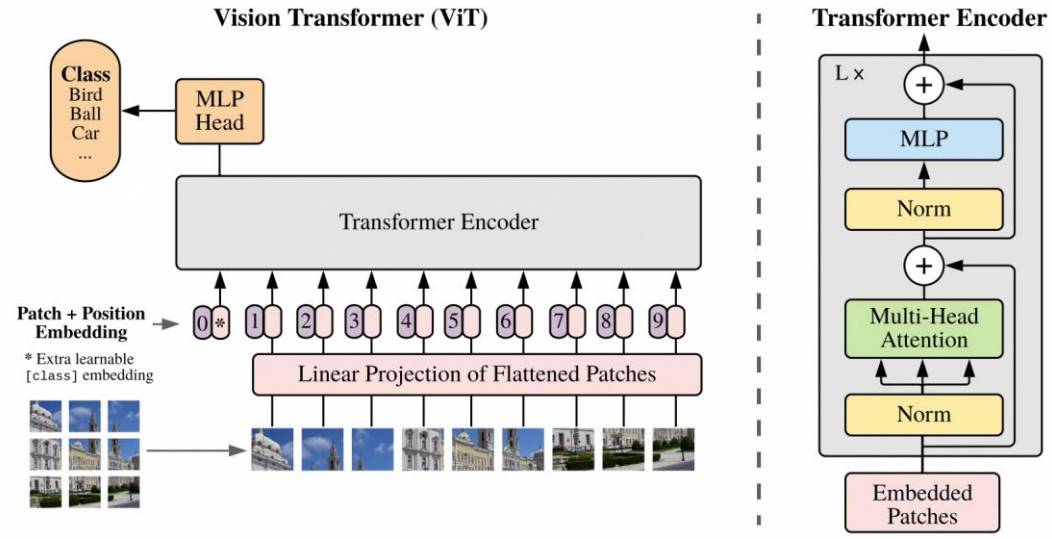
# Boosting Vanilla Lightweight Vision Transformers via Re-parameterization

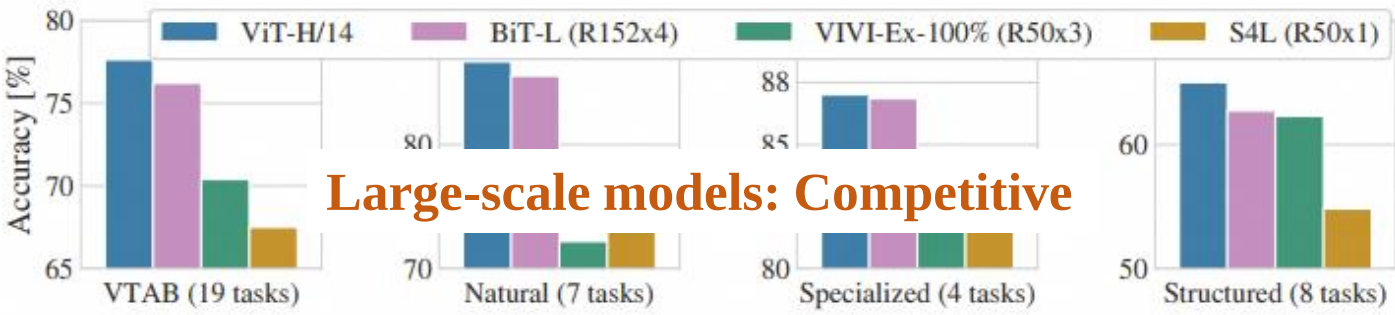
Alibaba Cloud[1], Alibaba Group[2],
University of Science and Technology of China[3],
East China Normal University[4],

Zhentao Tan[1,3], Xiaodan Lin[4,2], Yue Wu[1], Qi Chu[3], Le Lu[2], Nenghai Yu[3], Jieping Ye[1]

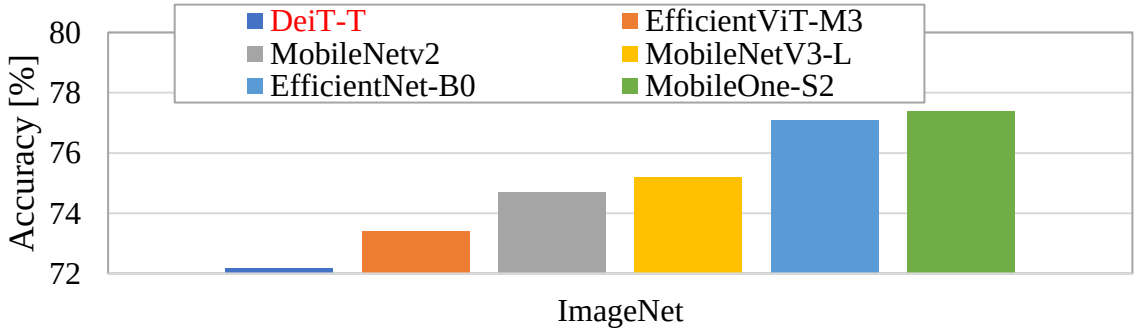
# Background: ViTs

**Performance difference between large-scale and lightweight vanilla vision Transformers.**




**Large-scale models: Competitive**




**Lightweight models: Unsatisfactory**

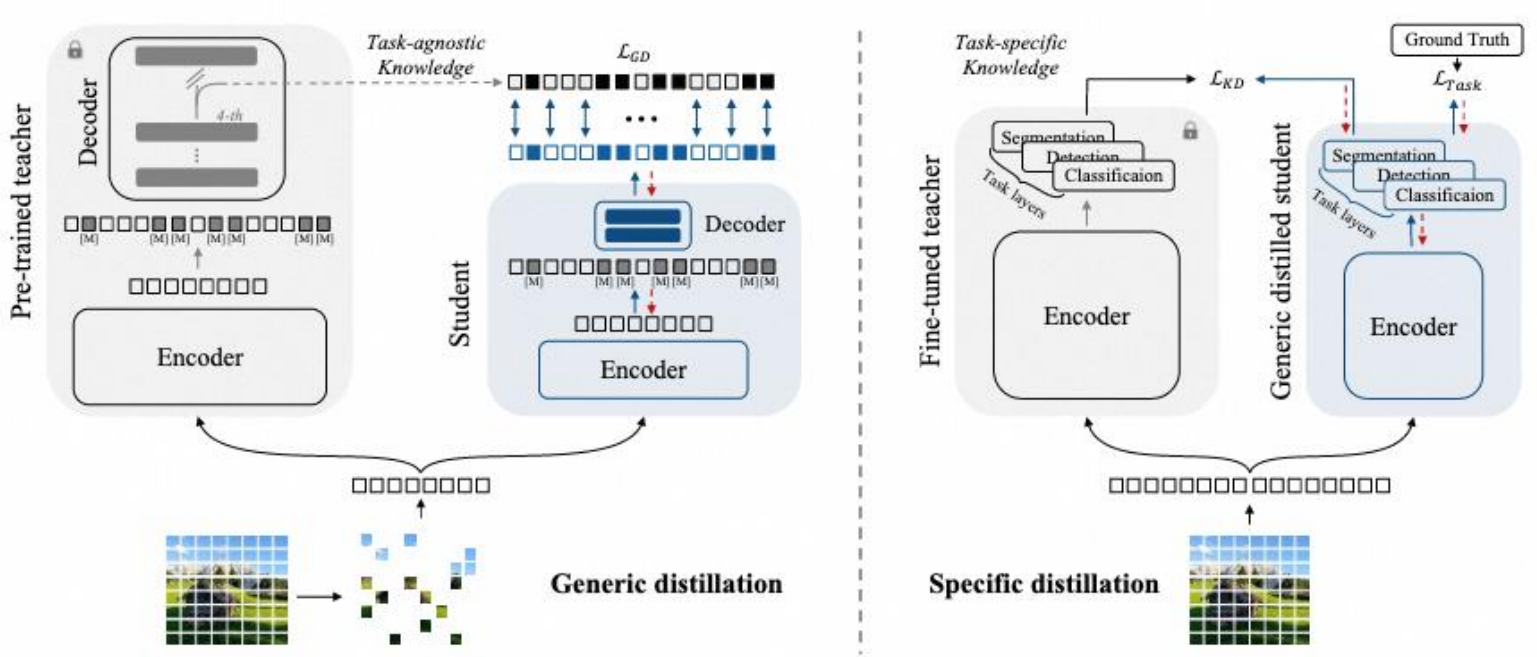


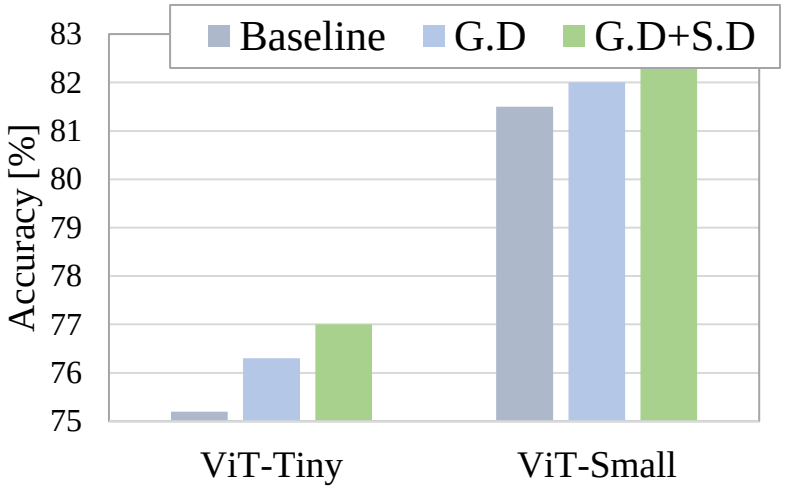
An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, ICLR, 2021

# **Background:** Distillation

**How can we improve the performance of vanilla lightweight ViTs without any inference pipeline changes?**

**Related works: Knowledge Distillation**



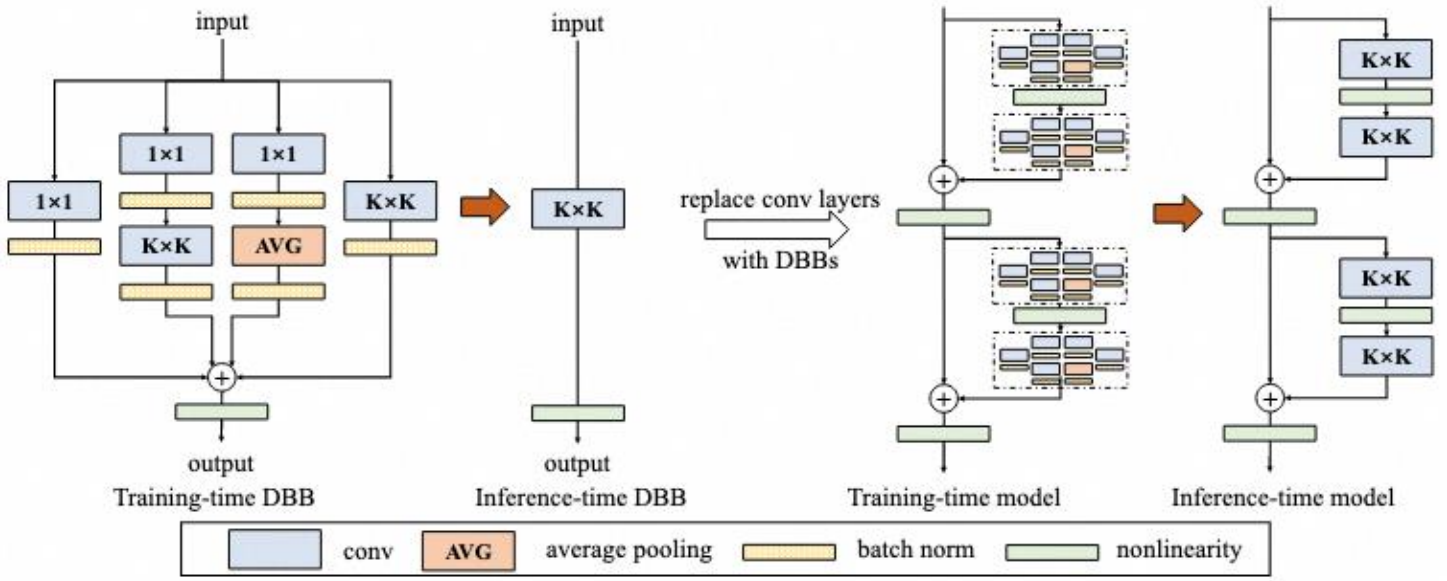

**Costs of pre-trained teachers (especially for S.D)**



Generic-to-Specific Distillation of Masked Autoencoders, CVPR, 2023

# **Background:** Re-parameterization

**Structure re-parameterization in CNNs**




- **Training:**
  - **multi-branch**
  - **over parameterization**
- **Inference:**
  - **single KxK convolution**

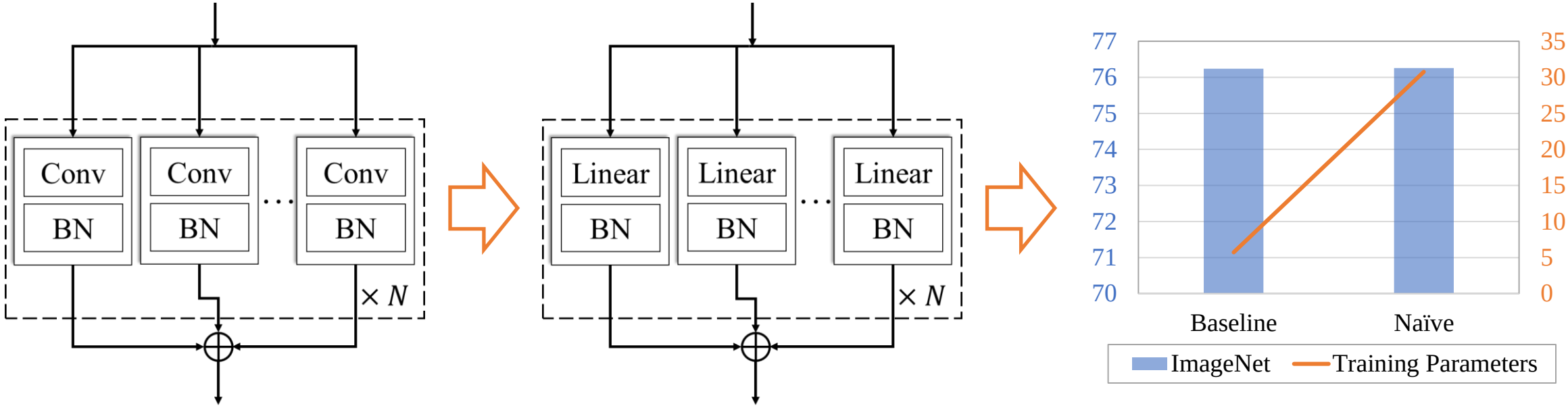
**Can we apply re-parameterization to vanilla ViTs?**



Diverse Branch Block: Building a Convolution as an Inception-like Unit, CVPR, 2021

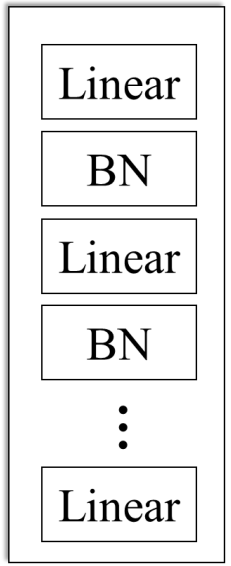
# **Method:** Single Experiments

**Single application: directly replacing convolution layers into linear layers.**



# **Method:** Linear Ensemble

**Stacking linear layers with batch normalization in-between them.**




- **Effectiveness: similar to MLP which is appropriate to transformers and plays an important role to represent rich intra-token information.**
- **Rationality: batch normalization is still can be used in-between layers while keeping original layer normalization unchanged.**
- **Operability: it can be fused to a single linear layer after training.**

**1. Merging linear and batch normalization**

$$\boldsymbol{W}'_{i,:} = \frac{\gamma_i}{\sigma_i}\boldsymbol{W}_{i,:}, \quad b'_i = \frac{(b_i - \mu_i)\gamma_i}{\sigma_i} + \beta_i,$$

**2. Merging two adjacent linear layers**

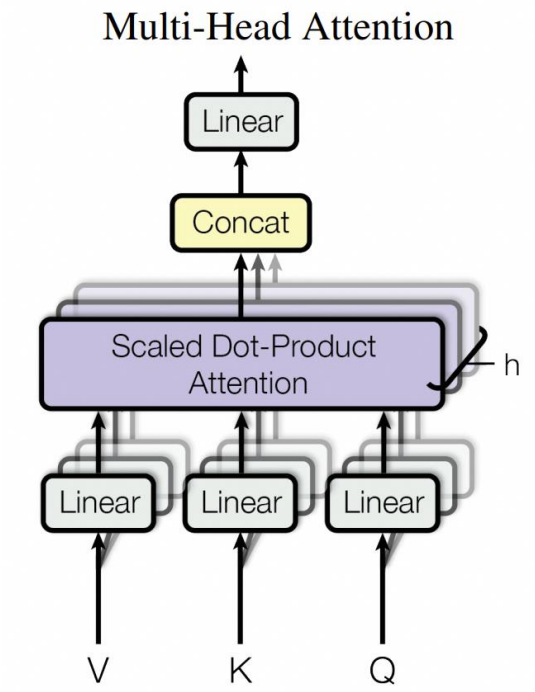
$$W'_{i,j}(l+1,l) = \sum_{k=1}^{k=C_l} W^{l+1}_{i,k} W^{l}_{k,j},$$

$$b'_i(l+1,l) = \sum_{k=1}^{C_l} b^l_k W^{l+1}_{i,k} + b^{l+1}_i,$$

# **Method:** Distribution Rectification

**Multi-branch re-parameterization will change the feature distribution.**

**Attention mechanism in vision transformers is sensitive to this distribution changes.**




**Normal Attention Operation:**

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = softmax(\frac{A}{\sqrt{C_k}})\boldsymbol{V}, \quad \boldsymbol{A} = \boldsymbol{Q}\boldsymbol{K}^T.$$

$$A_{i,j} = \sum_{k=1}^{k=C_k} Q_{i,k}K_{k,j}$$ Variance scales to $C_k$

**Re-parameterization Attention Operation:**

$$A'_{i,j} = \sum_{k=1}^{k=C_k}(\sum_{n=1}^{N_Q} Q_{i,k}^n)(\sum_{m=1}^{N_K} K_{k,j}^m) = \sum_{k=1}^{k=C_k}\sum_{n=1}^{N_Q}\sum_{m=1}^{N_K} Q_{i,k}^n K_{k,j}^m.$$

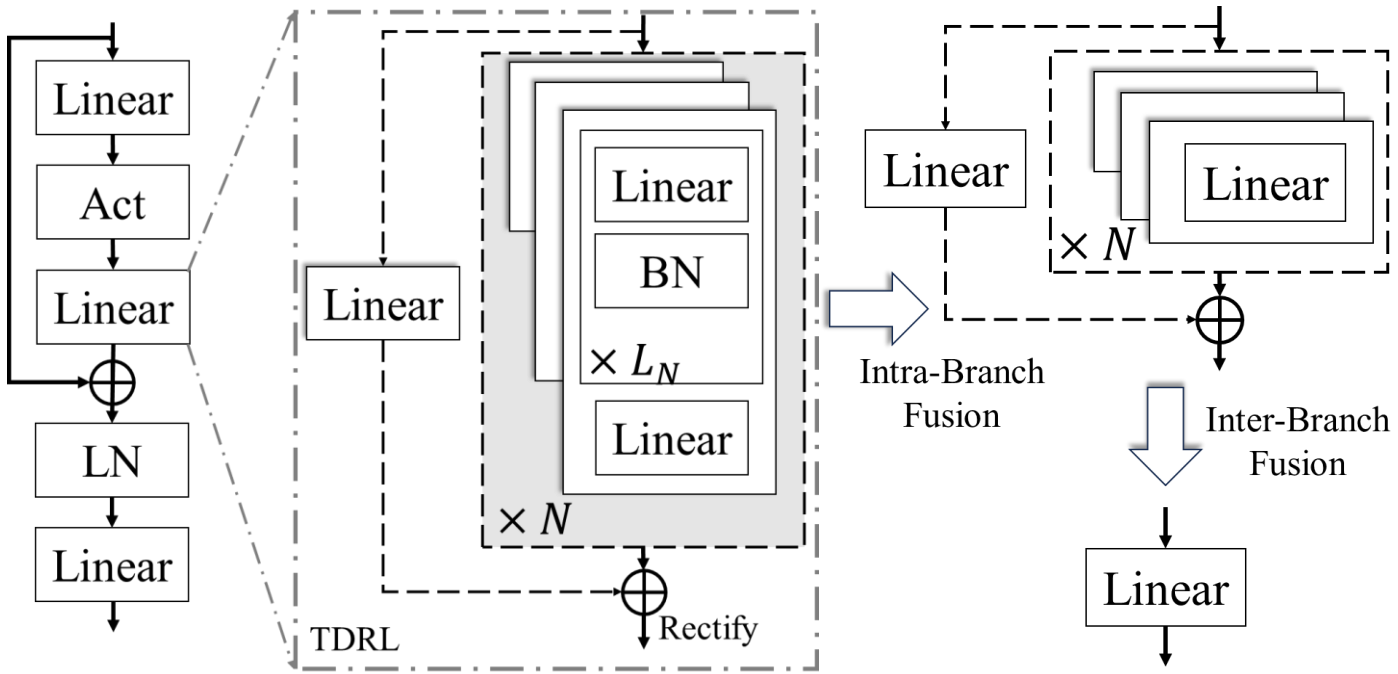
Variance scales to $C_k N_Q N_K$

$$R(x) = \begin{cases} BN(x), QKV \\ \frac{x}{\sqrt{C_k N_Q N_K}}, others \end{cases}$$

Attention is all you need, NeurIPs, 2017

# Method: Structure

**TDRL: Pyramid-wise Multi-branch**

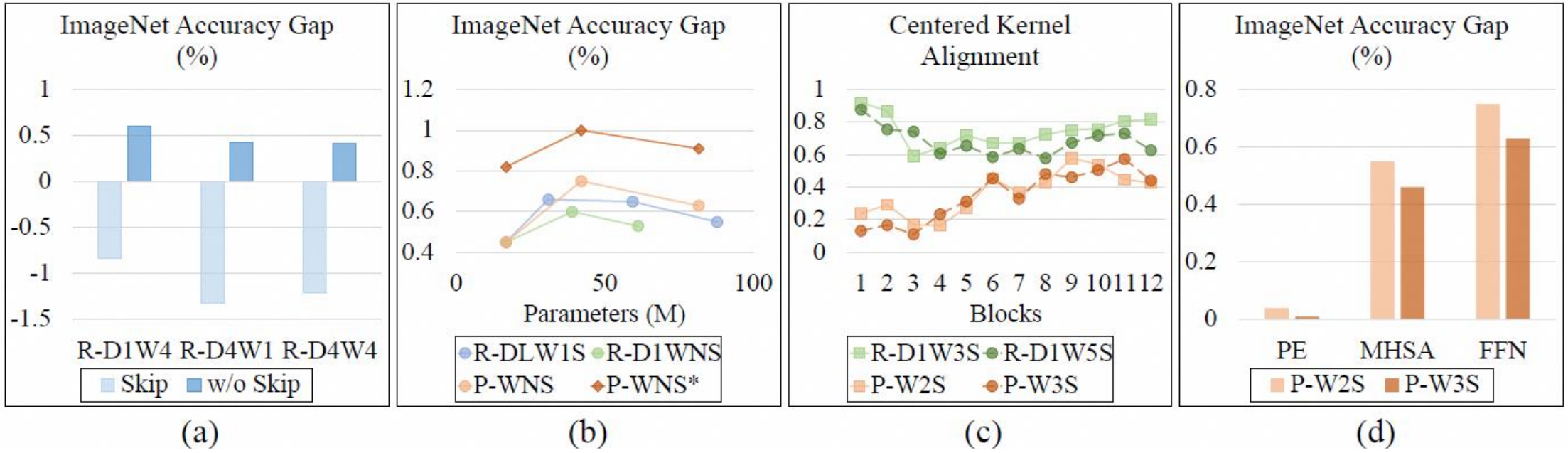



$$\boldsymbol{Y} = Rectify(Linear(\boldsymbol{X}) + \sum_{n=1}^{N} f_{n,L_n}(\boldsymbol{X})), \quad L_n = n,$$

- **Design:**
  - **Skip-branch**
  - **Rep-branch**
    - **Length: representation ability**
    - **Width: representation diversity**
- **Application:**
  - **Arbitrary linear layers**

# Experiments: Ablations

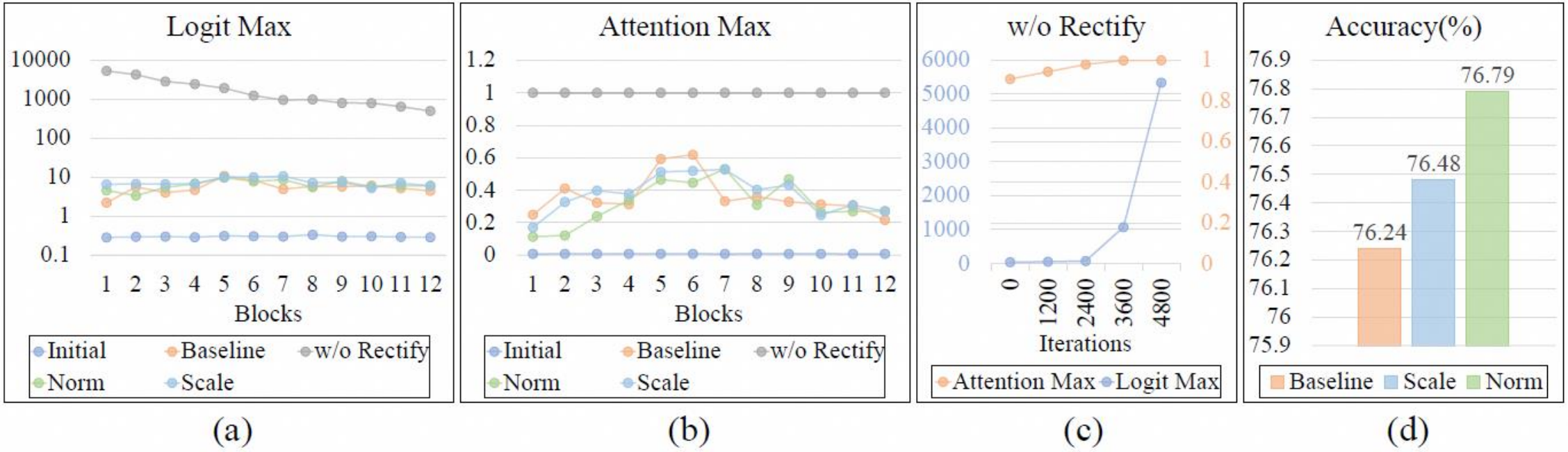
➢ **Structure Designs**




- **Skip connection plays an important role.**
- **More branches do not always lead to better performance.**
- **Pyramid-wise design leads to diversity representation between different branches.**
- **Applying TDRL to MHSA and FFN results in much more improvements.**

# Experiments: Ablations

- **Attention Distribution Rectification**




- **Without rectification, the maximum value in attention is prone to extreme values.**
- **Normalization is better than scaling as a rectification function.**

# Experiments: Classification

| Method | Network | Teacher | *FT* | *P* (M) | Acc(%) |
|---|---|---|---|---|---|
| Without Pre-training | | | | | |
| MobileNet-v3 (Howard et al., 2019) | CNNs | N/A | 600 | 6 | 75.2 |
| ConvNeXt-V1-F (Liu et al., 2022b) | CNNs | N/A | 600 | 5 | 77.5 |
| VanillaNet-5 (Chen et al., 2023) | CNNs | N/A | 300 | 15.5 | 72.5 |
| MobileViT-S (Mehta & Rastegari, 2021) | Hybrid | N/A | 300 | 6 | 78.3 |
| EfficientViT-M3 (Liu et al., 2023) | Hybrid | N/A | 300 | 7 | 73.4 |
| DeiT-Ti (Touvron et al., 2021) | ViTs | N/A | 300 | 5 | 72.2 |
| Manifold-Ti (Jia et al., 2021) | ViTs | CaiT-S24 | - | 6 | 75.1† |
| MKD-Ti (Liu et al., 2022a) | ViTs | CaiT-S24 | 300 | 6 | 76.4† |
| DeiT-Ti (Touvron et al., 2021) | ViTs | RegNetY | 300 | 6 | 74.5† |
| SSTA-Ti (Wu et al., 2022a) | ViTs | DeiT-S | 300 | 6 | 75.2† |
| ImageNet Pre-training | | | | | |
| DMAE-Ti (Bai et al., 2023) | ViTs | ViT-B | 100 | 6 | 74.9 |
| MAE-Lite (Wang et al., 2023) | ViTs | N/A | 100 | 6 | 76.2 |
| MAE-Ti (He et al., 2022) | ViTs | N/A | 200 | 6 | 75.2 |
| TinyMIM-Ti (Ren et al., 2023) | ViTs | TinyMIM-S | 200 | 6 | 75.8 |
| G2SD-Ti w/o S.D (Huang et al., 2023) | ViTs | ViT-B | 200 | 6 | 76.3 |
| G2SD-Ti (Huang et al., 2023) | ViTs | ViT-B | 200 | 6 | 77.0† |
| TDRL (ours) | ViTs | ViT-B | 200 | 6 | **78.3/78.6†** |
| MAE-Lite (Wang et al., 2023) | ViTs | N/A | 300 | 6 | 78.0 |
| D-MAE-Lite (Wang et al., 2023) | ViTs | ViT-B | 300 | 6 | 78.4 |
| TDRL (ours) | ViTs | ViT-B | 300 | 6 | **78.7/79.1†** |

**†means performing distillation during fine-tuning.**

- **TDRL achieves the best image classification accuracy under various epoch settings.**
- **When performing distillation during fine-tuning, the performance of TDRL can be further improved to 79.1%.**

# **Experiments:** Dense Prediction

| Method | #Params (M) Seg/Det | Segmentation mIoU | Detection $AP^{bbox}$ | Detection $AP^{mask}$ |
|---|---|---|---|---|
| Swin-T (Liu et al., 2021) | 59.9/47.8 | 44.5 | 46.0‡ | 41.6‡ |
| ConvNeXt-T (Liu et al., 2022b) | 60.0/48.1 | 46.0 | 46.2‡ | 41.7‡ |
| DINO-S (Caron et al., 2021) | 42.0/44.5 | 44.0 | 49.1 | 43.3 |
| iBOT-S (Zhou et al., 2021) | 42.0/44.5 | 45.4 | 49.7 | 44.0 |
| MAE-S (He et al., 2022) | 42.0/44.5 | 41.1/44.9† | 45.3 | 40.8 |
| MAE-Ti (He et al., 2022) | 11.0/27.7 | 36.9/42.0† | 37.9/43.5† | 34.9/39.0† |
| MAE-Lite (Wang et al., 2023) | 11.0/27.7 | 37.6 | 39.9* | 35.4* |
| D-MAE-Lite (Wang et al., 2023) | 11.0/27.7 | 42.0 | 42.3* | 37.4* |
| G2SD-Ti (Huang et al., 2023) | 11.0/27.7 | 41.4/44.5† | 44.0/46.3† | 39.6/41.3† |
| TDRL (ours) | 11.0/27.7 | **42.5/45.2**† | **46.5/47.4**† | **41.5/42.1**† |

- **TDRL can also benefit dense prediction tasks such as semantic segmentation and object detection.**

# Experiments: Generality

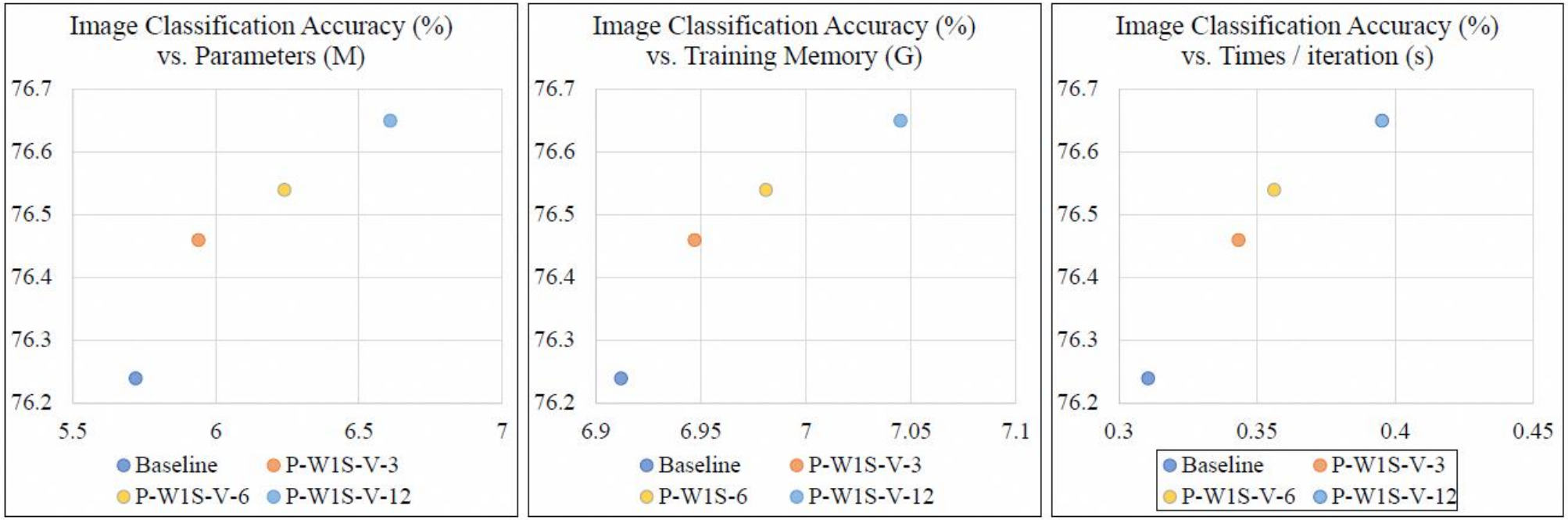
| TDRL | Classification Accuracy (%) ↑ | | | | Image Generation FID ↓ |
|---|---|---|---|---|---|
| | ViT-Small | Swin-Ti | Mobileone-S0 | VanillaNet-5 | DDPM |
| × | 80.8 | 76.2 | 71.3 | 71.1 | 10.4 |
| √ | 81.3 (+0.5) | 78.2 (+2.0) | 75.1 (+3.8) | 71.5 (+0.4) | 9.2 (+1.2) |

- **TDRL can be used in various networks and tasks:**
  - **Larger model (e.g., ViT-Small, Swin-Tiny)**
  - **CNN model (e.g., VanillaNet)**
  - **Hybrid model (e.g., MobileOne)**
  - **DDPM**

# Experiments: Efficiency

| Method | Pre-train time (hours) | Pre-train epoch | Fine-tune epoch | Accuracy (%) |
|---|---|---|---|---|
| Baseline (G2SD) | 32.33 | 300 | 100 | 76.24 |
| TDRL (ours) | 32.02 | 220 | 100 | 76.73 (+0.59) |

- **Under similar training costs, TDRL can still improve the performance of ViT-Tiny.**



# Summary

- **We propose a novel re-parameterization method, namely TDRL, to improve the performance of vanilla lightweight Vision Transformers.**
- **To improve the representation ability, we design a linear ensemble way and a pyramid-wise multi-branch structure.**
- **For stable training, we analyze the feature distribution change issues and propose a simple rectification method.**
- **Experiments on various tasks and backbone have demonstrated the effectiveness of our TDRL.**



**More details can be found in** https://openreview.net/pdf?id=3rmpixOjPS

ICLR

# Thanks



https://openreview.net/pdf?id=3rmpixOjPS