - Replace the core blocks of the backbone with the proposed module.
  - **Pre-train the modified backbones on ImageNet-1K.**
  - Evaluate the learned representations on **downstream tasks**:
    - Semantic segmentation
    - Pose estimation
- **Results:** A compact yet effective vision backbone combining global and local feature modeling.