JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Trainable Projected Gradient Method for Robust Fine-tuning

Junjiao Tian[1], Xiaoliang Dai[2], Chih-Yao Ma[2], Zecheng He[2], Yen-Cheng Liu[1], Zsolt Kira[1]

[1]Georgia Institute of Technology, [2]Meta

Paper ID: 4761

Poster session: TUE-PM-355

Meta

Georgia Tech

Transfer Learning using Pretrained Models

- Pre-trained models have become more powerful
- Pre-trained models improve model robustness and uncertainty [1]
- Fine-tuning does not generally preserve robustness.
  - A large learning rate can easily destroy the generalization capability in pre-trained models [2].

Robust Fine-tuning: maintain *the generalization capability* of the pre-trained model while fitting to a target task.

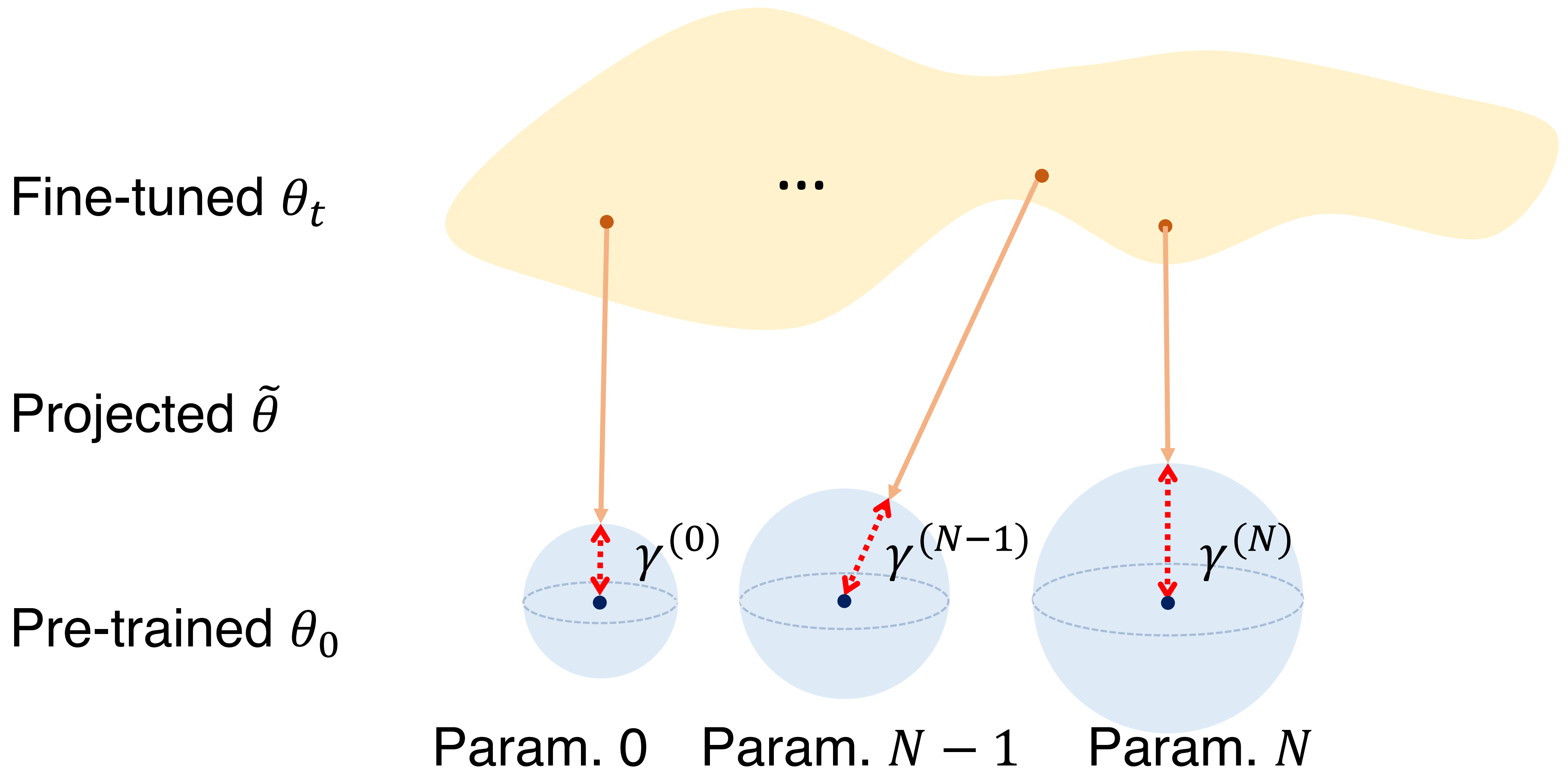
[1] Hendrycks, Dan, Kimin Lee, and Mantas Mazeika. "Using pre-training can improve model robustness and uncertainty." *ICML*, 2019.
[2] Wortsman, Mitchell, et al. "Robust fine-tuning of zero-shot models." CVPR. 2022.

Current Trends in Fine-tuning Research

- How to improve OOD generalization
  - Fine-tune a *subset* of layers [1]
  - Maintain a *close distance* to the pre-trained model [2]
  - Specialized fine-tuning for vision-language models [3]
- Open Questions
  - *Which* layers to fine-tune?
  - *How much* to fine-tune?

[1] Lee, Yoonho, et al. "Surgical fine-tuning improves adaptation to distribution shifts."ICLR, 2023.
[2] Gouk, Henry, Timothy M. Hospedales, and Massimiliano Pontil. "Distance-based regularisation of deep networks for fine-tuning." *ICLR, 2021.*
[3] Wortsman, Mitchell, et al. "Robust fine-tuning of zero-shot models." *CVPR,* 2022.

TPGM Diagram



Trainable Projected Gradient Descent

- Constrained bi-level minimization
  - Two types of parameters
    - Model parameters: θ
    - Hyper parameters: λ, γ (λ are optimization hyper-parameters)

$$\underbrace{\min_{\lambda,\gamma|(x,y)\in\mathcal{D}_{val}}}_{\text{Validation data}} \underbrace{\min_{\theta|(x,y)\in\mathcal{D}_{tr}}}_{\text{Training data}} \mathcal{L}(x,y;\theta,\lambda,\gamma) \quad \textbf{s.t.} \quad \underbrace{\|\theta-\theta_0\|_* \leq \gamma}_{\text{Constraints}}$$

Can we select a γ **for each layer** in neural networks through hyperparameter tuning on the validation data?

Trainable Projected Gradient Descent

- Observation: some projection operations have closed form solutions
  - Maximum Row Sum Norm (MARS norm) [1]

$$\Pi_{mars}(\theta_0, \theta_t, \gamma) : \tilde{\theta} = \theta_0 + \frac{1}{\max\left(1, \frac{\|\theta_t - \theta_0\|_\infty}{\gamma}\right)}(\theta_t - \theta_0)$$

- We can incorporate this operation into the computational graph.

[1] Gouk, Henry, Timothy M. Hospedales, and Massimiliano Pontil. "Distance-based regularisation of deep networks for fine-tuning." *ICLR, 2021.*

Trainable Projected Gradient Descent

TPGM iteratively optimizes the following objective function.

$$\underbrace{\min_{\lambda,\gamma|(x,y)\in\mathcal{D}_{val}}}_{\text{Step 2}} \underbrace{\min_{\theta|(x,y)\in\mathcal{D}_{tr}}}_{\text{Step 1}} \mathcal{L}(x,y;\theta,\lambda,\gamma) \quad \text{s.t.} \quad \underbrace{\|\theta-\theta_0\|_* \le \gamma}_{\text{Step 3}}$$

Algorithm 1: TPGM

```
Data: $\mathcal{D}_{tr}$,$\mathcal{D}_{val}$
Result: $\tilde{\theta}_{t+1}$
Initialize $\tilde{\theta}_0 = \theta_0, \gamma_0 = \epsilon$
for $t = \{0, ..., T-1\}$ do
Step 1    $\theta_{t+1} = \theta_t - \eta_t \nabla_\theta \mathcal{L}(x, y; \tilde{\theta}_t) \quad x, y \in \mathcal{D}_{tr}$
Step 2    $\gamma_{t+1} = \text{ProjectionUpdate}(\mathcal{D}_{val}, \theta_0, \theta_{t+1}, \gamma_t)$
Step 3    $\tilde{\theta}_{t+1} = \Pi(\theta_0, \theta_{t+1}, \gamma_{t+1})$
end
```

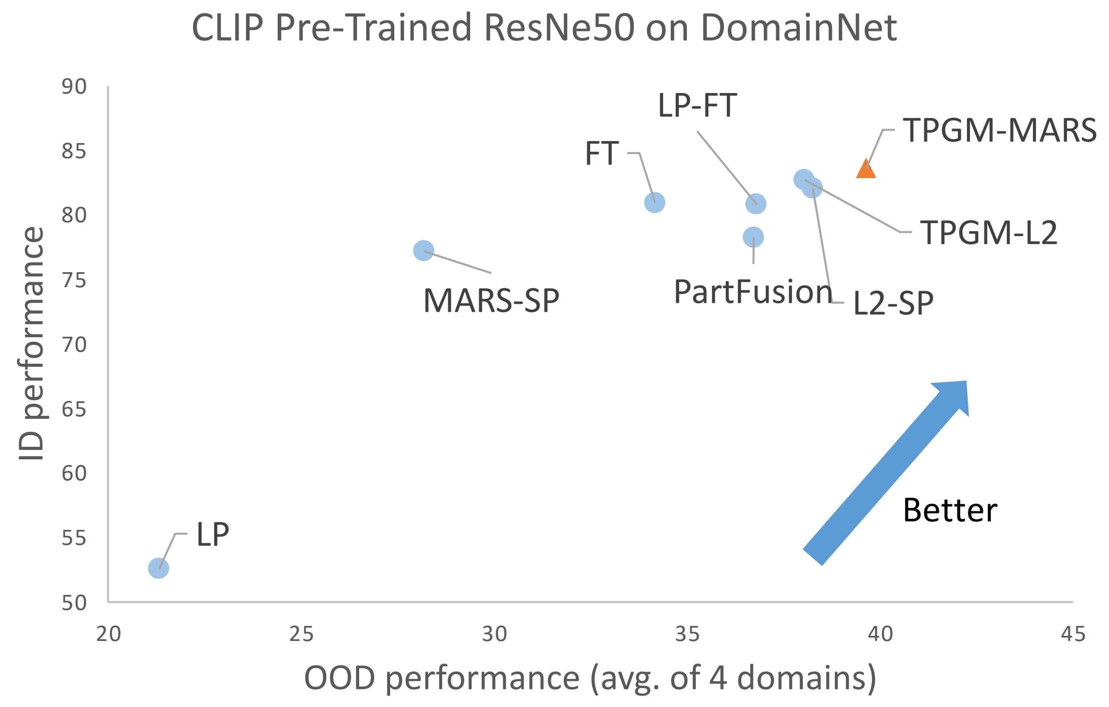
Results

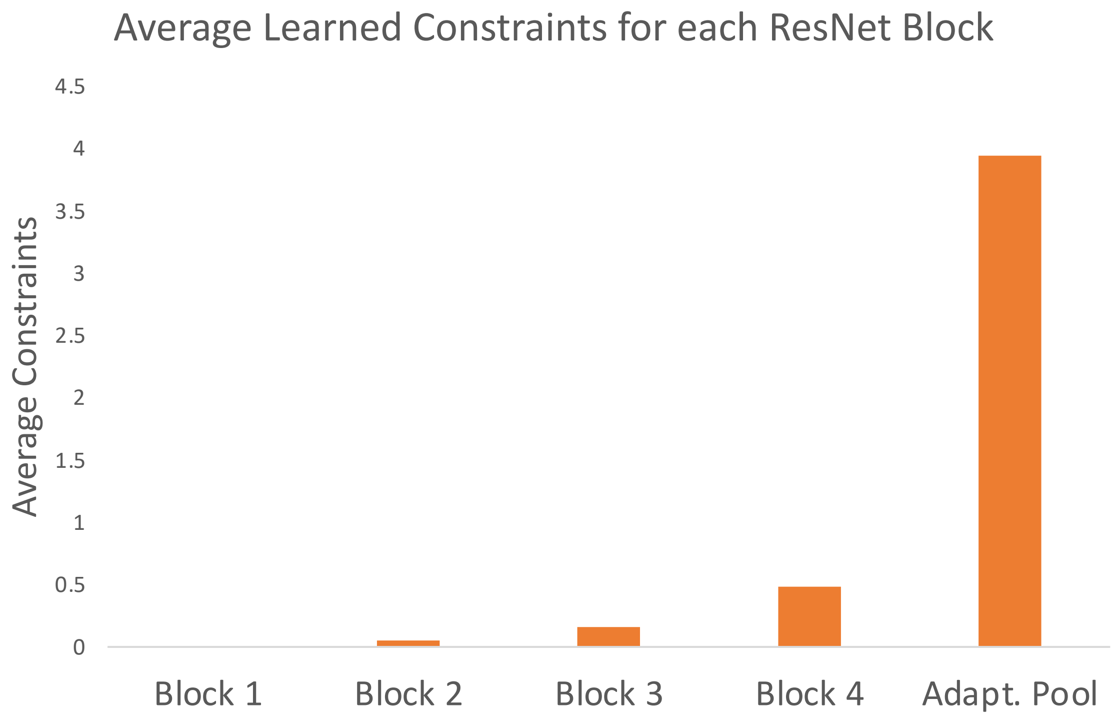
- Model: CLIP ResNet50
- DomainNet ID: Real
- DomainNet OOD: Sketch, Painting, Infograph, Clipart

clipart
infograph
painting

Source: http://ai.bu.edu/M3SDA/

Results





TPGM achieves the best ID performance and OOD robustness compared to prior works.

Under the distance constraints imposed by TPGM, most of the model changes are in the last adaptive pooling layer.

Conclusions and Limitations

- Benefits of TPGM
  - **Scalability**: TPGM can automatically learn a different constraint for each layer
  - **Explainability**: TPGM learns interpretable projection radii which agree with human intuitions.
  - **Performance**: TPGM improves robustness and achieves SOTA performance on several common benchmarks.
- Limitations
  - TPGM requires a separate nested training loop.