PRMU2022

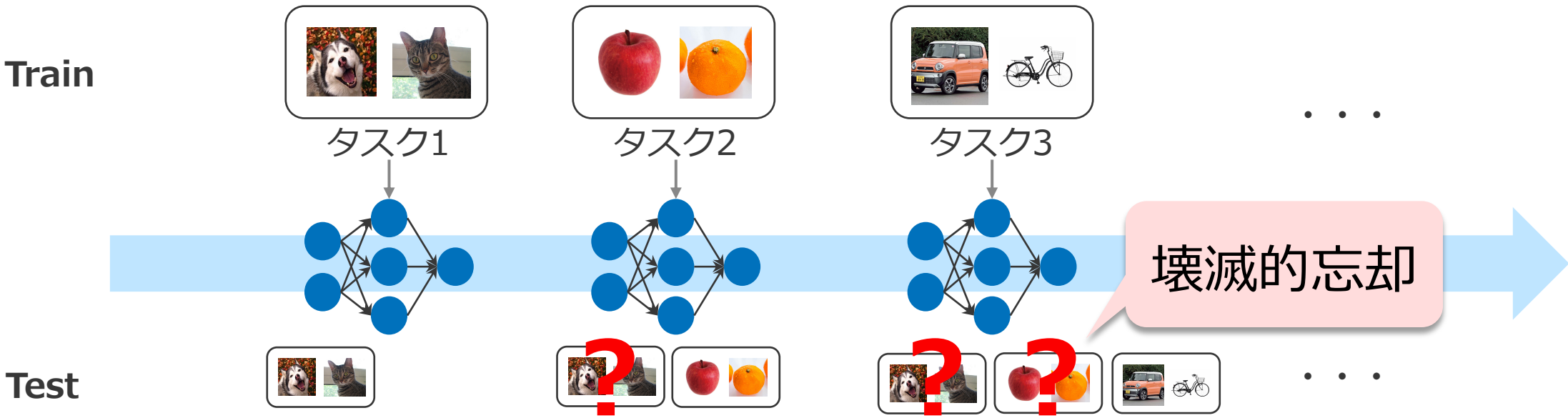
Vision Transformerにおける Continual Learning

電気通信大学　大学院　情報学専攻
武田 麻奈　柳井啓司

1. 研究背景

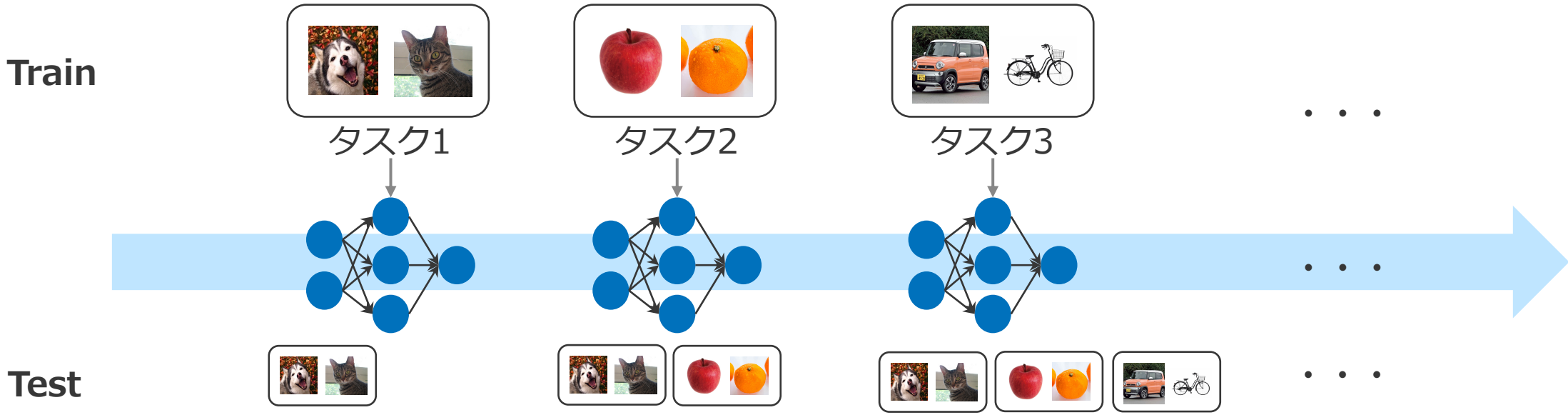
- 人間の脳は段階的に学び，スキルを身につけ，新しい課題の解決のためにそれまでの知識を応用できる

- それに対して，深層学習モデルでは，新たなタスクが与えられた場合，過去に学習したタスクを忘れてしまう（壊滅的忘却）



1. 研究背景

- Continual Learning では，この問題に対処し，過去に学習したタスクの知識を保持しながら，新しいタスクを継続的に学習する



1. 研究背景

- 近年，自然言語処理で使用されるアーキテクチャである Transformer を，コンピュータビジョンに活用した **Vision Transformer** がCNNを超える精度を示した

- 従来の Continual Leaning 手法は，一般的に CNN に適用することを考慮しているため，**Vision Transformer へ適用できる手法は限られる**

- CNN と比べてモデルサイズが大きい Vision Transformer は，Continual Learning 手法を適用した際の追加のモデルサイズが大きくなる

  → CNNへの適用を前提とする**従来手法と比べて，少ないパラメータ数で壊滅的忘却を抑制する必要がある**

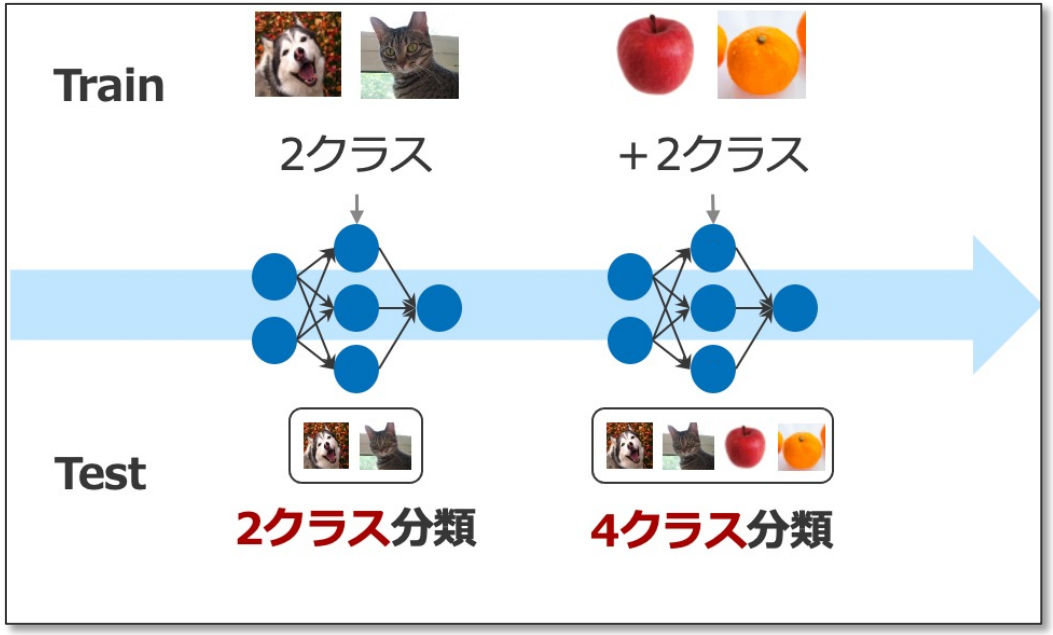
目的

Vision Transformer へ適用可能 かつ,

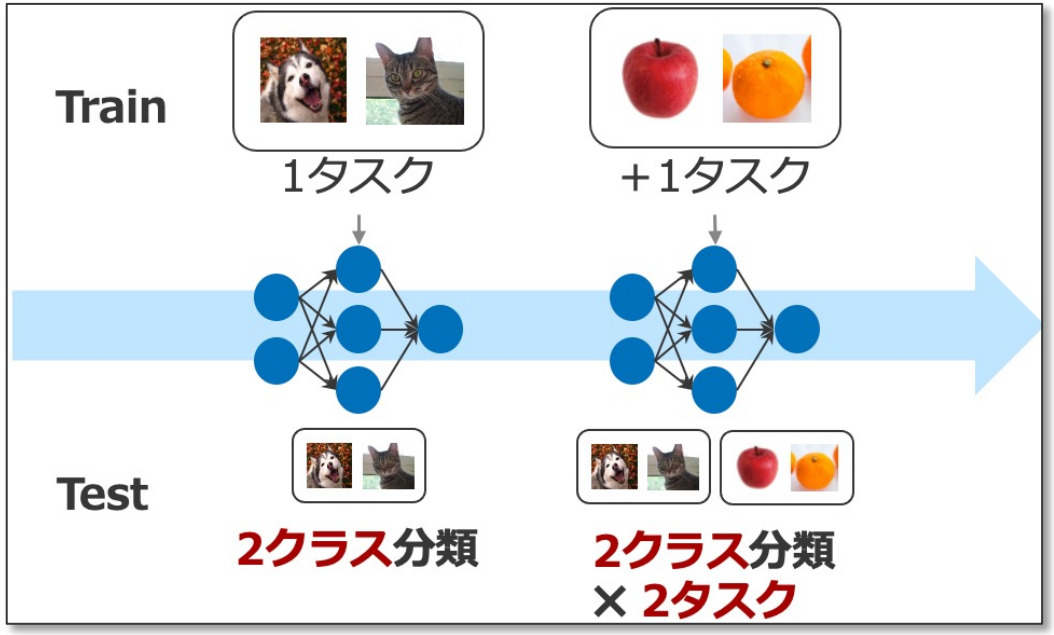
少ないパラメータ数で壊滅的忘却を抑制する 手法を提案する

3. 関連研究 ー Continual Learning ー

- Continual Learning では，過去に学習したタスクの知識を保持しながら，新しいタスクを継続的に学習する
  - **クラスインクリメンタル**：新たなクラスが追加される
  - **タスクインクリメンタル**：新たなタスクが追加される



▲ クラスインクリメンタル



▲ タスクインクリメンタル

3. 関連研究 ー Continual Learning ー

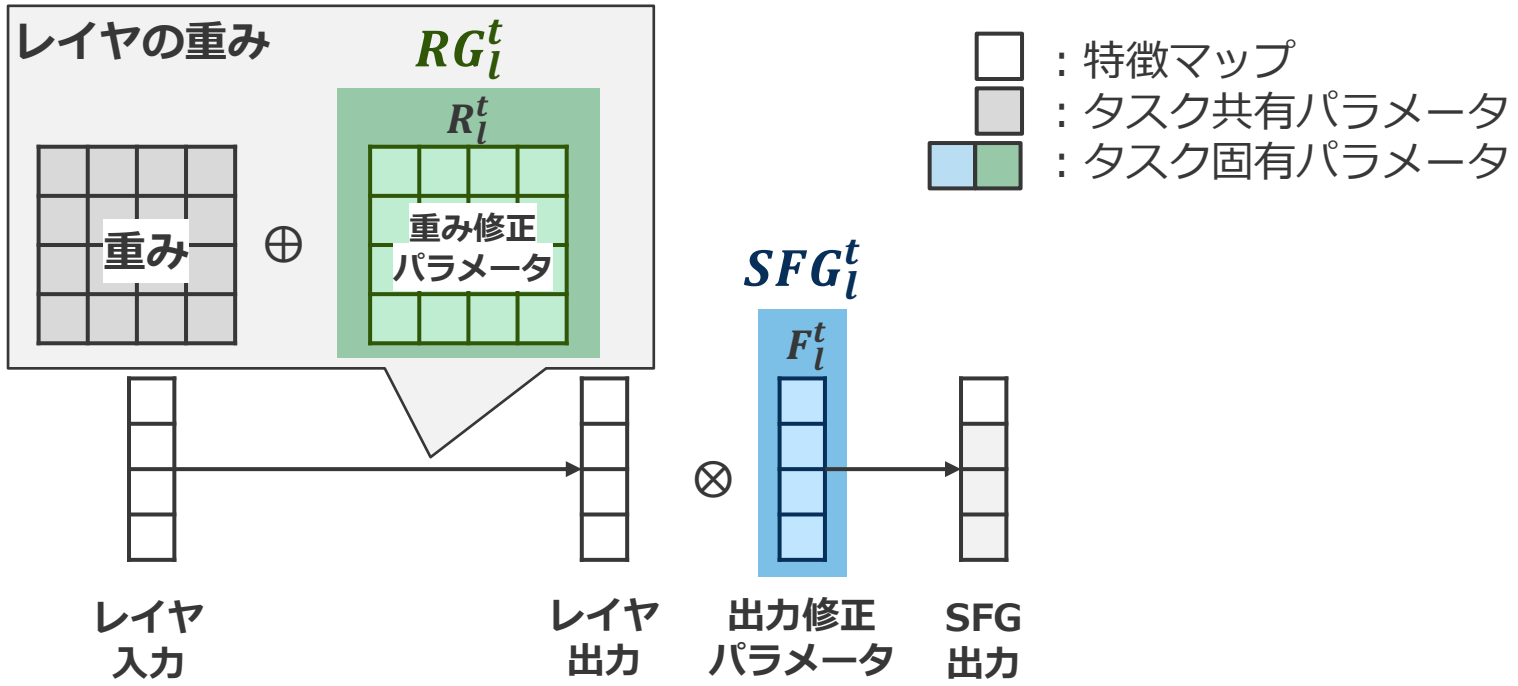
- **リハーサル** [Hetherington et al. 1989]
  - 古いサンプルで新しいサンプルを学習する
- **蒸留** **Learning without Forgetting [Liand Hoiem 2016]**
  - 学習済みモデルを使用して過去タスクのラベルを再現し，新しい学習に使用する
- **正則化** **Elastic Weight Consolidation [Kirkpatrick et al. 2016]**
  - 重みの重要性に応じて，新タスクの重みを学習する
- **プルーニング** **PackNet [Mallya et al. CVPR2018]**
  - 過去タスクの学習重みを枝刈りし，不要パラメータのみを新タスクの学習で上書き
- **重み選択** **Piggyback [Mallya et al. ECCV2018]**
  - 固定バックボーンネットワークからタスク固有の重みを選択する
- **タスク固有の重み追加** **Rectification-based Knowledge Retention [Singh et al. CVPR2021]**
  - バックボーンの重みは固定し，タスクごとに新しいパラメータを追加する

3. 関連研究 ー Continual Learning ー

- **Rectification-based Knowledge Retention (RKR)**

  [1] Singh et al. Rectification-based Knowledge Retention for Continual Learning. CVPR 2021

  – ベースパラメータに対して，タスク固有の修正パラメータを適用
    - **R**ectification **G**enerator (**RG**)：重みを修正するパラメータ
    - **S**caling **F**actor **G**enerator (**SFG**) ：中間出力を修正するパラメータ



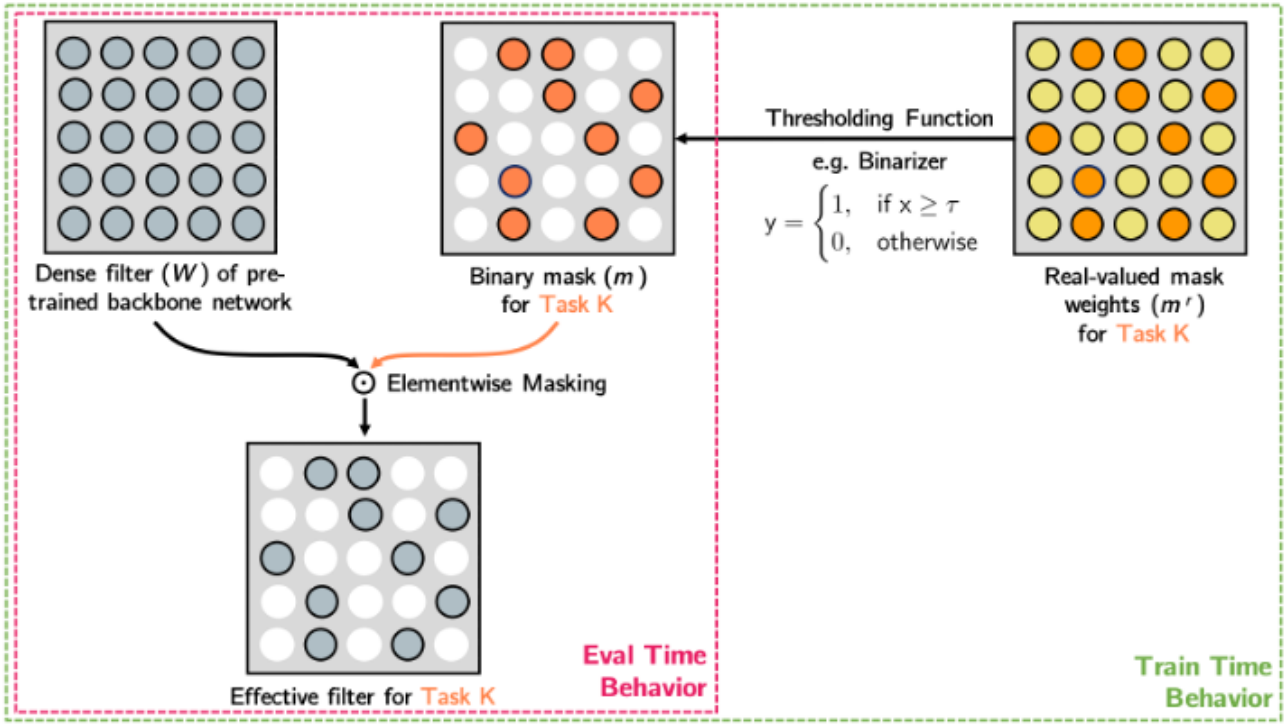
3. 関連研究 ー Continual Learning ー

- **Piggyback**

  [3] Arun et al. Piggyback: Adding multiple tasks to a single, fixed network by learning to mask. ECCV 2018

  – ベースモデルの重みに対して，学習した重みマスクを適用して出力を変換する

  – 重みマスクはバイナリマスクで表されるため，追加パラメータ数は少ない



3. 関連研究 ー Vision Transformer ー

- **ViT**

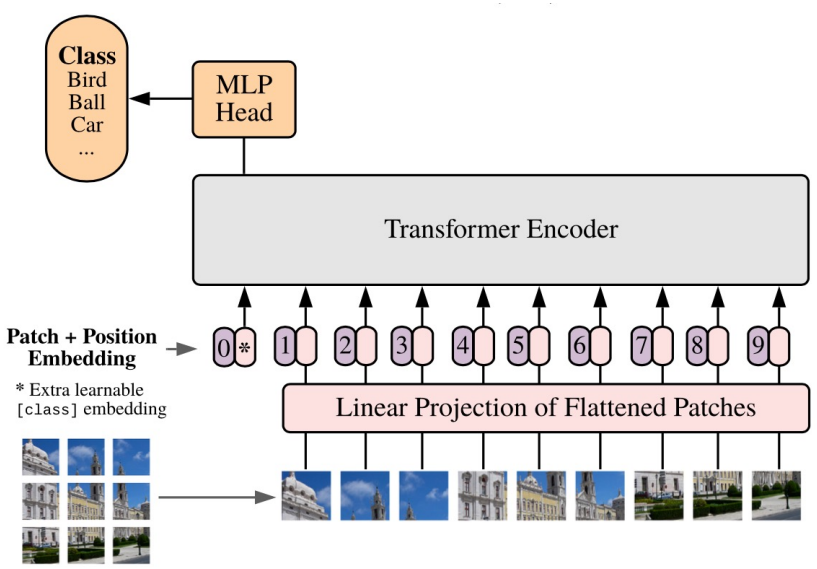
  [2] Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

  – 画像パッチのシーケンスに対して標準のTransformerを直接適用した手法

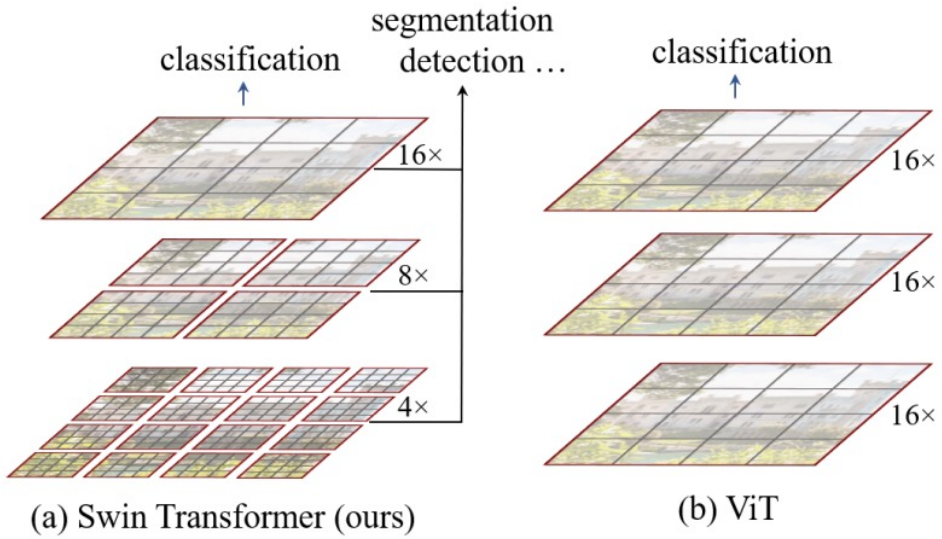
- **Swin Transformer**

  [3] Liu et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. CVPR 2021.

  – ViTの問題点の，物体検出の分解能が限られる点と入力パッチ数が膨大になる点を解決した手法



▲ ViT



▲ Swin Transformer

3. 関連研究 ー Vision Transformer における Continual Learning ー

- **DyTox**

  [19] Arthur et al. Dytox: Transformers for continual learning with dynamic token expansion. CVPR 2022.

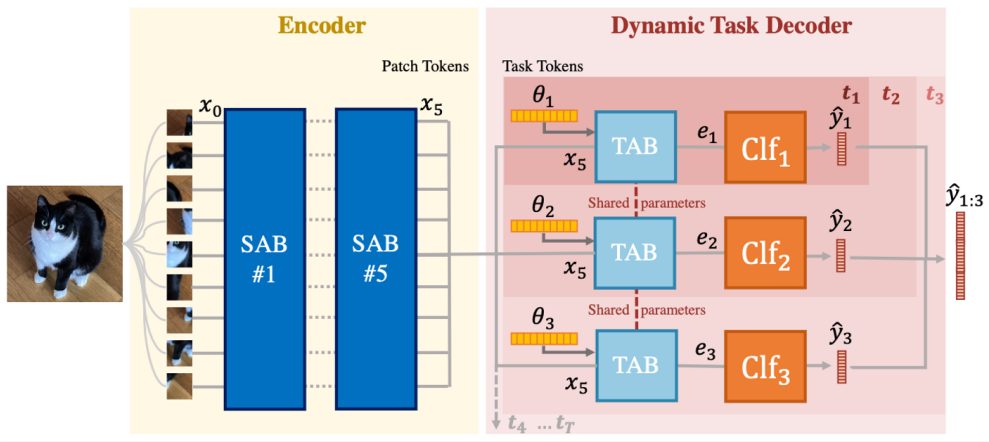
  – タスク固有のトークンを使用し，タスクに特化した埋め込みを生成する

- **Learning to Prompt for Continual Learning (L2P)**

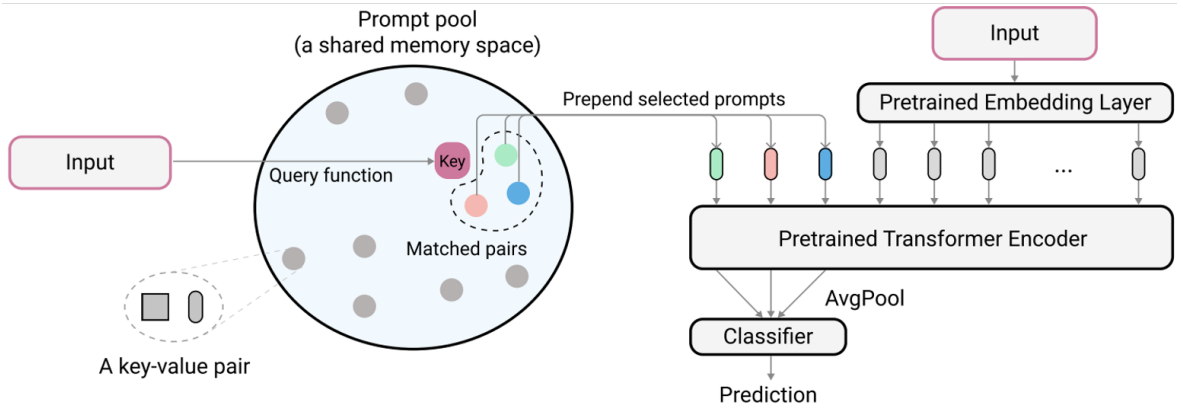
  [20] Zifeng et al. Learning to prompt for continual learning. arXiv:2112.08654, 2021.

  – 自然言語処理分野におけるプロンプト学習を応用した手法

- これらの手法は，クラスインクリメンタル手法のため比較できない



▲ DyTox



▲ L2P

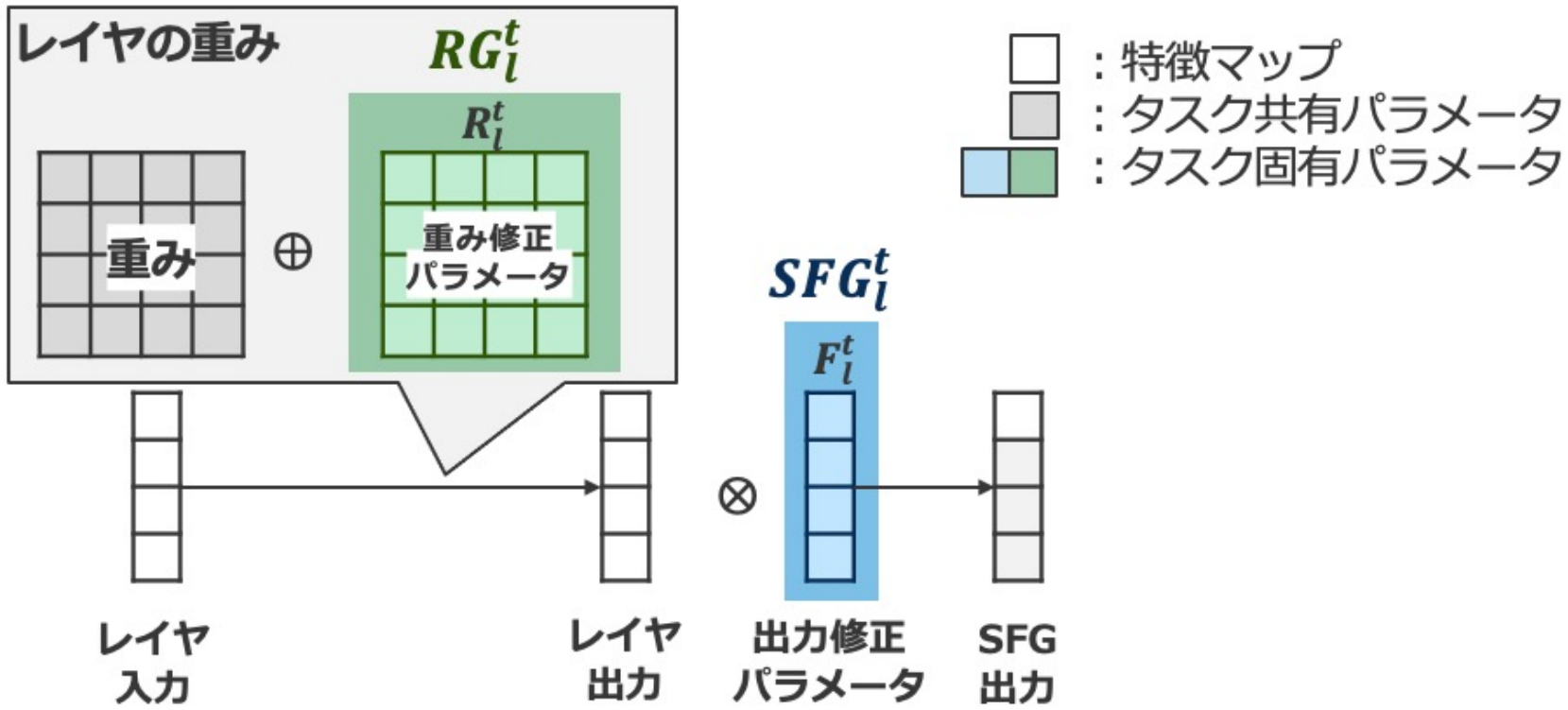
4. 手法　ー手法概要ー

- 本研究では，タスクインクリメンタルな Continual Learning を行う手法として **Mask-RKR** を提案する

- **Mask-RKR** は，
  - ベースとなる Rectification-based Knowledge Retention（RKR）に対して Piggyback を適用する
  - 主な特徴
    - RKRによるタスクへの適応
    - Piggybackによるパラメータ削減

4. 手法 ー RKRによるタスクへの適応 ー

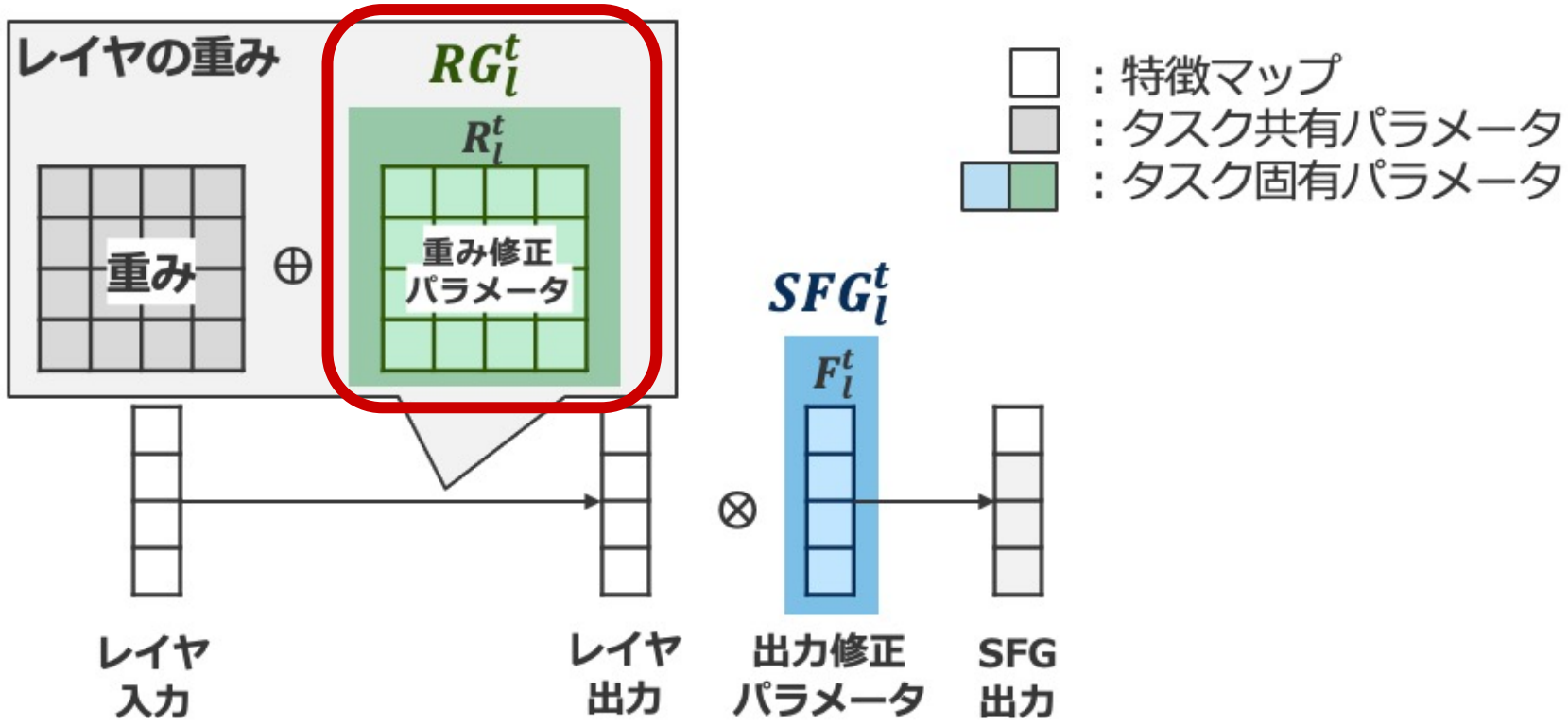
- Mask-RKR ではベースに RKR を用いることで，ネットワークを各タスクに適応させる
- RKR では，**R**ectification **G**enerator (**RG**) と **S**caling **F**actor **G**enerator (**SFG**) という二つのジェネレータを使用してネットワークの重みと中間出力を修正する



4. 手法 ー RKRによるタスクへの適応 ー

RGの概要(1/2)

- RG では，大規模データセットで事前学習済の各タスク各レイヤの重みに，タスクとレイヤ固有の重み修正パラメータが加算される

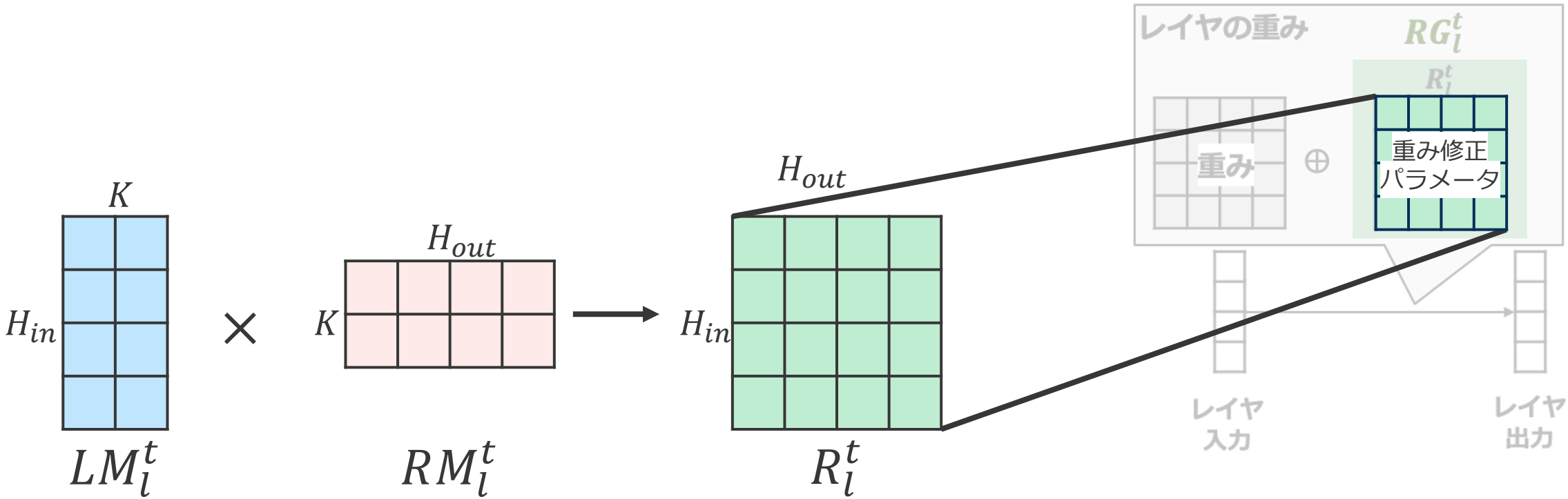


4. 手法 ー RKRによるタスクへの適応 ー

RGの概要(2/2)

- **低ランク近似**でパラメータ削減を行う

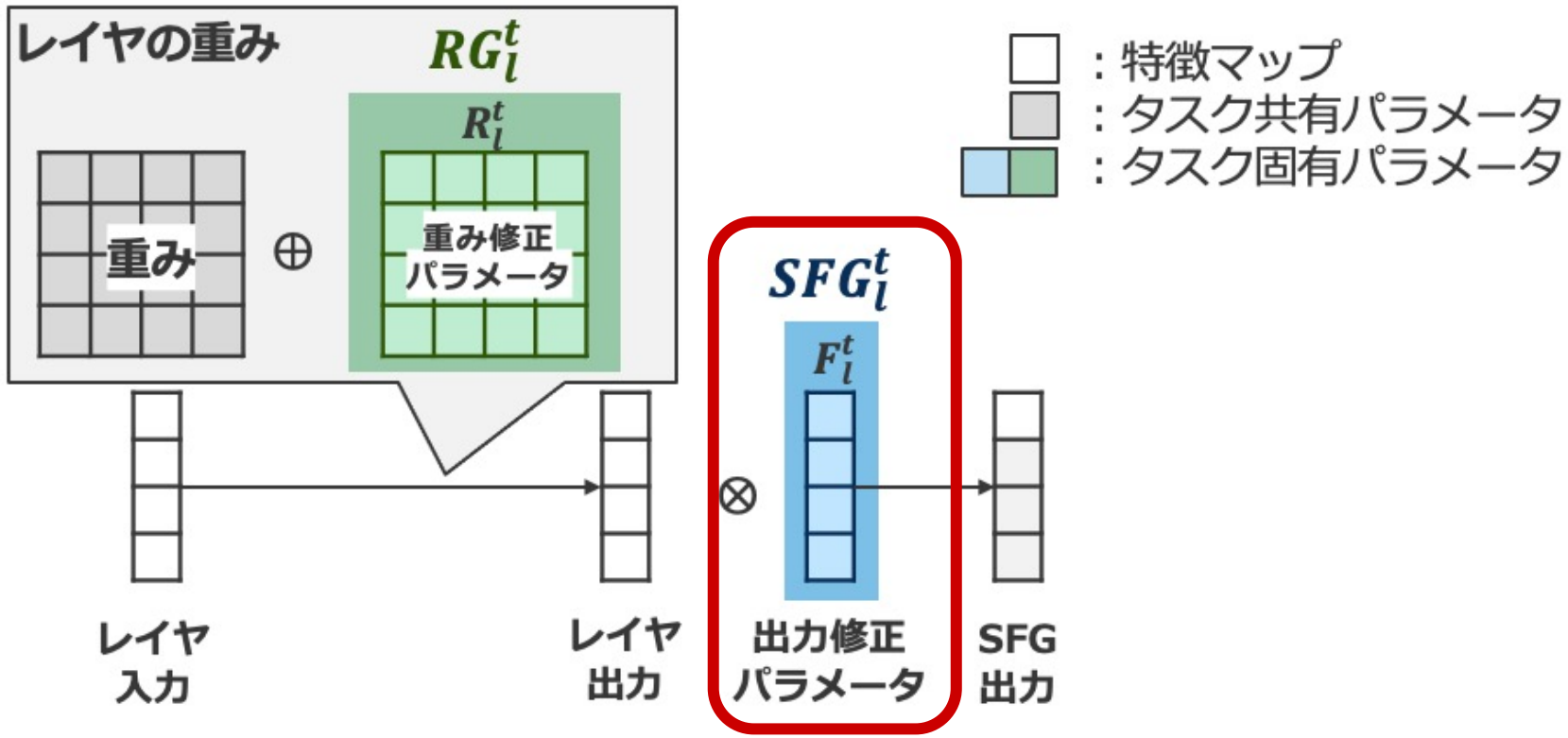
- サイズが小さい2つの行列LMとRMを学習し，これらの積により重み修正のパラメータを生成する



4. 手法 ー RKRによるタスクへの適応 ー

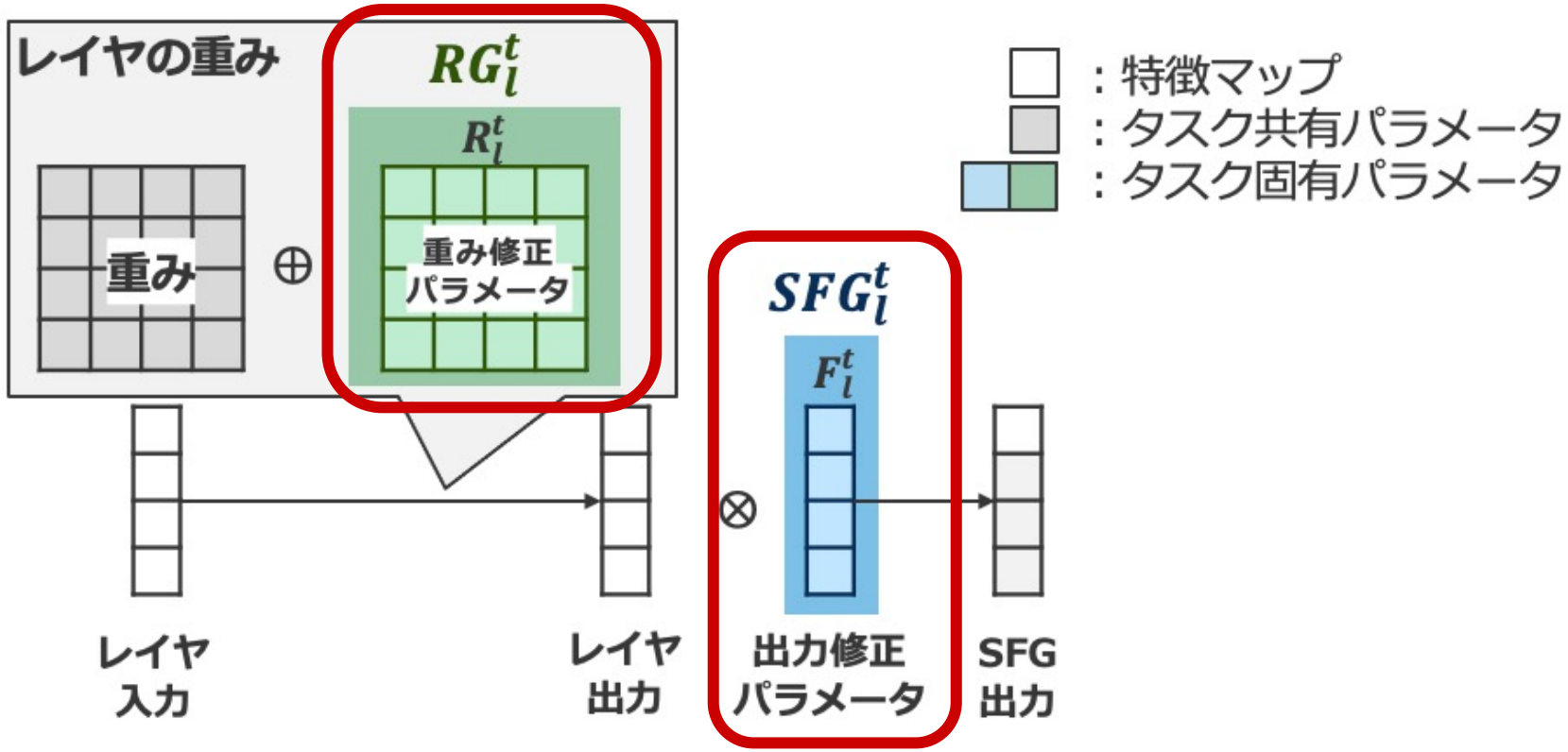
SFGの概要

- SFGでは，各タスク各レイヤの中間出力に，各タスクと各レイヤ固有の中間出力修正パラメータが乗算される



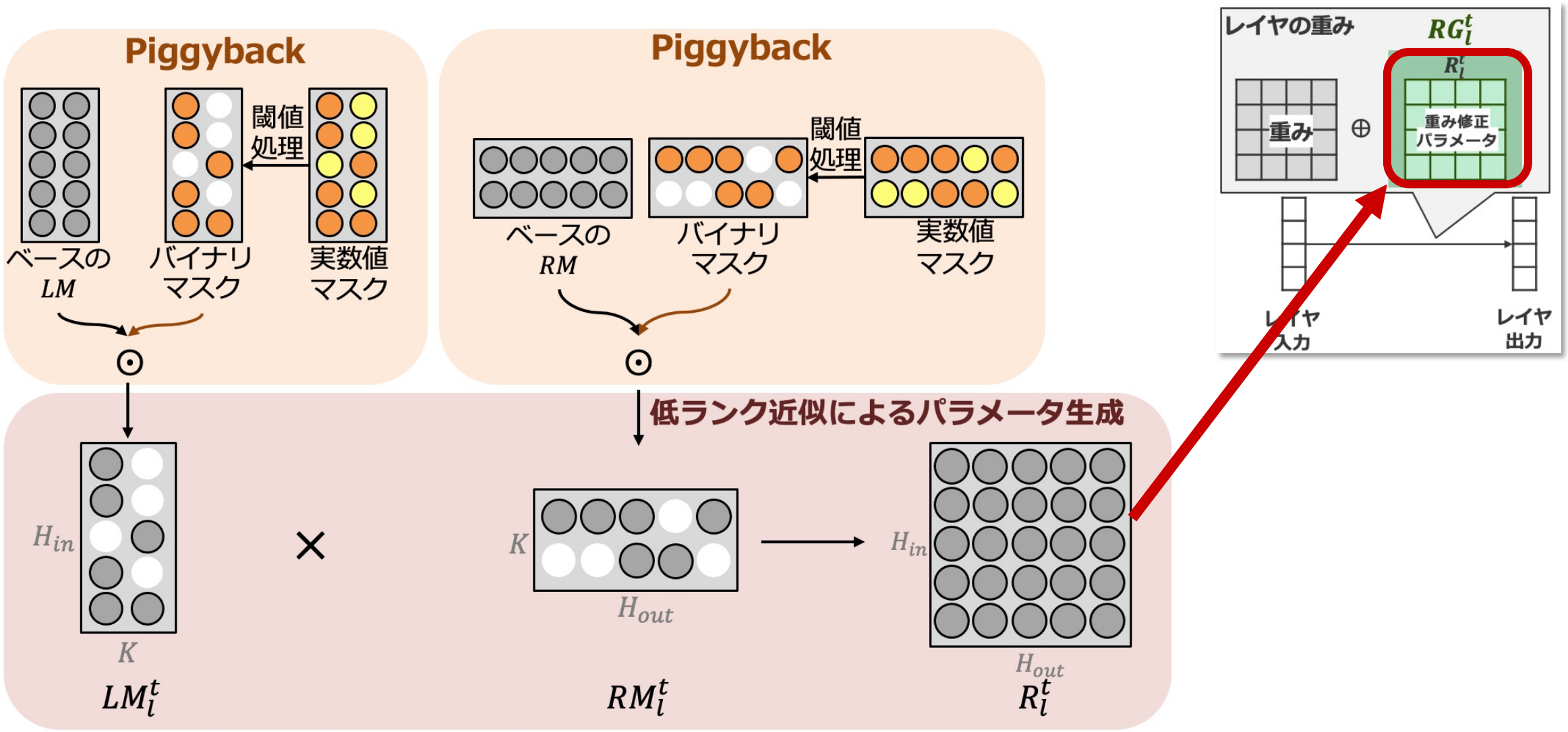
4. 手法 ー Piggyback によるパラメータ削減 ー

- Piggyback は，ベースの重みに対して学習した重みマスクを適用して出力を変換する
- Mask-RKR は，RKR のパラメータに対して Piggyback を適用することでさらなるパラメータ数の削減を行う



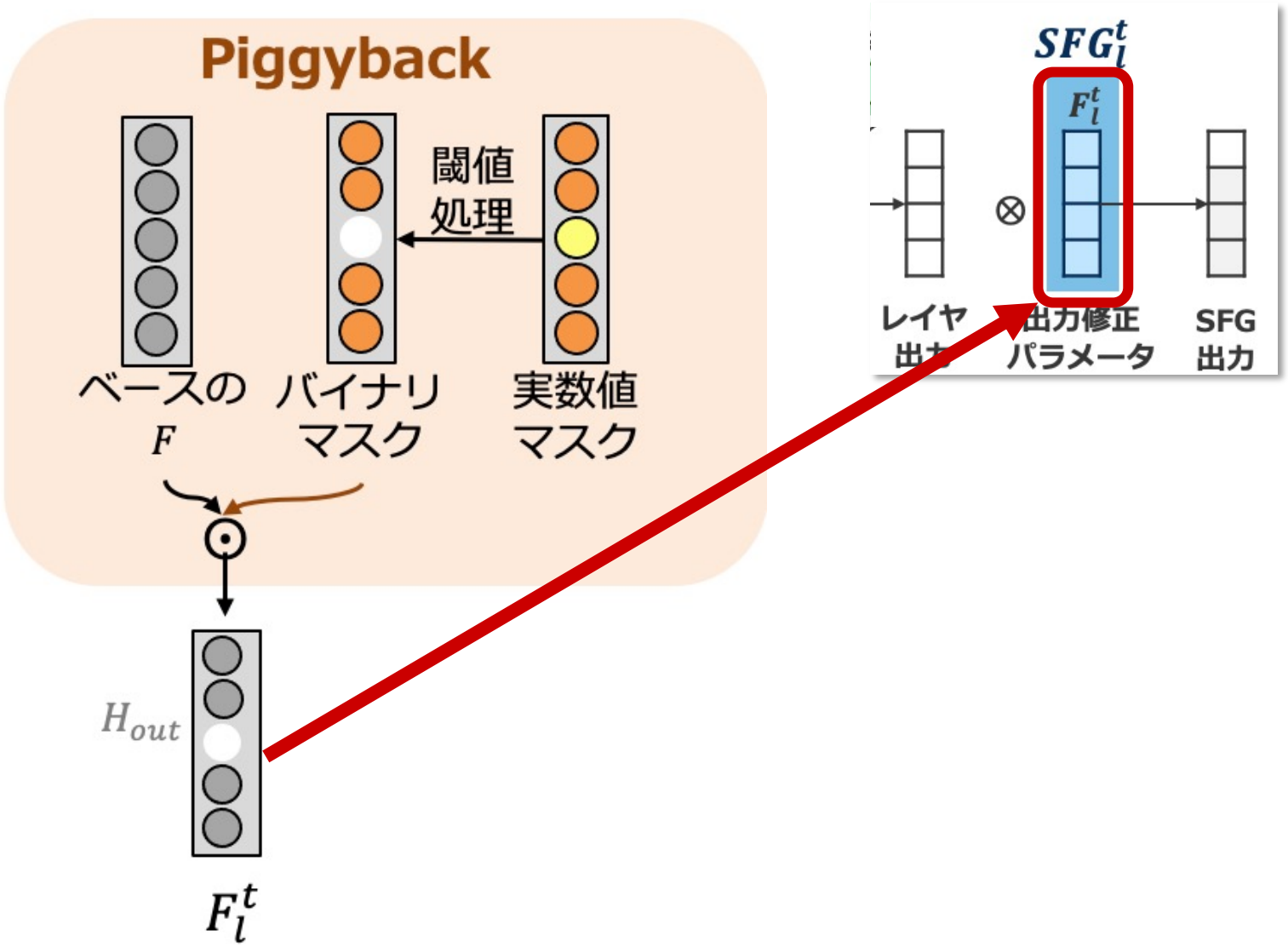
4. 手法 ー Piggybackによるパラメータ削減 ー

RGでのパラメータ削減



4. 手法 ー Piggybackによるパラメータ削減 ー

SFGでのパラメータ削減



5. ベースラインとの比較実験 ー実験概要 ー

- Mask-RKRの性能を検証するために3つのContinual Learning設定で実験を行った

- モデル
  - ResNet-18，ViT，Swin Transformer

- ベースライン
  - **Single**：各タスクを固有のモデルで学習
  - **Multi Head**：最終出力層のみをタスクごとに入れ替え
  - **RKR(K=2)**：タスクごとにネットワークの重みと中間出力を修正する手法
  - **Piggyback**：学習した重みマスクを適用して出力を変換する手法

  - Ours
    - **Ours(K=2)**：提案手法のMask-RKR
    - **Ours K+**：Kの値を調整して，「RKR」と同じパラメータ数にしたMask-RKR

5. ベースラインとの比較実験 ー 実験1： CIFAR-100 を用いた実験ー

- 動植物や機器，乗り物など100クラスが含まれる**CIFAR-100**を使用
  - 10クラスを持つ10タスクに分割したものを順に学習
    （タスク1 → タスク2 → ・・・ → タスク10）

手法＼モデル	平均精度			パラメータ数[M]		
	ResNet-18	ViT	Swin	ResNet-18	ViT	Swin
			0.876	111.72 (+900.00%)	856.59 (+900.00%)	11.98 (+900.00%)
			0.768	11.22 (+0.41%)	85.73 (+0.08%)	1.22 (+1.45%)
RKR(	0.794	0.843	0.858	11.74 (+5.05%)	89.88 (+4.92%)	1.43 (+19.72%)
Piggyback	**0.804**	0.838	**0.875**	14.71 (+31.65%)	112.27 (+31.07%)	1.56 (+30.29%)
Ours(K=2)	↓ 0.781	↓ 0.840	↓ 0.841	↓ 11.28 (+1.01%)	↓ 86.26 (+0.70%)	↓ 1.24 (+3.79%)
Ours K+	↑ 0.796	↑ **0.845**	↑ 0.858	↓ 11.74 (+5.05%)	↓ 89.87 (+4.92%)	↓ 1.43 (+19.56%)

パラメータの増加を抑えつつ，
高い精度を達成

5. ベースラインとの比較実験 ー 実験2： ImageNet-1k を用いた実験ー

- 1000クラスを持つ大規模データセットである**ImageNet-1k**を使用
  - 100クラスを持つ10タスクに分割したものを順に学習
    （タスク1 → タスク2 → ・・・ → タスク10）

手法＼モデル	平均精度			パラメータ数[M]		
	ResNet-18	ViT	Swin	ResNet-18	ViT	Swin
			0.902	112.18 (+900.00%)	858.76 (+900.00%)	868.46 (+900.00%)
			0.887	11.68 (+4.12%)	86.57 (+0.81%)	87.77 (+1.06%)
RKR(	0.545	**0.885**	0.892	12.20 (+8.73%)	90.71 (+5.64%)	92.34 (+6.33%)
Piggyback	0.440	0.881	0.805	15.17 (+35.22%)	113.11 (+31.71%)	113.94 (+31.20%)
Ours(K=2)	0.557	0.879	0.870	11.75 (+4.71%)	87.10 (+1.42%)	88.35 (+1.74%)
Ours K+	↑ **0.582**	↑ **0.885**	↑ **0.894**	↓ 12.43 (+10.83%)	↓ 90.71 (+5.63%)	↓ 92.30 (+6.28%)

パラメータの増加を抑えつつ，
高い精度を達成

5. ベースラインとの比較実験 ー 実験3： 異なるドメインのデータセットを用いた実験ー

- 異なるドメインのデータセットを使用
  - 5タスクを順に学習

（D. Textures → GTSRB → SVHN → UCF101 → VGG-Flower）

手法＼モデル	平均精度			パラメータ数[M]		
	ResNet-18	ViT	Swin	ResNet-18	ViT	Swin
[illegible]	[illegible]	[illegible]	0.842	111.91 (+900.00%)	857.39 (+900.00%)	594.62 (+900.00%)
[illegible]	[illegible]	[illegible]	0.682	11.32 (+1.17%)	85.89 (+0.18%)	59.59 (+0.22%)
RKR([illegible]	0.714	0.791	**0.840**	11.58 (+3.49%)	87.97 (+2.60%)	61.49 (+3.41%)
Piggyback	**0.723**	**0.809**	0.839	13.07 (+16.76%)	99.16 (+15.66%)	68.75 (+15.62%)
Ours(K=2)	0.695	0.775	0.824	11.38 (+1.71%)	86.36 (+0.72%)	60.02 (+0.94%)
Ours(K+)	↓ 0.720	↓ 0.778	↓ 0.831	↓ 11.52 (+2.95%)	↓ 87.67 (+2.25%)	↓ 61.39 (+3.24%)

パラメータの増加は抑えるが，

精度が低下

6. アブレーション実験 ー マスクの有用性の検証 ー

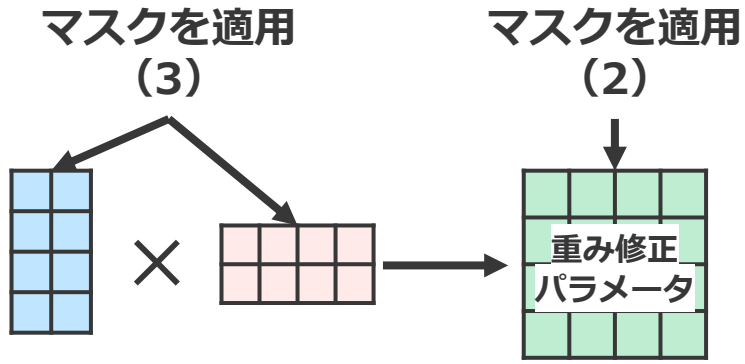
- RGとSFGのそれぞれにマスクを適用する/しない場合を比較し，有用性を検証した
  - 「RG w/ Mask」：RGにマスクを適用する
  - 「SFG w/ Mask」：SFGにマスクを適用する

- 本実験では，最もパラメータ数を抑える**RGとSFGにPiggybackを適用したモデル**を使用

RG w/ Mask	SFG w/ Mask	平均精度			パラメータ数[M]		
		ResNet-18	ViT	Swin	ResNet-18	ViT	Swin
x	x	**0.794**	0.843	**0.858**	11.74 (+5.05%)	89.88 (+4.92%)	1.43 (+19.72%)
✓	x	0.780	<u>0.844</u>	<u>0.846</u>	11.33 (+1.38%)	87.07 (+1.64%)	1.28 (+6.59%)
x	✓	**0.794**	**0.845**	**0.858**	11.69 (+4.68%)	89.15 (+4.08%)	1.40 (+17.20%)
✓	✓	<u>0.781</u>	0.840	0.841	⬇ 11.28 (+1.01%)	⬇ 86.26 (+0.70%)	⬇ 1.24 (+3.79%)

6. アブレーション実験 ーPiggybackの適用場所の検証 ー

- RGにおけるマスクの適用場所を検証した
  - （1）適用しない
  - （2）重み修正パラメータに適用する
  - （3）低ランク近似されたパラメータに適用する（Mask-RKR）



- パラメータ数を抑えるには，**LMとRMそれぞれにPiggybackを適用**した方が効果的

手法	平均精度			パラメータ数[M]		
	ResNet-18	ViT	Swin	ResNet-18	ViT	Swin
(1)	0.794	**0.845**	**0.858**	11.69 (+4.68%)	89.15 (+4.08%)	1.40 (+17.20%)
(2)	➡ **0.805**	➡ **0.845**	➡ 0.847	⬆ 14.41 (+29.00%)	⬆ 110.05 (+28.48%)	⬆ 1.55 (+29.32%)
(3)	0.781	0.840	0.841	⬇ 11.28 (+1.01%)	⬇ 86.26 (+0.70%)	⬇ 1.24 (+3.79%)

7. まとめ

- CNN と Vision Transformer の両方へ適用可能なContinual Learning手法である Mask-RKRを提案した
- 実験から，従来手法と比べて，パラメータ数の増加を最小限に抑えつつ高い精度を達成できることを示した
- RKRよりもパラメータ数を抑えることができ，パラメータ数に制限を持つ現実社会の問題に対して，より幅広く対応できる汎用性を持つといえる

8. 今後の課題

- 異なるドメインのデータセットを用いたContinual Learningにも対応できる柔軟性を持たせる
- 画像生成におけるContinual Learningにも適用できるかを検証する


UEC
TOKYO

3. 関連研究 ー Continual Learning ー

- Continual Learning では，過去に学習したタスクの知識を保持しながら，新しいタスクを継続的に学習する
  - **クラスインクリメンタル**：新たなクラスが追加される
  - **タスクインクリメンタル**：新たなタスクが追加される