Foundation Models on a Budget: Approximating Blocks in Large Vision Models

Large vision foundation models (e.g., ViTs, DiNO, DEiT) achieve state-of-the-art performance but are extremely resource-hungry, limiting accessibility and sustainability. While compression methods like pruning, distillation, or early exiting exist, they typically require retraining or fine-tuning. This paper introduces **Transformer Block Approximation (TBA)**, a novel, training-free approach to reduce model size by exploiting **intra-network similarities**. Prior work mainly studied inter-model representation similarities (e.g., for model stitching), but this work shows that within a single model, multiple transformer blocks often produce **redundant representations**.

Our method works in two-stages by first identifying detect block pairs whose outputs are highly similar using a simple metric yet effective metric (i.e., MSE), then it **r**eplaces intermediate blocks with a simple closed-form **linear transformation estimated on a small subset** that maps the output of an earlier block directly to the later one. This removes parameters and computations without retraining or fine-tuning. See Figure 1 for a visualization of the method.

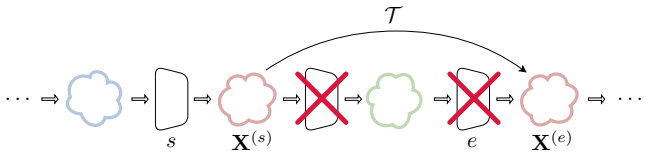

Figure 1: **Framework Description**. Given two latent spaces $\mathbf{X}^{(s)}$ and $\mathbf{X}^{(e)}$ representing the output of two blocks s and e for a random subset of n data points from the training set, we define a transformation matrix $\mathcal{T}$ such that: $\mathbf{X}^{(e)} \approx \mathcal{T}(\mathbf{X}^{(s)})$.

Results in Table 1 demonstrate that TBA reduces parameter count while largely preserving representation fidelity. On image classification benchmarks (CIFAR-10/100, ImageNet-1k), TBA compresses different models with **minimal accuracy drop**, sometimes even **improving performance**. Additionally, learned transformations generalize across datasets, showing potential for privacy-sensitive domains where retraining is impossible.

Table 1: **Image Classification Performance Across Architectures.** Classification accuracy scores for `DiNO-B` and `DEiT-S` using multiple datasets, and 3 seeds. The "Approx." column specifies the blocks used for approximation, where the first value represents the block whose output is used to approximate the second block's output. The "Params." column shows the number of parameters removed by the approximation compared to the original model.

			Accuracy ↑		
	Approx.	Params.	`CIFAR-10`	`CIFAR-100F`	`ImageNet1k`
`DiNO-B`	1 →5	`-26.5M`	86.53 ± 0.21 (-11.28%)	62.11 ± 0.86 (-28.62%)	30.74 ± 0.45 (-58.57%)
	2 →5	`-19.5M`	92.03 ± 0.10 (-5.87%)	70.04 ± 0.72 (-19.50%)	53.23 ± 0.09 (-28.26%)
	7 →10	`-19.5M`	93.41 ± 0.34 (-4.46%)	74.81 ± 1.32 (-14.02%)	47.58 ± 0.77 (-35.88%)
	2 →4	`-13M`	96.31 ± 0.18 (-1.49%)	81.25 ± 0.57 (-6.62%)	68.49 ± 0.12 (-7.70%)
	9 →11	`-13M`	87.01 ± 0.30 (-10.76%)	72.65 ± 1.86 (-14.36%)	46.46 ± 0.23 (-27.74%)
	1 →2, 4 →5	`-13M`	96.33 ± 0.18 (-1.47%)	81.69 ± 0.07 (-6.11%)	68.42 ± 0.20 (-7.79%)
	0 →1	`-6.5M`	**97.72** ± 0.05 (**-0.05**%)	**86.13** ± 0.14 (**-1.01**%)	**73.46** ± 0.21 (**-1.00**%)
	3 →4	`-6.5M`	97.40 ± 0.04 (-0.38%)	84.59 ± 0.22 (-2.78%)	73.00 ± 0.52 (-1.62%)
	9 →10	`-6.5M`	97.01 ± 0.16 (-0.78%)	84.17 ± 0.28 (-3.26%)	66.71 ± 0.17 (-10.09%)
	10 →11	`-6.5M`	96.78 ± 0.22 (-1.01%)	83.33 ± 0.52 (-4.23%)	63.94 ± 0.63 (-13.83%)
	original	`86.58M`	97.77 ± 0.24	87.01 ± 0.30	74.20 ± 0.68
`DEiT-S`	1 →5	`-6.51M`	78.35 ± 0.17 (-13.55%)	50.57 ± 0.29 (-28.90%)	43.70 ± 0.27 (-40.88%)
	2 →5	`-4.88M`	85.73 ± 0.31 (-5.41%)	60.55 ± 0.16 (-14.87%)	62.04 ± 0.21 (-16.05%)
	7 →10	`-4.88M`	89.17 ± 0.04 (-1.61%)	69.15 ± 0.33 (-2.78%)	57.48 ± 0.06 (-22.24%)
	2 →4	`-3.26M`	88.95 ± 0.05 (-1.85%)	66.60 ± 0.50 (-6.37%)	70.00 ± 0.32 (-5.32%)
	9 →11	`-3.26M`	90.90 ± 0.12 (+0.30%)	71.92 ± 0.17 (+1.11%)	69.95 ± 0.24 (-5.39%)
	1 →2, 4 →5	`-3.26M`	85.43 ± 0.25 (-5.74%)	61.66 ± 0.13 (-13.31%)	66.04 ± 0.13 (-10.68%)
	0 →1	`-1.63M`	85.00 ± 0.27 (-6.21%)	61.95 ± 0.39 (-12.91%)	62.55 ± 0.12 (-15.38%)
	3 →4	`-1.63M`	90.50 ± 0.10 (-0.14%)	70.25 ± 0.20 (-1.24%)	73.03 ± 0.10 (-1.22%)
	9 →10	`-1.63M`	90.90 ± 0.20 (+0.30%)	71.74 ± 0.09 (+0.86%)	72.34 ± 0.16 (-2.15%)
	10 →11	`-1.63M`	**91.07** ± 0.18 (**+0.49**%)	**71.95** ± 0.17 (**+1.15**%)	**73.97** ± 0.17 (**+0.05**%)
	original	`21.82M`	90.63 ± 0.22	71.13 ± 0.26	73.93 ± 0.14

Conclusion

TBA is a practical, training-free method for simplifying foundation vision models by leveraging block-wise similarities. It improves computational accessibility and sustainability, offering a promising direction for efficient foundation models.