DTL: Disentangled Transfer Learning for Visual Recognition

Minghao Fu, Ke Zhu, Jianxin Wu*
National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
fumh@lamda.nju.edu.cn, zhuk@lamda.nju.edu.cn, wujx2001@gmail.com

The 38th Annual AAAI Conference on Artificial Intelligence
FEBRUARY 20-27, 2024 | VANCOUVER, CANADA
VANCOUVER CONVENTION CENTRE – WEST BUILDING

1. Introduction & Motivation

Current parameter-efficient transfer learning (PETL) methods are facing the dilemma that during training the GPU memory footprint is *not* effectively reduced as trainable parameters. PETL will likely fail, if the full fine-tuning encounters the out-of-GPU-memory issue.

Why?

➢ Trainable parameters from these methods are generally **entangled** with the backbone.

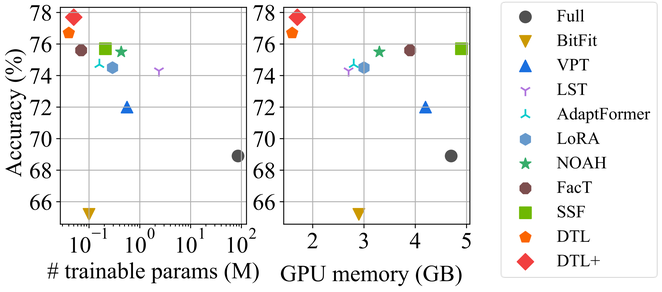
$$\frac{\partial L}{\partial o_{i+1}} = \frac{\partial L}{\partial o_{i+2}} \frac{\partial o_{i+2}}{\partial z_{i+2}} \frac{\partial z_{i+2}}{\partial o_{i+1}} = \frac{\partial L}{\partial o_{i+2}} \sigma'_{i+1} W_{i+1}$$

- All corresponding $\{\sigma'_i\}$ in the chain rule have to be cached during fine-tuning, which dominates the GPU memory usage.

Our Solution

➢ Disentangled Transfer Learning (DTL) which disentangles the trainable parameters from the backbone using a lightweight Compact Side Network (CSN) with:

- A few low-rank linear mappings to extract information.
- Adding information back to backbone for feature adaptation.



2. Framework: DTL & DTL+

- DTL: Simplicity Matters
  - Low rank linear transformation:

$$w_i = a_i c_i \in \mathbb{R}^{d \times d}, \quad a_i \in \mathbb{R}^{d \times d'}, \quad c_i \in \mathbb{R}^{d' \times d}, \quad d' \ll d$$

  - Progressively information gathering:

$$h_{i+1} = h_i + z_i w_i$$
$$z_{i+1} = b_i(z_i)$$

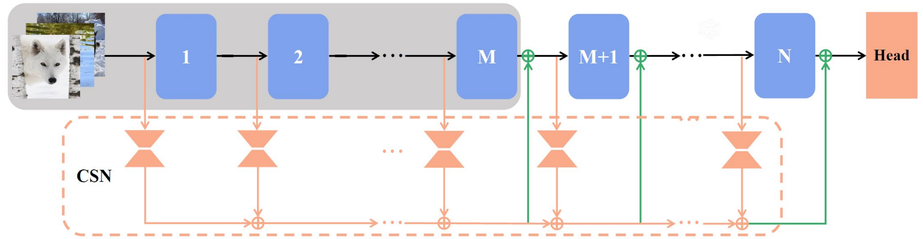
  - Post feature adaptation:

$$z'_{i+1} = \begin{cases} z_{i+1} + \theta(h_{i+1}) & \text{if } i \geq M \\ z_{i+1} & \text{otherwise} \end{cases}$$

- DTL+: Effectiveness Matters
  - append an additional global depthwise separable convolution (DWConv) layer g.

$$z'_{i+1} = \begin{cases} z_{i+1} + g(\theta(h_{i+1})) & \text{if } i \geq M \\ z_{i+1} & \text{otherwise} \end{cases}$$

  - g is shared across all CSN layers.



✓ Disentangled
- $\{\sigma'_i\}$ is drastically reduced.
- *Compatible* with other backbones.
- Multi-task inference: 45% faster

✓ Simple
- Simplest structure.

Advantages

Method	Source	#unit
LoRA	low-rank matrices in W_q, W_v	24
NOAH	low-rank matrices, bottlenecks, prompts	36
FacT	decomposed tensors	144
SSF	pairs of γ, β	148
DTL	low-rank matrices	12
DTL+	low-rank matrices, DWConv	13

✓ Effective
- SOTA downstream accuracy.
- Least trainable parameters.
- Least GPU memory footprint.

3. Experiments

SOTA Comparison

	#param (M)	GPU mem (GB)	Natural: Cifar100	Caltech101	DTD	Flower102	Pets	SVHN	Sun397	Specialized: Camelyon	EuroSAT	Resisc45	Retinopathy	Structured: Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	Average
Traditional Fine-Tuning																						
Full	85.8	4.7	68.9	87.7	64.3	97.2	86.9	87.4	38.8	79.7	95.7	84.2	73.9	56.3	58.6	41.7	65.5	57.5	46.7	25.7	29.1	68.9
Linear	0	0.6	64.4	85.0	63.2	97.0	86.3	36.6	51.0	78.5	87.5	68.5	74.0	34.3	30.6	33.2	55.4	12.5	20.0	9.6	19.2	57.6
PETL methods																						
BitFit	0.10	2.9	72.8	87.0	59.2	97.5	85.3	59.9	51.4	78.7	91.6	72.9	69.8	61.5	55.6	32.4	55.9	66.6	40.0	15.7	25.1	65.2
VPT	0.56	4.2	78.8	90.8	65.8	98.0	88.3	78.1	49.6	81.8	96.1	83.4	68.4	68.5	60.0	46.5	72.8	73.6	47.9	32.9	37.8	72.0
LST	2.38	2.7	59.5	91.5	69.0	99.2	89.9	79.5	54.6	86.9	95.9	85.3	74.1	81.8	61.8	52.2	81.0	71.7	49.5	33.7	45.2	74.3
LoRA	0.29	3.0	67.1	91.4	69.4	98.8	90.4	85.3	54.0	84.9	95.3	84.4	73.6	**82.9**	**69.2**	49.8	78.5	75.7	47.1	31.0	44.0	74.5
AdaptFormer	0.16	2.8	70.8	91.2	70.5	99.1	90.9	86.6	54.8	83.0	95.8	84.4	**76.3**	81.9	64.3	49.3	80.3	76.3	45.7	31.7	41.1	74.7
NOAH	0.43	3.3	69.6	92.7	70.2	99.1	90.4	86.1	53.7	84.4	95.4	83.9	75.8	82.8	68.9	49.9	81.7	81.8	48.3	32.8	44.2	75.5
FacT	0.07	3.9	70.6	90.6	70.8	99.1	90.7	88.6	54.1	84.8	96.2	84.5	75.7	82.6	68.2	49.8	80.7	80.8	47.4	33.2	43.0	75.6
SSF	0.21	4.9	69.0	92.6	**75.1**	99.4	91.8	90.2	52.9	87.4	95.9	87.4	75.5	75.9	62.3	**53.3**	80.6	77.3	54.9	29.5	37.9	75.7
DTL	**0.04**	**1.6**	69.6	94.8	71.3	99.3	91.3	83.3	56.2	87.1	96.2	86.1	75.0	82.8	64.2	48.8	81.9	93.9	53.9	34.2	47.1	76.7
DTL+	0.05	1.7	70.4	**95.1**	71.5	**99.4**	**91.8**	87.5	56.8	87.7	96.6	86.9	74.7	81.6	65.1	51.3	**82.3**	**97.2**	**54.9**	**36.0**	49.3	77.7
DTL+*	0.05	3.1	**74.1**	94.8	71.8	99.4	91.7	**90.4**	**57.2**	**87.9**	**96.7**	**87.5**	74.8	81.9	64.7	51.5	81.9	93.9	54.0	35.6	**50.3**	**77.9**

- Saving of 43% trainable parameters, 41% GPU memory compared to previous SOTA.
- Different variants are consistently effective in terms of recognition accuracy.

Different Backbone

Method	#p	#m	Nat.	Spe.	Str.	Avg.
Full	86.7	6.1	79.2	86.2	59.7	75.0
Linear	0	0.9	73.5	80.8	33.5	62.6
BitFit	0.20	3.7	74.2	80.1	42.4	65.6
VPT	0.16	4.6	76.8	84.5	53.4	71.6
FacT	0.14	5.6	83.1	86.9	62.1	77.4
DTL	**0.09**	**1.5**	82.4	**87.0**	64.2	77.9
DTL+	0.13	1.6	82.4	86.8	**66.0**	78.4
DTL+*	0.14	4.0	**83.2**	87.0	65.7	**78.6**

- Generalizable to Hierarchical Transformer.

Inference Latency

Methods	Throughput (imgs/sec)		
	bs=1	bs=4	bs=16
Full	161	636	952
LST	71	279	729
NOAH	79	306	798
AdaptFormer	108	436	876
DTL+	120	469	877
DTL	131	528	892

- More speed up compared to PETL methods.

4. Contributions & Conclusions

✓ We analyze limitations of existing PETL methods from the perspective of GPU memory usage, which has a critical influence on the feasibility of fine-tuning.

✓ We propose DTL, a disentangled and simple framework for efficiently fine-tuning large-scale pre-trained models with significantly less trainable parameters and GPU memory usage.

✓ Extensive experiments are conducted to verify the effectiveness of DTL, which outperforms existing methods with a large margin.

First Author | Second Author | Professor

LAMDA Learning And Mining from DatA