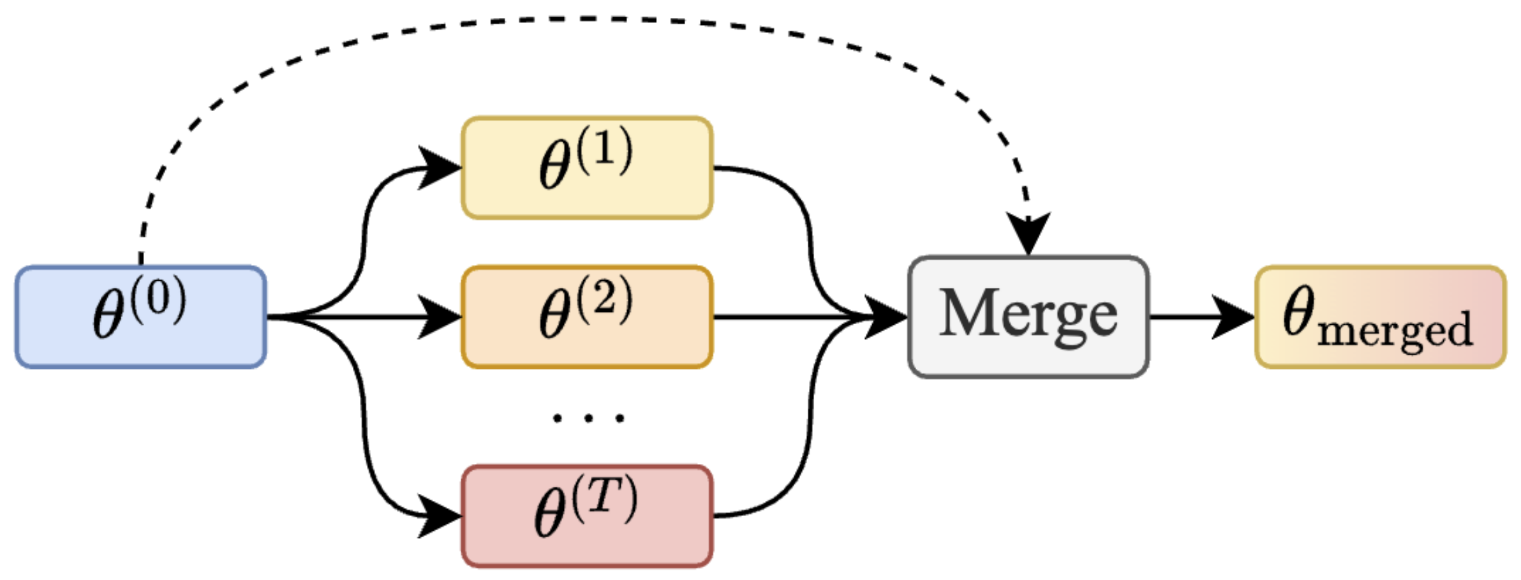
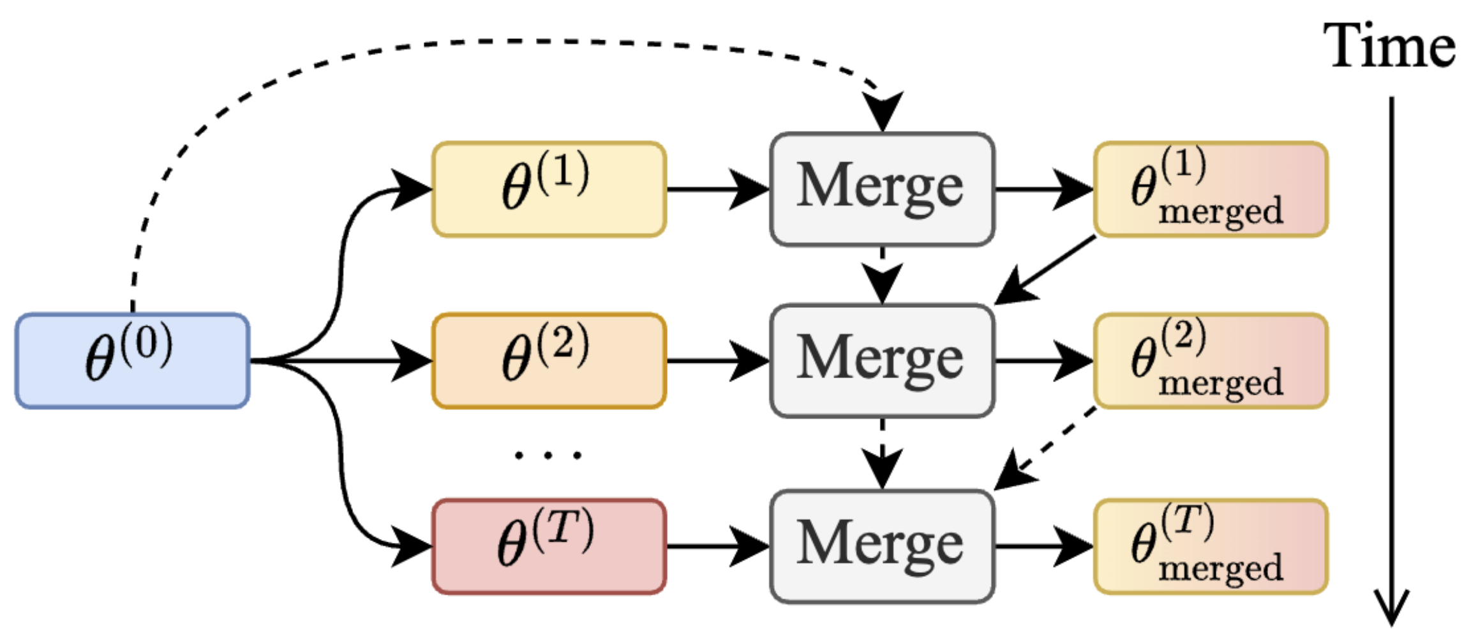
Merging on the Fly Without Retraining:
A Sequential Approach to Scalable Continual Model Merging

Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Lefei Zhang, Bo Du, Dacheng Tao

Two Types of Model Merging Paradigms



Conventional model merging
models are available simultaneously


Time
$\theta^{(0)}$
$\theta^{(1)}$
$\theta^{(2)}$
...
$\theta^{(T)}$
Merge
Merge
Merge
$\theta^{(1)}_{\text{merged}}$
$\theta^{(2)}_{\text{merged}}$
$\theta^{(T)}_{\text{merged}}$

Continual model merging
models are available sequentially

Why Continual Model Merging?

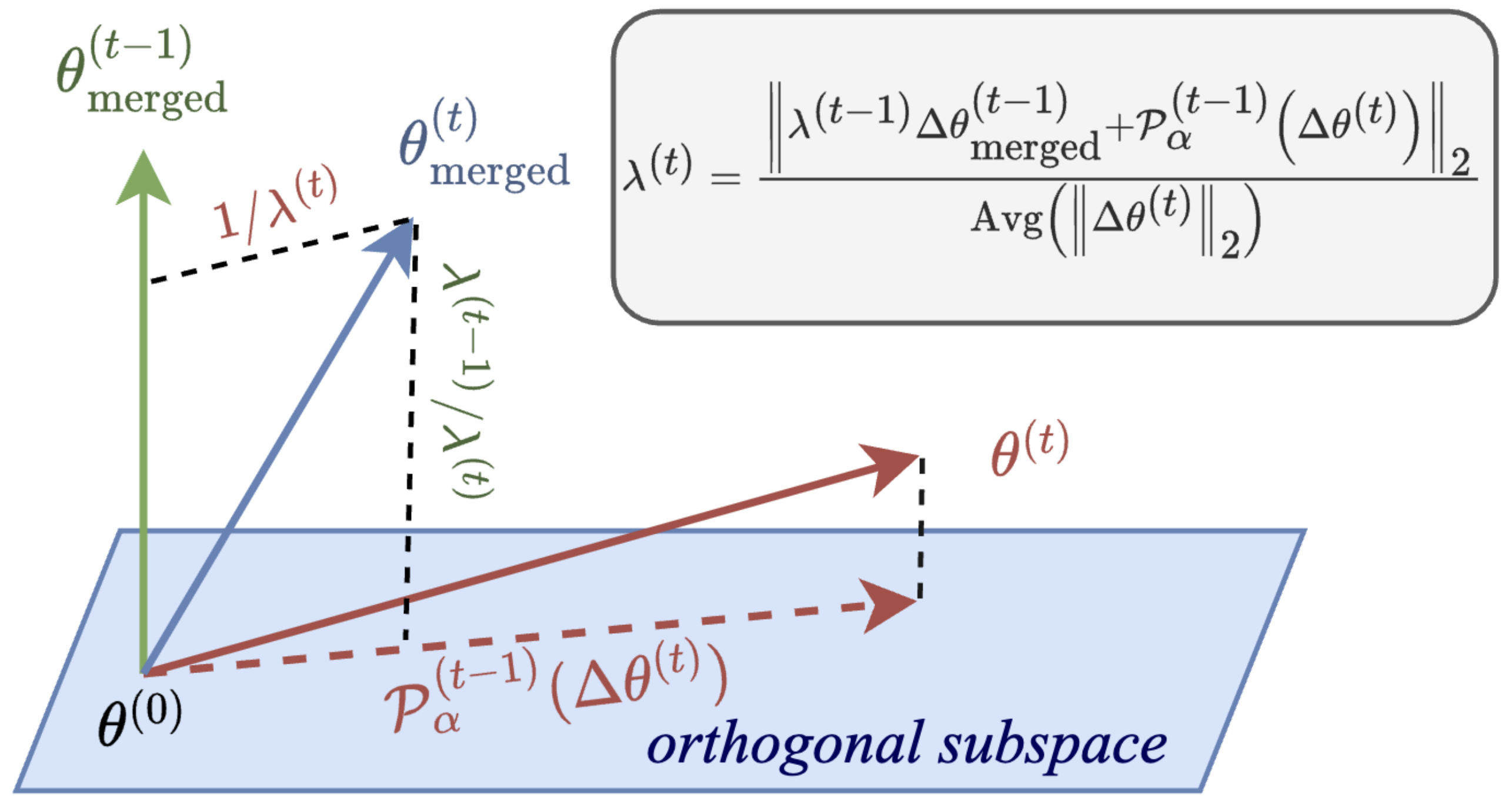
- **Memory Efficiency**: Conventional merging needs loading all expert models simultaneously, which scales linearly with the number of models. Continual merging keeps constant memory by only storing the current merged model + the new model.
- **Cheap**: The proposed continual model merging is training-free, so it avoids extra optimization and data access.
- **Better scalability**: Continual merging supports large task collections and frequent updates, unlike one-shot merging that becomes harder as the number of models grows.

Abstract of this paper

Deep model merging represents an emerging research direction that combines multiple fine-tuned models to harness their specialized capabilities across different tasks and domains. Current model merging techniques focus on merging all available models simultaneously, with weight interpolation-based methods being the predominant approach. However, these conventional approaches are not well-suited for scenarios where models become available sequentially, and they often suffer from high memory requirements and potential interference between tasks. In this study, we propose a training-free projection-based continual merging method that processes models sequentially through orthogonal projections of weight matrices and adaptive scaling mechanisms. Our method operates by projecting new parameter updates onto subspaces orthogonal to existing merged parameter updates while using an adaptive scaling mechanism to maintain stable parameter distances, enabling efficient sequential integration of task-specific knowledge. Our approach maintains constant memory complexity to the number of models, minimizes interference between tasks through orthogonal projections, and retains the performance of previously merged models through adaptive task vector scaling. Extensive experiments on CLIP-ViT models demonstrate that our method achieves a 5-8% average accuracy improvement while maintaining robust performance in different task orderings.

The Proposed Method

The **O**rthogonal **P**rojection-based **C**ontinual **M**erging (OPCM)


$\theta^{(t-1)}_{\text{merged}}$
$\theta^{(t)}_{\text{merged}}$
$1/\lambda^{(t)}$
$\lambda^{(t-1)}/\lambda^{(t)}$
$\theta^{(t)}$
$\theta^{(0)}$
$\mathcal{P}^{(t-1)}_{\alpha}\left(\Delta\theta^{(t)}\right)$
orthogonal subspace

$$\lambda^{(t)} = \frac{\left\|\lambda^{(t-1)}\Delta\theta^{(t-1)}_{\text{merged}} + \mathcal{P}^{(t-1)}_{\alpha}\left(\Delta\theta^{(t)}\right)\right\|_2}{\text{Avg}\left(\left\|\Delta\theta^{(t)}\right\|_2\right)}$$

Mark this paper

Project page

FusionBench
(Benchmark on model merging)

Algorithm 1 OPCM

```
Initialize $\theta^{(1)}_{\text{merged}} = \theta^{(1)}$, average norm of the task vectors $n = \|\Delta\theta^{(1)}\|_2$, scaling factor $\lambda^{(1)} = 1$.
for $t = 2$ to $T$ do
    for weight matrices $W \in \mathbb{R}^{m\times n}$ in linear layers do
        $\Delta W^{(t-1)}_{\text{merged}} \leftarrow W^{(t-1)}_{\text{merged}} - W^{(0)}$
        $\Delta W^{(t)} \leftarrow W^{(t)} - W^{(0)}$
        $\Delta W^{(t)}_{\text{proj}} \leftarrow \mathcal{P}^{(t-1)}_{\alpha}\left(\Delta W^{(t)}; \Delta W^{(t-1)}_{\text{merged}}\right)$
    end for
    for other parameters $p$ do
        $\Delta p^{(t)} \leftarrow p^{(t)} - p^{(0)}$
    end for
    $\Delta\theta^{(t-1)}_{\text{merged}} \leftarrow \theta^{(t-1)}_{\text{merged}} - \theta^{(0)}$
    Concatenate $\Delta W^{(t)}_{\text{proj}}$ and $\Delta p^{(t)}$ into $\Delta\theta^{(t)}_{\text{proj}}$
    $n \leftarrow \frac{t-1}{t}n + \frac{1}{t}\|\Delta\theta^{(t)}\|_2$
    $\lambda^{(t)} \leftarrow \left\|\lambda^{(t-1)}\Delta\theta^{(t-1)}_{\text{merged}} + \Delta\theta^{(t)}_{\text{proj}}\right\|_2 / n$
    $\theta^{(t)}_{\text{merged}} \leftarrow \theta^{(0)} + \frac{\lambda^{(t-1)}\Delta\theta^{(t-1)}_{\text{merged}} + \Delta\theta^{(t)}_{\text{proj}}}{\lambda^{(t)}}$
end for
return $\theta^{(T)}_{\text{merged}}$
```