タイトル

三ツ井 智哉

1. はじめに

機械学習モデルを実社会で運用する際には, 入力データの変化に柔軟に適応し, 長期間にわたり高い性能を保ち続けることが求められる. 監視システム, 製造ライン, オンラインサービスなど多くの応用領域において, 外部環境は絶えず変化するため, モデルは新しいデータや概念に順応し, 自身の知識を更新し続ける必要がある. 一方で, 新しいタスクに適応するたびに, 画像分類モデルである Vision Transformer (ViT) [1] などの大規模モデル全体を停止し, 過去の全データを再収集して一から再学習を行うことは, 計算資源の浪費やデータ管理のコスト, そしてサービス継続性の観点から非現実的である.

こうした制約を背景に, 新規データのみを用いて既存モデルを逐次更新する 継続学習 (Continual Learning; CL) が重要な研究課題となっている [2,3]. 特に, クラスインクリメンタル学習 (CIL) [4,5] と呼ばれる設定では, 未知のクラスを段階的に学習しつつ, 過去に学習したクラスの識別能力も維持することが求められる. しかし, 新しいデータのみでモデルを更新すると, 過去のタスクに最適化されていたパラメータが上書きされ, 獲得済みの知識が急速に失われる 破滅的忘却 (Catastrophic Forgetting) [6,7] が発生することが知られる. これは, ニューラルネットワークにおける「安定性と可塑性のジレンマ (Stability-Plasticity Dilemma)」 [8] として知られる根本的な課題に起因する. モデルが新規タスクに適応するためにパラメータを大きく更新すれば可塑性は高まるが, 既存の知識構造は破壊され安定性が失われる. 逆に安定性を重視して更新を抑制すれば, 新しい知識の獲得が阻害される. これを抑制し, 両者を高い次元で両立するために, これまで多様な手法が提案されている.

近年では, 大規模な事前学習済みモデルを凍結し, 少量の追加パラメータのみを更新して新規データに適応させる パラメータ効率型微調整 (Parameter-Efficient Fine-Tuning; PEFT) [9] が注目されている. なかでも Low-Rank Adaptation (LoRA) [10] は, 推論コストを増大させずに追加学習が可能であるため実運用に適している. しかし, LoRA を継続学習に単純適用した場合, 現在のタスクへの適応と過去の知識の保持という相反する要求を, 単一の低ランク行列の中で処理しなければならない. LoRA は全パラメータ空間を極めて低次元の部分空間で表現するため, 異なるタスク間でのパラメータ共有度が高く, 新規タスクのための更新が過去タスクの重要パラメータと干渉 (Interference) しやすいという構造的な問題がある. その結果, 新しい情報を学習しようとすれば過去が破壊され, 過去を守ろうとすれば学習が停滞するという, 構造的なトレードオフに直面することになる.

この問題に対し, アダプタを複数並列に管理する Mixture-of-Adapters (MoA) 系手法 [11,12] も提案されている. これらは, タスクごとに異なる専門家アダプタを空間的に切り替えるアプローチだが, 新たなアダプタは常にゼロから学習されるため, 過去の学習結果を次の学習に活用することが難しい. また, タスク数が増えた場合にはアダプタが無制限に増加し, メモリ管理の観点から現実的でない. さらに重要な課題は, 従来の LoRA や MoA が, データの流れの時間的な性質を考慮していない点にある.

実運用環境に流入するデータには, 瞬間的なノイズや一時的な流行もあれば, 長期にわたり頻出する本質的なパターンも混在している. この点において, 生物の脳機能は重要な示唆を与えてくれる. 認知神経科学における相補的学習システム (CLS) 理論 [13,14] によれば, 人間の脳は, 海馬における急速な学習（短期記憶）と, 大脳新皮質における緩やかな構造化（長期記憶）という, 学習率の異なる二つのシステムの相互作用によって知識を形成し, 安定性と可塑性を両立しているとされる. 機械学習モデルにおいても, パラメータの更新頻度や役割を時間軸で分離しなければ, 重要な知識が一時的な変動によって絶えず上書きされるリスクがある.

この問題に対し, Nested Learning [12,15] が提唱した時間階層を持つ記憶構造は理論的には有力な解決策である. しかし, 既存の Nested Learning はモデル全体に時間スケールを組み込む必要があり, 巨大モデルを一から再学習し直す必要があるため, 実応用に直結する形で採用するのは困難である. 他方, 実際の応用では, 事前学習済みモデルを大幅に変更する必要はなく, 追加パラメータとしての LoRA のみが適応を担えば十分なケースが多い. 事前学習済みモデルは基礎能力として維持し, 外付けの記憶モジュールだけを柔軟にアップデートするほうが実運用システムの要請に合致する.

そこで本研究では, Nested Learning の思想である「記憶の時間スケール分離」を, LoRA の内部構造として軽量に実装した Dual LoRA を提案する. 本手法では, Fast LoRA と Slow LoRA という役割の異なる二つのアダプタを導入する. Fast LoRA は高い学習率で毎ステップ更新され, 一時的, 変動的なパターンを迅速に吸収する短期記憶として機能する. 一方 Slow LoRA は低い学習率かつタスク境界などのタイミングで Fast の知識を統合し, 頻出, 本質的な構造を徐々に蓄積する長期記憶として機能する.

本研究ではこの構造に基づき, 事前学習済み ViT-B/16 の各 Transformer 層に Fast, Slow LoRA の 2 つの LoRA を接続し, CIFAR-100 [16] の CIL 設定において挙動を分析した. さらに, Fast LoRA を複数用意して並列学習し, 各タスクの検証性能に基づいて最適な Fast を選択する拡張も検討し, 短期記憶の多様性が性能に与える効果について初期検討を行った. この実験結果から, 提案手法が一つの LoRA を継続的に用いるベースライン手法を上回る平均精度を達成することを確認した. さらに, 異なる学習率の複数の Fast LoRA を用いて学習し, 最良のものを選択する拡張手法 Multiple Fast LoRA (MFL) では, さらに高

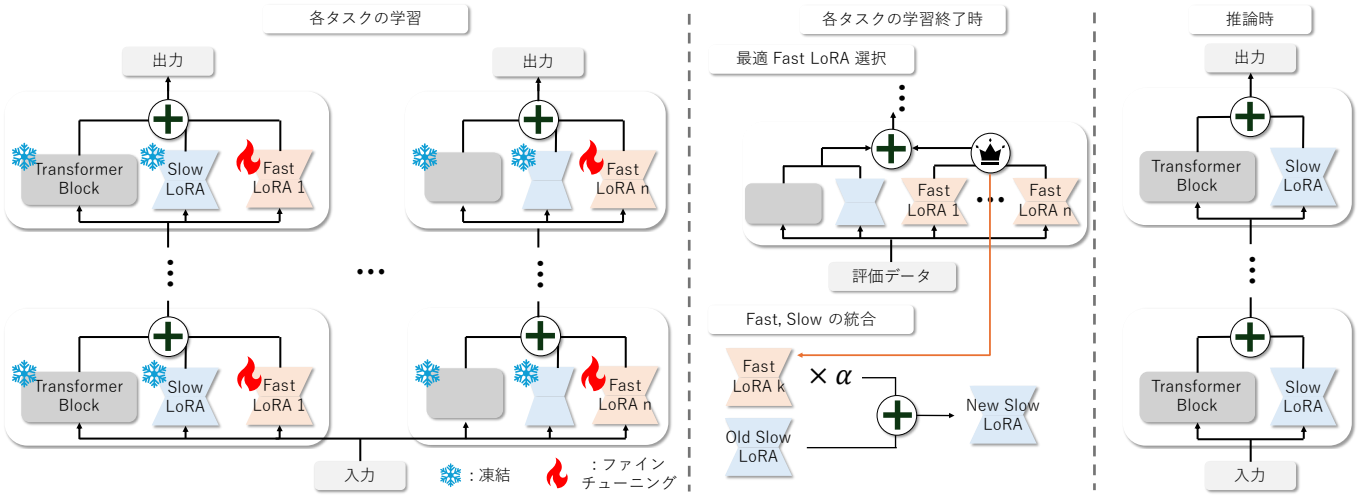

図 1: 提案手法（Dual LoRA）の全体概要左：学習フェーズでは, 事前学習済みモデルと Slow LoRA を凍結し, 学習率の異なる複数の Fast LoRA のみを並列に学習させる. 中央：学習終了後, 検証データを用いて最も性能の良い Fast LoRA を選択し, そのパラメータを係数 α で重み付けして Slow LoRA に統合する. 右：推論フェーズでは, 短期記憶をリセットし, 長期記憶を蓄積した Slow LoRA のみを用いて推論を行う.

い平均精度 87.57% を達成することを確認した.

本研究の貢献は以下のとおりである.

- LoRA に時間軸の概念を導入し, 短期適応と長期記憶を分離した初の PEFT ベース継続学習手法を提示する.
- 巨大モデルを再学習せずに Nested Learning のエッセンスを活かす軽量フレームワークを実現する. 巨大モデルを再学習することなく, Nested Learning の利点を軽量な追加モジュールのみで実現するフレームワークを構築する.
- 実験において, 提案手法, 特に MFL により過去タスクの知識保持能力および学習後半のタスクにおける適応能力が向上することを実証した.

2. Dual LoRA with Distinct Update Speeds

本研究では, 事前学習済みモデルの知識を維持しつつ, 新規タスクへの適応と過去知識の蓄積を両立させるため, Nested Learning [15] の時間階層概念を PEFT に導入した Dual LoRA を提案する. 本手法は, 短期的な変動を吸収する Fast LoRA と, 長期的な知識を蓄積する Slow LoRA の二重構造を持ち, タスク境界において短期記憶を長期記憶へ統合する機構を備える.

2.1 予備知識: Low-Rank Adaptation (LoRA)

本研究ではベースモデルとして Vision Transformer (ViT-B/16) を採用し, そのパラメータ $W_0 \in \mathbb{R}^{d\times k}$ を凍結する. LoRA [10] は, 重み更新量 ΔW を低ランク行列の積 BA ($B \in \mathbb{R}^{d\times r}, A \in \mathbb{R}^{r\times k}$) によって近似する手法である.

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{1}$$

ここで $r \ll \min(d, k)$ はランク数である. 本研究では, この LoRA モジュールを Fast と Slow の 2 種類並列に配置する構成をとる.

2.2 Dual LoRA の構成

提案モデルは, 各 Transformer 層に対して以下の 3 つの要素で構成される.

a） Fast LoRA (短期記憶)

Fast LoRA ($A_{\text{fast}}, B_{\text{fast}}$) は, 現在のタスク固有の特徴や一時的なデータ変動に迅速に適応する役割を担う. 高い学習率 (例: $\eta_{\text{fast}} = 1.0\times10^{-2}$) で最適化され, 高い可塑性 (Plasticity) を持つ.

b） Slow LoRA (長期記憶)

Slow LoRA ($A_{\text{slow}}, B_{\text{slow}}$) は, タスクを跨いで共有される普遍的な知識を蓄積する役割を担う. Slow LoRA はゼロで初期化され ($\Delta W_{\text{slow}} = 0$), 学習中の直接的な勾配更新は行わない（もしくは極めて低い学習率を設定する）. 後述する統合フェーズにおいてのみ Fast LoRA から知識が転送されることで, 安定性 (Stability) を保ちながら長期記憶を形成する.

2.3 fast と slow の記憶の統合

タスク T_i の学習終了時, Fast LoRA が獲得した短期的な知識を Slow LoRA へと統合 (Consolidate) し, 次のタスク T_{i+1} に備えて Fast LoRA をリセットする. 本研究では, 以下の 2 つの統合戦略を比較・検討する.

a） A. Task Arithmetic (単純加算)

最も単純なアプローチとして, 学習済みの Fast LoRA のパラメータを重み付き和として Slow LoRA に加算する.

$$\theta_{\text{slow}} \leftarrow \theta_{\text{slow}} + \alpha \cdot \theta_{\text{fast}} \tag{2}$$

ここで θ は LoRA のパラメータ (A, B) を指し, α は統合率 (例: 0.1) である. この手法は計算コストが低い一方, タスク間で勾配方向が競合する場合, 干渉を考慮できない欠点がある.

b） B. PAM-lite (符号整合による干渉抑制)

モデルマージにおける干渉を防ぐ手法として, Sokar ら [17] は Parameter Alignment for Merging (PAM) を提案している. PAM は, 新しく学習したパラメータを過去のパラメータと統合する際, パラメータのアライメント（整列）を学習中に明示的に行うことで, 干渉を最小限に抑える手法である.

本研究では, PAM の概念を単純化し, 事後的な符号整合のみを行う **PAM-lite** を導入する. これは, Fast LoRA と Slow LoRA の更新方向が一致する（正の転移が期待される）要素のみを加算し, 符号が異なる（忘却や干渉につながる）要素の更新を棄却する戦略である. 具体的には, パラメータの各要素 j について, 以下の条件に従って Slow LoRA を更新する.

$$(\theta_{\text{slow}})_j^{\text{new}} \leftarrow (\theta_{\text{slow}})_j^{\text{old}} + \delta_j \tag{3}$$

$$\delta_j = \begin{cases} \alpha(\theta_{\text{fast}})_j & \text{if } (\theta_{\text{slow}})_j^{\text{old}} \cdot (\theta_{\text{fast}})_j \geqq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

ここで θ は LoRA のパラメータ (A, B) を指し, α は統合率 (例: 0.1) である. この単純なフィルタリングにより, 長期記憶にとって有害な急激な変動を遮断し, 安定した知識のみを蓄積する.

2.4 推論時の挙動

学習時は Fast と Slow の両方を用いて予測を行うが, 推論時（運用時）においては Fast LoRA (リセット済み) は使用せず, Slow LoRA のみを用いて推論を行う. これは, 統合機構によってタスクに依存しない汎用的な特徴のみが Slow LoRA に蓄積されているためである. また, クラスインクリメンタル学習 (CIL) の設定では, 推論時に入力データがどのタスクに属するか（タスク ID）が不明であるため, 特定のタスクに過剰適合した Fast ではなく, 汎化性能の高い Slow のみを使うことが合理的である. したがって, 推論時の出力 h は次式となる.

$$h = W_0 x + \Delta W_{\text{slow}} x \tag{5}$$

2.5 拡張: Multiple Fast LoRA

Fast LoRA の学習率設定は, 新規タスクへの適応速度を決定する重要な要因である. 単一の学習率では多様なデータ分布に対応しきれない可能性があるため, 本研究では学習率の異なる複数の Fast LoRA を並列学習させて, 最良の Fast LoRA を選択する拡張手法を提案する.

具体的には, K 個の Fast LoRA $\{\text{Fast}_1, \ldots, \text{Fast}_K\}$ を用意し, それぞれ異なる学習率 (例: $\eta \times \{1.0, 0.5, 0.1, \ldots\}$) で並列に学習させる. タスク学習終了後, 検証用データ (Validation Set) を用いて各 Fast LoRA の性能を評価し, 最も精度の高いモデル Fast_{k^*} を選択する.

$$k^* = \underset{k}{\text{argmax}}\, \text{Accuracy}(\text{Fast}_k; \mathcal{D}_{\text{val}}) \tag{6}$$

選択された Fast_{k^*} のみを前述の記憶統合によって Slow LoRA へ統合する. これにより, タスクの難易度や性質に応じた最適な可塑性を適応的に選択することが可能となる.

a） 計算コストについて

この拡張手法では, 学習時に K 個の Fast LoRA を並列に計算するため, 学習コスト（GPU メモリおよび計算量）は単一の LoRA に比べて約 K 倍に増加する. しかし, 推論時には前述の通り単一の Slow LoRA のみが使用されるため, 運用時の推論コストやレイテンシはベースモデル + LoRA 1 つ分と変わらず, 軽量性を維持できる点が利点である.

3. 実験設定

本節では, 提案手法である Dual LoRA およびその拡張である Multiple Fast LoRA 戦略の有効性を検証するための実験条件について述べる.

3.1 データセット

継続学習における標準的なベンチマークである CIFAR-100 を用いた. CIFAR-100 は 100 クラス, 50,000 枚の訓練画像, 10,000 枚のテスト画像から構成され, 各画像は 32×32 の RGB 画像である. 本実験では, クラスインクリメンタル学習 (Class-Incremental Learning; CIL) の設定に従い, 全 100 クラスをランダムにシャッフルした後, 10 クラスずつの 10 タスクに分割して順次学習を行った (10-task Split CIFAR-100). データの前処理として, 学習時には 224×224 解像度へのリサイズおよび RandomHorizontalFlip を適用し, テスト時には CenterCrop を適用した.

3.2 実装詳細

ベースモデルとして, ImageNet で事前学習済みの ViT-B/16 を採用し, バックボーンのパラメータは実験を通じて凍結した. 追加パラメータとして, 各 Transformer 層に Dual LoRA (Rank=16) を挿入した. 最適化手法には SGD を用い, Fast LoRA の学習率は 0.01, Slow LoRA の学習率は 0.001 に設定した. また, Multiple Fast LoRA (MFL) 設定においては, 異なる学習率設定を持つ Fast LoRA を複数並列化して学習し, 検証データ (各タスクの訓練データの 10%) に基づいて最適な Fast LoRA を選択する手法を採用した. 各タスク終了後の統合手法として, 単純な加算平均である Task Arithmetic (TA) と, 干渉を抑制する PAM-lite の 2 種類を比較検証した.

4. 実験結果と考察

4.1 定量的評価

CIFAR-100 の 10 タスク分割設定における最終的な全クラス分類精度 (Final Accuracy) および, 全タスク学習過程における平均精度 (Average Accuracy) を表 1 に示す. 比較対象として, LoRA を単一モジュールとして継続学習させるベースライン (LoRA 単体), Fast LoRA を単一の経路のみで学習させる基本構成 (Dual LoRA), および提案手法である並列選択拡張 (Parallel Selection) の比較を行った.

a） LoRA 単体ベースラインとの比較

LoRA 単体ベースラインは Final Accuracy 75.33%, Average Accuracy 86.48% を達成した. 一方, 提案する Dual LoRA 構造を導入することで, Final Accuracy は 81.34% (+6.01%), Average Accuracy は 87.31% (+0.83%) と大幅な改善が得られた. この改善は, Fast/Slow の二重構造による時間スケール分離が, 破滅的忘却の抑制に有効であることを示している.

b） 旧タスクと新タスクの精度バランス

表 1 の「旧タスク」列と「新タスク」列に注目すると, LoRA 単体ベースラインは新タスク (Task 5〜9) で 85.6% と高い精度を維持する一方, 旧タスク (Task 0〜4) では 65.1% と著しい忘却が見られる. これに対し, Dual LoRA は旧タスクの精度を 82.4% に向上させ (+17.3%), 新旧タスク間のバランスを大幅に改善した. これは, Slow LoRA が長期記憶として過去タスクの知識を安定的に保持していることを示唆する.

c） Multiple Fast LoRA (MFL) の効果

MFL 拡張を導入した場合, Average Accuracy は 87.57% とさらに向上し (+0.26%), 特に新タスクの精度が 81.1% と Dual LoRA (80.3%) より改善された. これは, タスクごとに最適な学習率の Fast LoRA を選択することで, 新タスクへの適応と旧タスクの保持を両立できたためと考えられる.

4.2 タスクごとの推移と忘却の分析

提案手法の効果をより詳細に分析するため, LoRA 単体ベースライン, Dual LoRA, および MFL におけるタスクごとの精度推移を表 2 に示す.

表 2 より, LoRA 単体は学習初期 (T0: 98.6%) では高い精度を示すが, タスクが進むにつれて精度の低下が顕著であり, 最終タスク終了時には 75.33% まで低下した. 一方, Dual LoRA および MFL は, T0 では LoRA 単体より低い (96.5%) ものの, 後半タスクでの精度維持が改善され, 特に T7〜T9 において LoRA 単体より +2〜6% 高い精度を維持している.

4.3 クラスグループ別精度の分析

各手法の忘却パターンをより詳細に分析するため, Task 9 終了後のクラスグループ別精度を表 3 に示す.

表 3 から, 以下の特徴が読み取れる.

a） LoRA 単体の忘却パターン

LoRA 単体は, 最も古いタスクであるクラス 00-09 で 50.3%, クラス 10-19 で 56.4% と, 旧タスクにおいて深刻な忘却が発生している. 一方, 最新タスクであるクラス 80-89 (93.9%) やクラス 90-99 (92.0%) では

表 1: CIFAR-100 10-task CIL における精度比較. LoRA 単体ベースラインと提案手法の比較を示す. 旧タスクは Task 0〜4（クラス 00-49）, 新タスクは Task 5〜9（クラス 50-99）の平均精度である.

手法	Final Acc (%)	Avg Acc (%)	旧タスク (%)	新タスク (%)
LoRA 単体 (Baseline)	75.33	86.48	65.1	85.6
Dual LoRA + TA	81.34	87.31	82.4	80.3
Dual LoRA + PAM	81.27	87.29	82.2	80.2
MFL + TA (提案)	**81.69**	**87.57**	82.3	**81.1**
MFL + PAM (提案)	**81.71**	87.50	**82.5**	80.9

表 2: 各タスク終了時における Top-1 精度 (%) の推移. 各列はそのタスク終了時点での全既習クラスに対する精度を示す.

手法	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9
LoRA 単体	98.6	95.0	92.0	90.42	87.0	85.43	84.31	80.31	76.37	75.33
Dual LoRA	96.5	92.75	90.97	89.55	87.14	85.25	85.16	82.66	81.80	81.34
MFL	96.5	92.65	91.10	89.28	87.44	85.60	85.71	83.29	82.47	81.69

表 3: Task 9 終了後のクラスグループ別精度 (%). 各列は対応するタスクで学習されたクラス（10 クラス）の精度を示す.

手法	00-09	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-89	90-99
LoRA 単体	50.3	56.4	73.6	73.7	71.3	75.7	81.3	85.1	93.9	92.0
Dual LoRA	78.7	77.3	89.1	83.1	83.7	74.3	85.4	74.7	85.2	81.9
MFL	79.7	79.9	89.6	83.6	78.7	78.3	85.5	76.3	84.3	81.0

高い精度を維持しており, 典型的な「新しいタスクに偏った」忘却パターンを示している.

b） Dual LoRA による旧タスクの保持

Dual LoRA は, クラス 00-09 で 78.7% (+28.4%), クラス 10-19 で 77.3% (+20.9%) と, 旧タスクの精度を大幅に改善した. これは, Slow LoRA が長期記憶として機能し, 過去タスクの知識を統合・保持していることを示す重要な証拠である.

c） MFL の効果

MFL は, 特にクラス 10-19 で 79.9% と Dual LoRA (77.3%) より +2.6% 高い精度を達成した. ただし, クラス 40-49 では 78.7% と Dual LoRA (83.7%) より -5.0% 低下しており, Expert 選択が一部のタスクで最適でなかった可能性がある. これは, 検証データ (10%) のサイズが限られているため, Expert 選択の精度に限界があることを示唆している.

5. 統合手法の比較と課題

統合手法として比較した Task Arithmetic (TA) と PAM-lite については, 表 1 が示す通り, その性能差は微小 (0.1% 未満) であった. PAM-lite は符号の不整合をフィルタリングすることで干渉を抑制する手法であるが, 本実験設定においては単純な加算平均 (TA) と同等の結果となった. また, クラスグループ 40-49 (Task 4) においては, Step1 と比較して Step2 で精度低下が見られた. これは MoE の選択プロセスにおいて, 検証データ (10%) では最適と判断された Expert が, テストデータに対しては必ずしも最適ではなかった可能性を示唆しており, Expert 選択基準の改善やアンサンブル手法の導入が今後の課題として挙げられる.

6. ま と め

本研究では, 継続学習における破滅的忘却を抑制するため, LoRA に時間スケールの概念を導入した Dual LoRA を提案した. 提案手法は, 高学習率で新タスクに適応する Fast LoRA（短期記憶）と, 低学習率で過去の知識を保持する Slow LoRA（長期記憶）の二重構造を持ち, タスク境界において Fast から Slow へ知識を統合する機構を備える.

CIFAR-100 の 10 タスク分割設定における実験の結果, 以下の知見が得られた. (1) Dual LoRA は LoRA 単体ベースラインと比較して Final Accuracy を +6.01% (75.33% → 81.34%) 改善し, 特に旧タスク (Task 0〜4) の精度を 65.1% から 82.4% へと大幅に向上させた. (2) 複数の学習率を持つ Fast LoRA を並列学習させ, 検証精度に基づいて最適な候補を選択する Multiple Fast LoRA (MFL) により, Average Accuracy は 87.57% とさらに向上した. (3) 統合手法として Task Arithmetic と PAM-lite を比較したが, 性能差は 0.1% 未満と微小であった.

今後の課題として, 複数シードでの統計的検証, Expert 選択手法の改善（検証データサイズの増加やアンサンブル手法の導入）, および他データセット・他モダリティへの拡張が挙げられる.

文　献

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021, pp. 1–22.

[2] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 46, no. 8, pp. 5362–5383, 2024.

[3] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural networks*, vol. 113, pp. 54–71, 2019.

[4] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2001–2010.

[5] M. Mundt, Y. Hong, I. Pliushch, and V. Ramesh, "A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning," *Neural Networks*, vol. 160, pp. 306–336, 2023.

[6] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[7] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[8] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," p. 504, 2013.

[9] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799.

[10] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022.

[11] J. Yu, Y. Zhuge, L. Zhang, P. Hu, D. Wang, H. Lu, and Y. He, "Boosting continual learning of vision-language models via mixture-of-experts adapters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 23 219–23 230.

[12] H. Wang, H. Lu, L. Yao, and D. Gong, "Self-expansion of pre-trained models with mixture of adapters for continual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 10 087–10 098.

[13] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory." *Psychological review*, vol. 102, no. 3, p. 419, 1995.

[14] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? complementary learning systems theory updated," *Trends in cognitive sciences*, vol. 20, no. 7, pp. 512–534, 2016.

[15] A. Behrouz, M. Razaviyayn, P. Zhong, and V. Mirrokni, "Nested learning: The illusion of deep learning architectures," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[16] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[17] G. Sokar, G. K. Dziugaite, A. Arnab, A. Iscen, P. S. Castro, and C. Schmid, "Continual learning in vision-language models via aligned model merging," *arXiv preprint arXiv:2506.03189*, 2025.