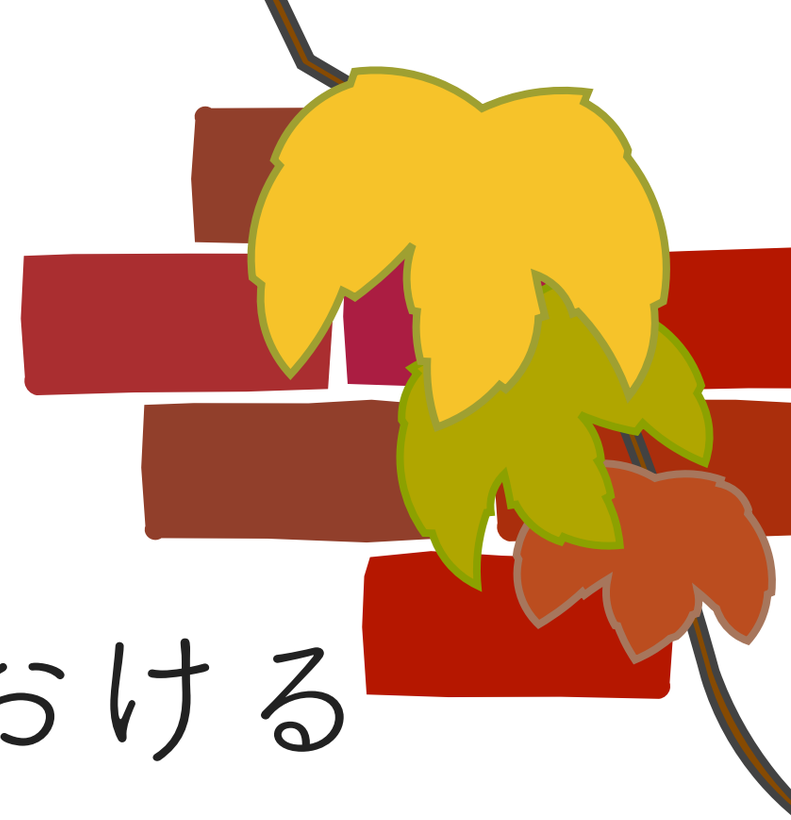


Computer Visionにおける TransformerとMLP-Mixerの現状

立教大学 人工知能科学研究科

瀧 雅人

今日のメッセージ

ニューラルネット・深層学習の研究を牽引してきた畳み込みネットワークは、**必ずしも唯一のモデルではない。**

Plan

- CNNと帰納バイアス（10min）
- Visual Transformer（20min）
- MLP-Mixer（20min）
- Conclusion & Discussion（5min）

1. CNNと帰納バイアス

深層学習の能力と汎用性

柔軟なアーキテクチャ

適用先のデータ・ドメインに関する帰納バイアス・事前知識を、ネットワークデザインを通じて自在に反映

柔軟な訓練プロトコル

シンプルな勾配法で訓練できるため、工夫を加えやすい

implicitな正則化効果

表現能力の高い大規模なモデルにもかかわらず、過学習が抑制

機械学習＝帰納推論 (inductive reasoning)

観察した**有限個**の事例 → **普遍**法則

汎化generalization

フランシス・ベーコン

source : https://commons.wikimedia.org/wiki/File:Francis_Bacon,_Viscount_St_Alban_from_NPG_(2).jpg

機械学習＝帰納推論 (inductive reasoning)

フランシス・ベーコン

観察した有限個の事例 → 普遍法則

汎化generalization

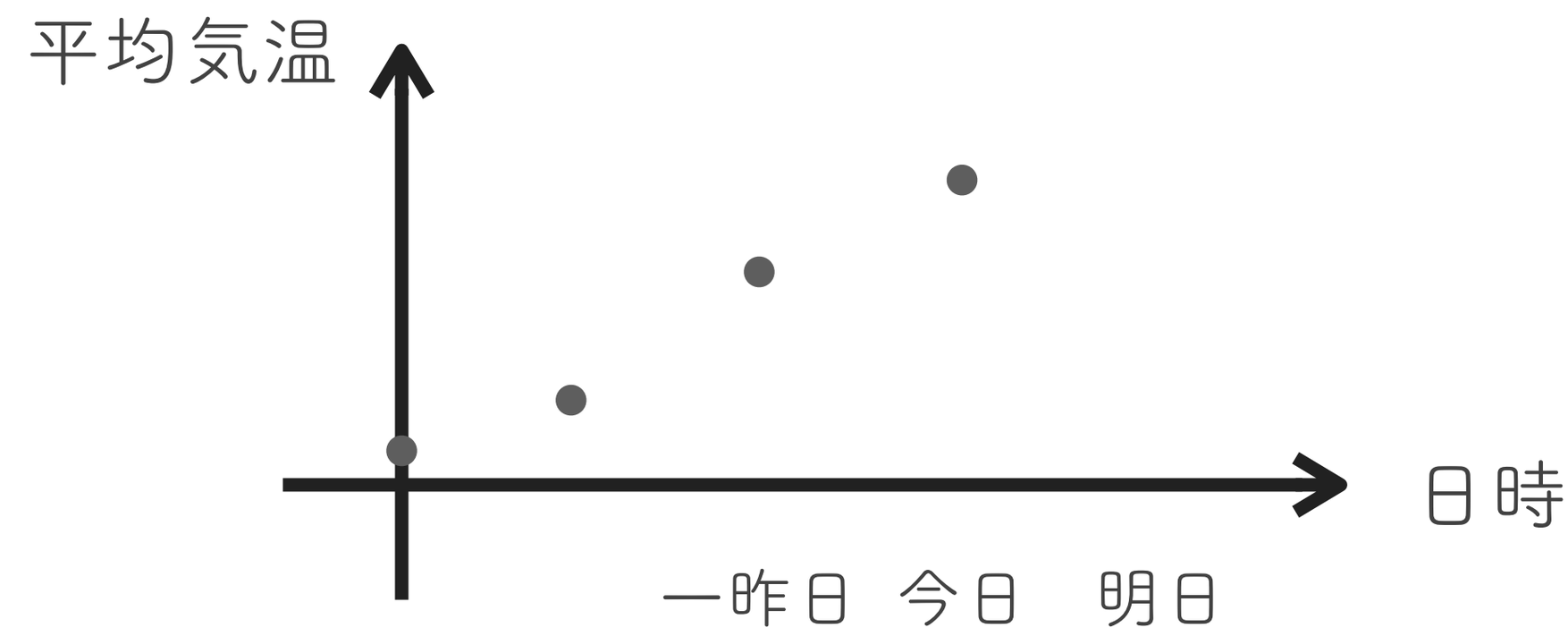
帰納は合理的**正当化が不可能**

ヒュームの懐疑

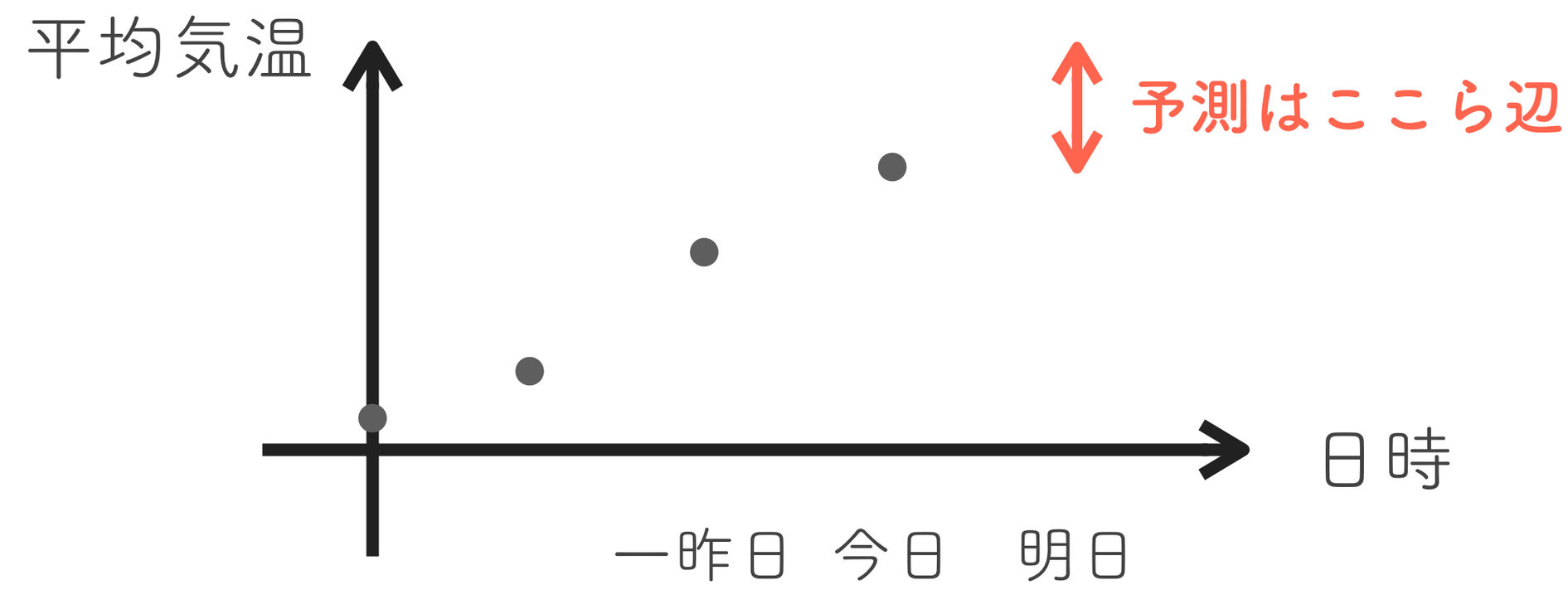
デイヴィット・ヒューム

source : https://commons.wikimedia.org/wiki/File:David_Hume.jpg

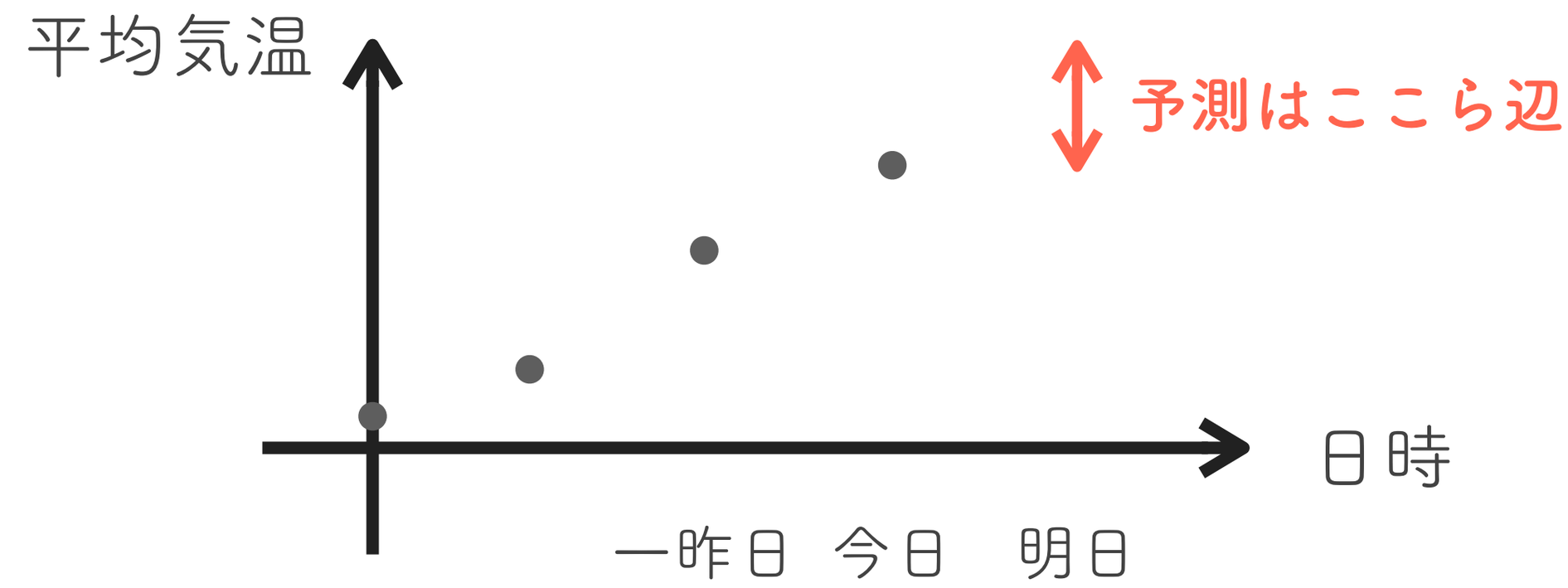
機械学習＝帰納推論 (inductive reasoning)



機械学習＝帰納推論 (inductive reasoning)



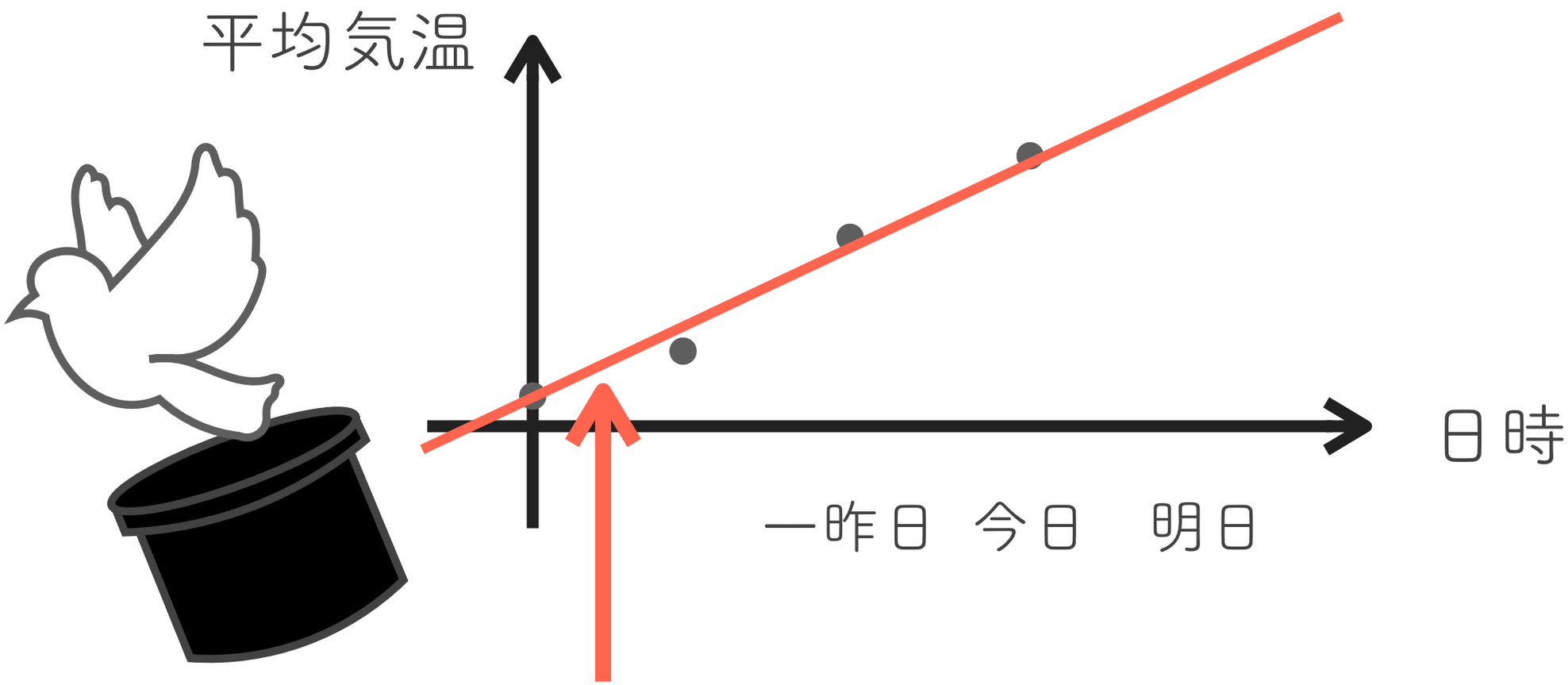
機械学習＝帰納推論 (inductive reasoning)



斉一性・単純性(剃刀)などを仮定して初めて予測できる

一切の仮定がないと、数学的・論理的には丸暗記した四日分の気温を答える事しかできない

機械学習＝帰納推論 (inductive reasoning)



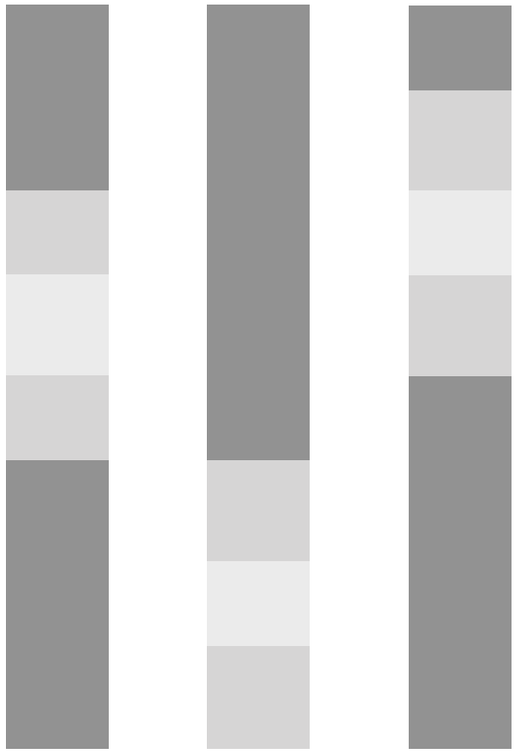
帰納バイアス：データ以外のモデル・訓練アルゴリズムへの仮定（結果へのバイアス）。これがなければ汎化などしない

CNNの帰納バイアス

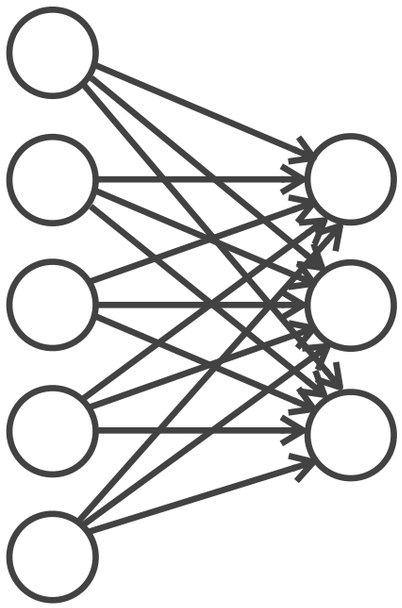
画像のパターン認識：パターンの局所性、並進不変性、パターンの階層性、…

CNNの帰納バイアス

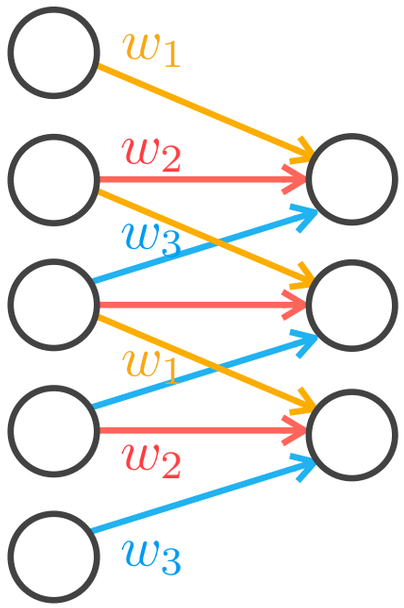
1次元画像の例：現れる局所パターンの並進不変性



vanilaニューラルネット



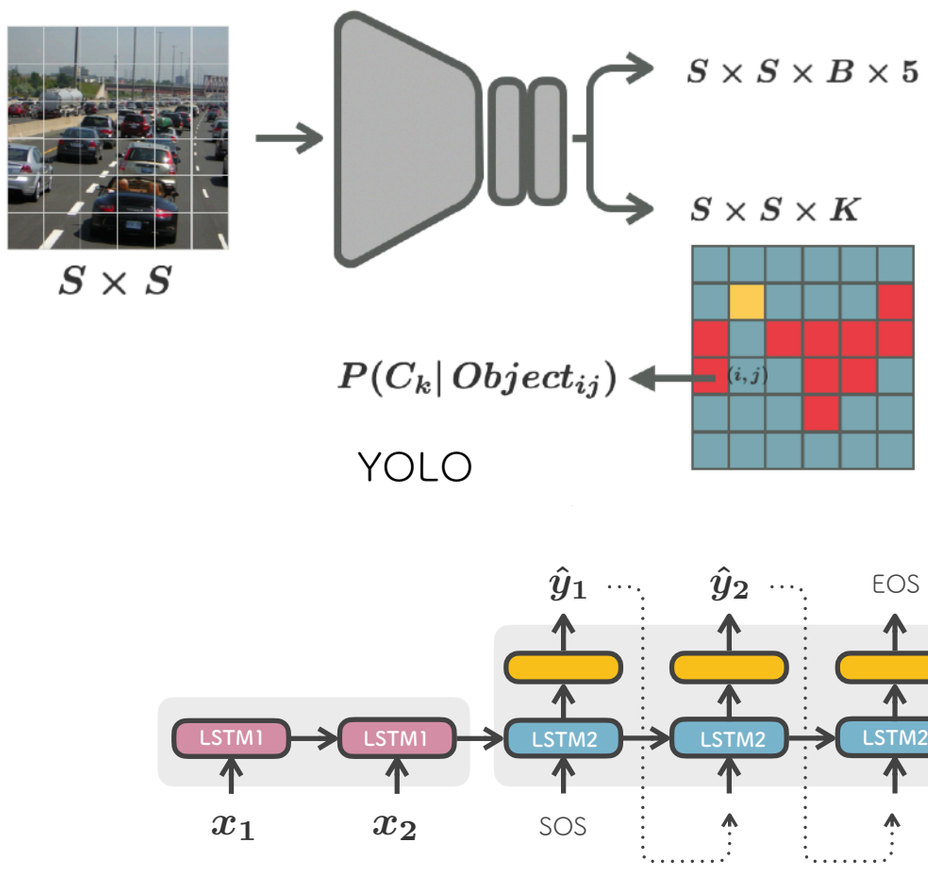
畳み込みニューラルネット

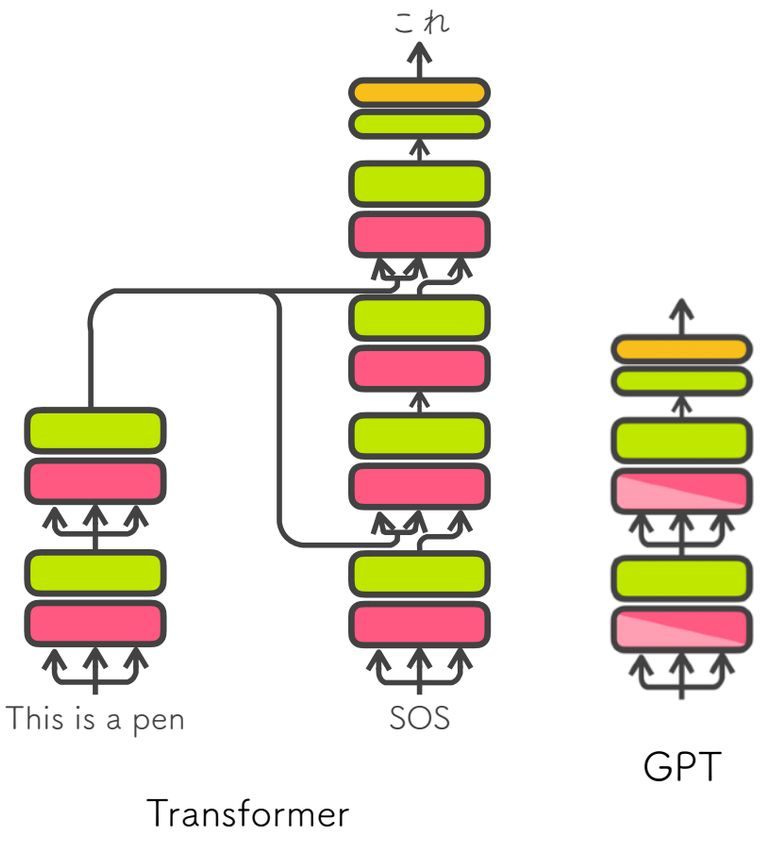


→ CNNはhard codeされた強い帰納バイアスを持つ

→ 高い汎化能力

深層学習の帰納バイアスさまざま





→ さまざまなタスクで汎化させるために、さまざまなアーキテクチャ≒帰納バイアスが開発されてきた

CNNの帰納バイアス

これまではCNNがSOTAを更新してきた：

→ 長距離の依存関係、globalな情報がそこまでうまく捉えられない（CNNの帰納バイアスの代償）

畳み込みは帰納バイアスがハードコードされすぎていて、実は限界がある？？（→ ViT）

2. Vision Transformer

自然言語処理モデルにおける進展：Transformer

Transformer

[Vaswani et al., Attention is All You Need, 2017]

自己アテンション(self-attention)により、コンテキスト下のトークンの埋め込み表現を学習。長距離依存関係まで学習可能。

→ 飛躍的な性能向上が可能になっただけではなく、超巨大なデータで訓練した超巨大なモデルが続々と誕生（後述）

database = {'apple':1.5, 'berry': 1.2, 'cherry':3.1}

↓

database['apple'] = ?

自然言語処理モデルにおける進展：Transformer

Transformer

[Vaswani et al., Attention is All You Need, 2017]

自己アテンション(self-attention)により、コンテキスト下のトークンの埋め込み表現を学習。長距離依存関係まで学習可能。

→ 飛躍的な性能向上が可能になっただけではなく、超巨大なデータで訓練した超巨大なモデルが続々と誕生（後述）

database = {'apple':1.5, 'berry': 1.2, 'cherry':3.1}

↓

database['apple'] = 1.5

クエリ(query)に関してデータベース内を検索

自然言語処理モデルにおける進展：Transformer

Transformer

[Vaswani et al., Attention is All You Need, 2017]

自己アテンション(self-attention)により、コンテキスト下のトークンの埋め込み表現を学習。長距離依存関係まで学習可能。

→ 飛躍的な性能向上が可能になっただけではなく、超巨大なデータで訓練した超巨大なモデルが続々と誕生（後述）

database = {'apple':1.5, 'berry': 1.2, 'cherry':3.1}

↓

database['strawberry'] = ?

自然言語処理モデルにおける進展：Transformer

Transformer

[Vaswani et al., Attention is All You Need, 2017]

自己アテンション(self-attention)により、コンテキスト下のトークンの埋め込み表現を学習。長距離依存関係まで学習可能。

→ 飛躍的な性能向上が可能になっただけではなく、超巨大なデータで訓練した超巨大なモデルが続々と誕生（後述）

database = {'apple':1.5, 'berry': 1.2, 'cherry':3.1}

↓

0.008　0.982　0.01　クエリにとってのキーの重要度

database['strawberry'] = ?

自然言語処理モデルにおける進展：Transformer

Transformer

[Vaswani et al., Attention is All You Need, 2017]

自己アテンション(self-attention)により、コンテキスト下のトークンの埋め込み表現を学習。長距離依存関係まで学習可能。

→ 飛躍的な性能向上が可能になっただけではなく、超巨大なデータで訓練した超巨大なモデルが続々と誕生（後述）

database = {'apple':1.5, 'berry': 1.2, 'cherry':3.1}

↓

0.008　0.982　0.01　クエリにとってのキーの重要度

database['strawberry'] = 0.008*1.5 + 0.982*1.2 + 0.01*3.1

= 1.2214　token mixing

自然言語処理モデルにおける進展：Transformer

Transformer

[Vaswani et al., Attention is All You Need, 2017]

自己アテンション(self-attention)により、コンテキスト下のトークンの埋め込み表現を学習。長距離依存関係まで学習可能。

→ 飛躍的な性能向上が可能になっただけではなく、超巨大なデータで訓練した超巨大なモデルが続々と誕生（後述）

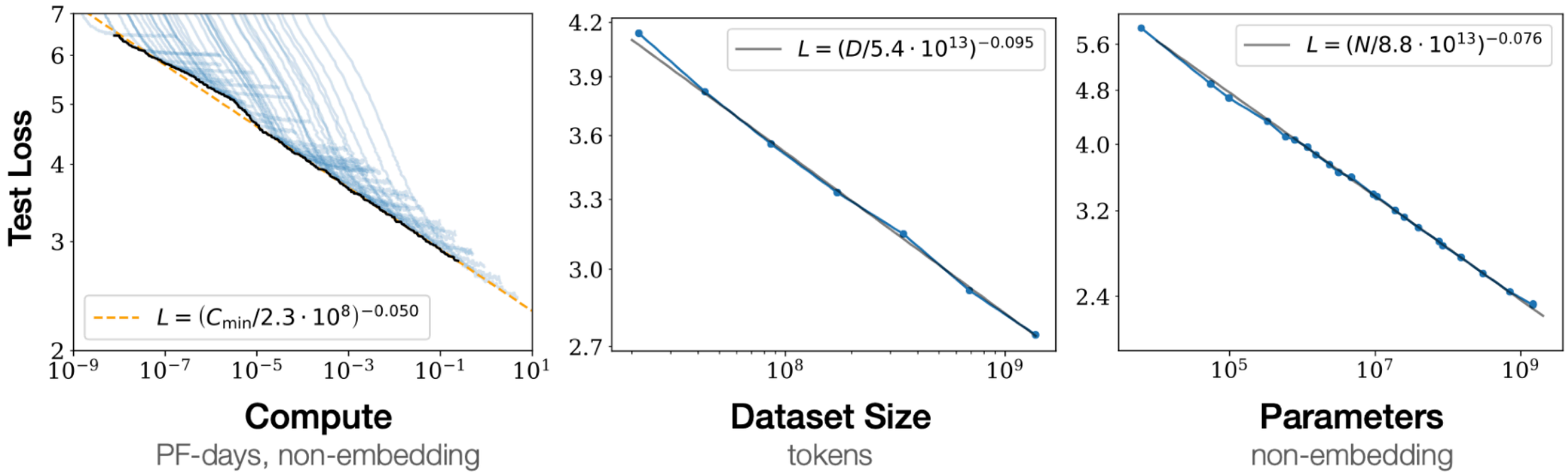
database = {'This':1.5, 'is': 1.2, 'a':-1.3, 'pen':3.1}

↓ このコンテキスト下でのクエリにとってのキーの重要度

database['pen'] = 1.8*1.5 + 0.2*1.2 + 0.001*-1.3 + 1.2*3.1

≠ 3.1　コンテキストを取り込んだ埋め込み表現

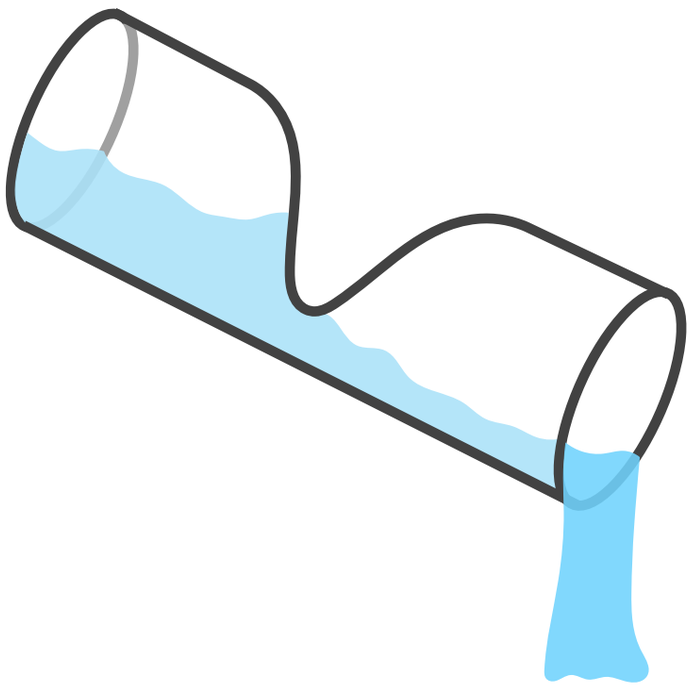
Transformer（GPT）のスケーリング則



$\mathbf{1\,PF\text{-}day = 10^{15} \times 24 \times 60 \times 60 = 8.64 \times 10^{19}}$不動小数点演算

WebText2 datasetで訓練

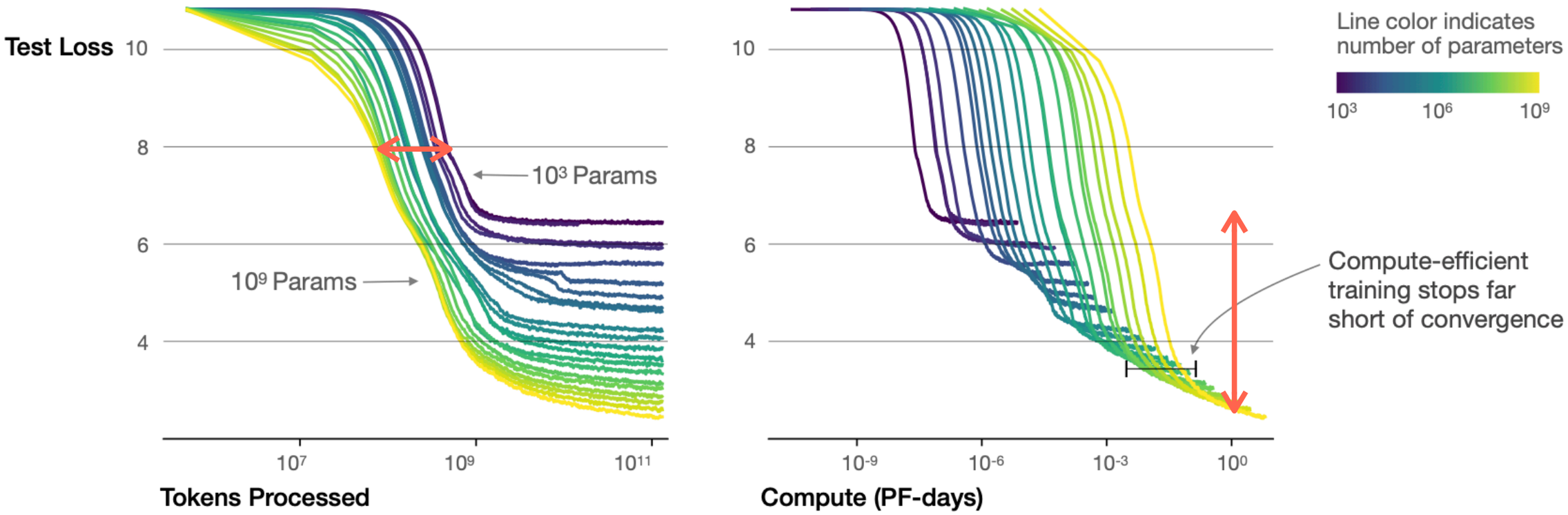
[OpenAI, Scaling Laws for Neural Language Models, 2020]



（深さや幅のような詳細にはあまり依存せず）３スケールさえ増やせばボトルネックなくスケールしてゆく！まだ飽和は見られない。

→ **まだまだTransformerでいける！**投資の効果もスケーリング則で予測できる！

Transformer（GPT）のスケーリング則



[OpenAI, Scaling Laws for Neural Language Models, 2020]

データ効率：大きいモデルの方が少ないサンプル(と少ないステップ)で同じ性能に到達する

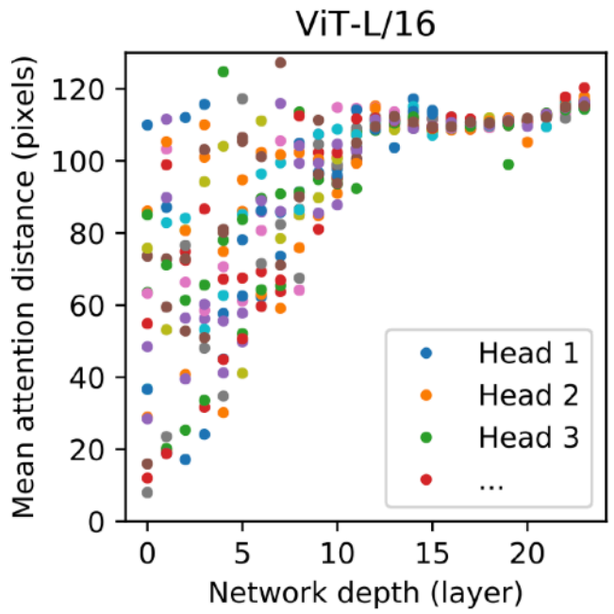
コンピュータビジョンへの影響

Vision Transformer

[Google Research, Brain Team, AN IMAGE IS WORTH 16X16 WORDS, 2020]

CNNのハードコードされた帰納バイアスは不要。より弱く柔軟な帰納バイアス(Transformer)でもSOTA画像認識モデルができる。

→ 大規模なデータでは、必要な帰納バイアスはTransformerがデータから学びとる。CNNにはないグローバルなパターンの学習も。



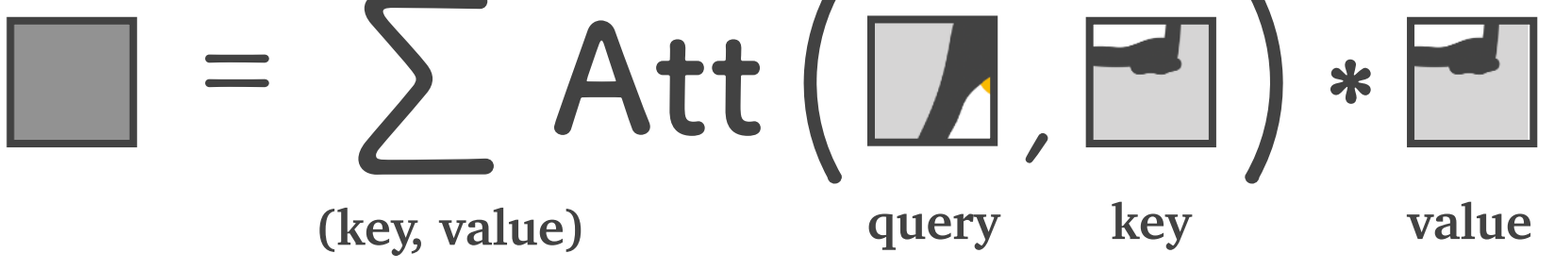
Vision Transformer (ViT)

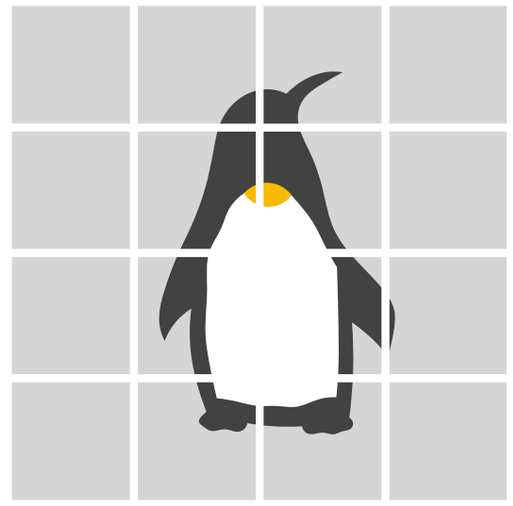
単語と異なり、ピクセルは明確な意味を運ばない。

→ **パッチをトークンにする**（若干の二次元情報）

Vision Transformer (ViT)

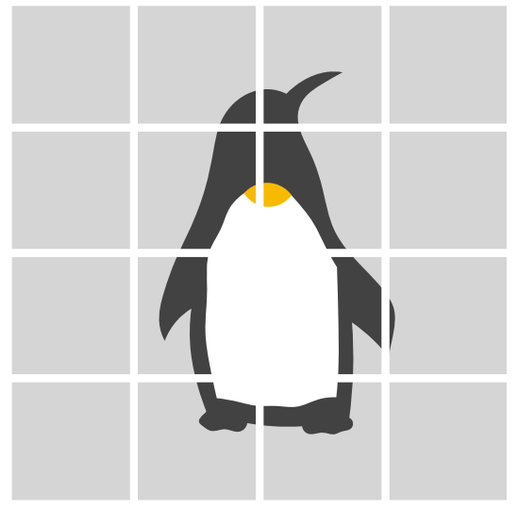
画像の各パッチの埋め込み表現ベクトルを、周辺のコンテキストを自己アテンションで取り込むことで学習

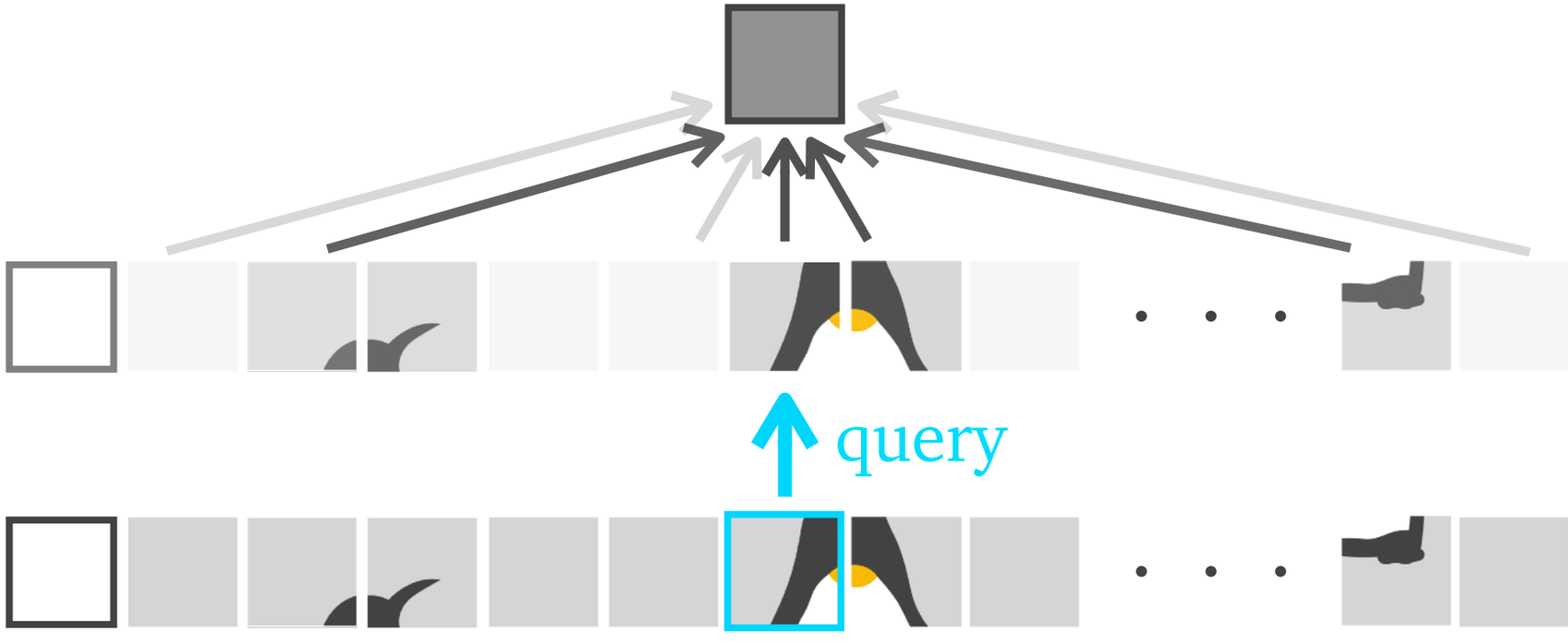




Vision Transformer (ViT)

画像の各パッチの埋め込み表現ベクトルを、周辺のコンテキストを自己アテンションで取り込むことで学習





Vision Transformer (ViT)

「ImageNetのような**中規模の**データセットで、強力な正則化を行わずに学習した場合、これらのモデルは、同等のサイズのResNetsよりも数％ポイント低い控えめな精度を得ます。この一見がっかりするような結果は、予想されたものです。**トランスフォーマーには、CNN固有の帰納的バイアスがないため、十分な量のデータでトレーニングしてもうまく汎化できない**からです。」

「しかしより大規模なデータセット（**14M〜300M枚の画像**）でモデルを学習させると、状況は一変します。**大規模な学習が帰納的バイアスに勝ることがわかったのです**。・・・公開されているImageNet-21kデータセットまたはGoogle社内のJFT-300Mデータセットで事前学習を行うと、ViTは複数の画像認識ベンチマークにおいて最先端に近づくか、または凌駕します。」

[Google Research, Brain Team, AN IMAGE IS WORTH 16X16 WORDS, 2020]

ImageNet acc@1: 88.55%

ImageNet-ReaL acc@1: 90.72%

CIFAR-100 acc@1: 94.55%

Vision Transformer (ViT)：Cons & Pros

・大規模データでは、CNNを凌駕する性能

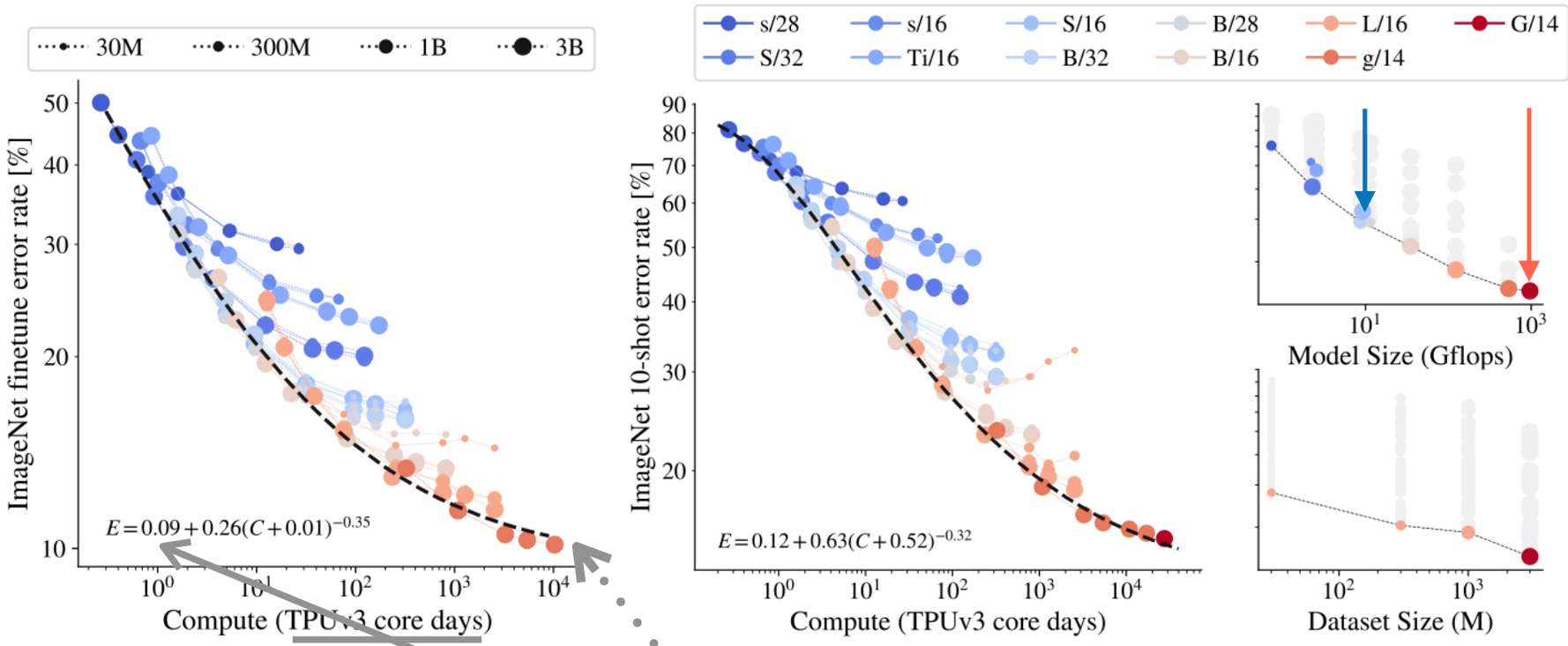
・転移学習で下流タスクでも良いモデルができる

・物体検知など、さまざまなタスクでも研究が進展

・過学習しやすいので、訓練プロトコルに注意が必要

・小・中規模データでスクラッチからの訓練が困難

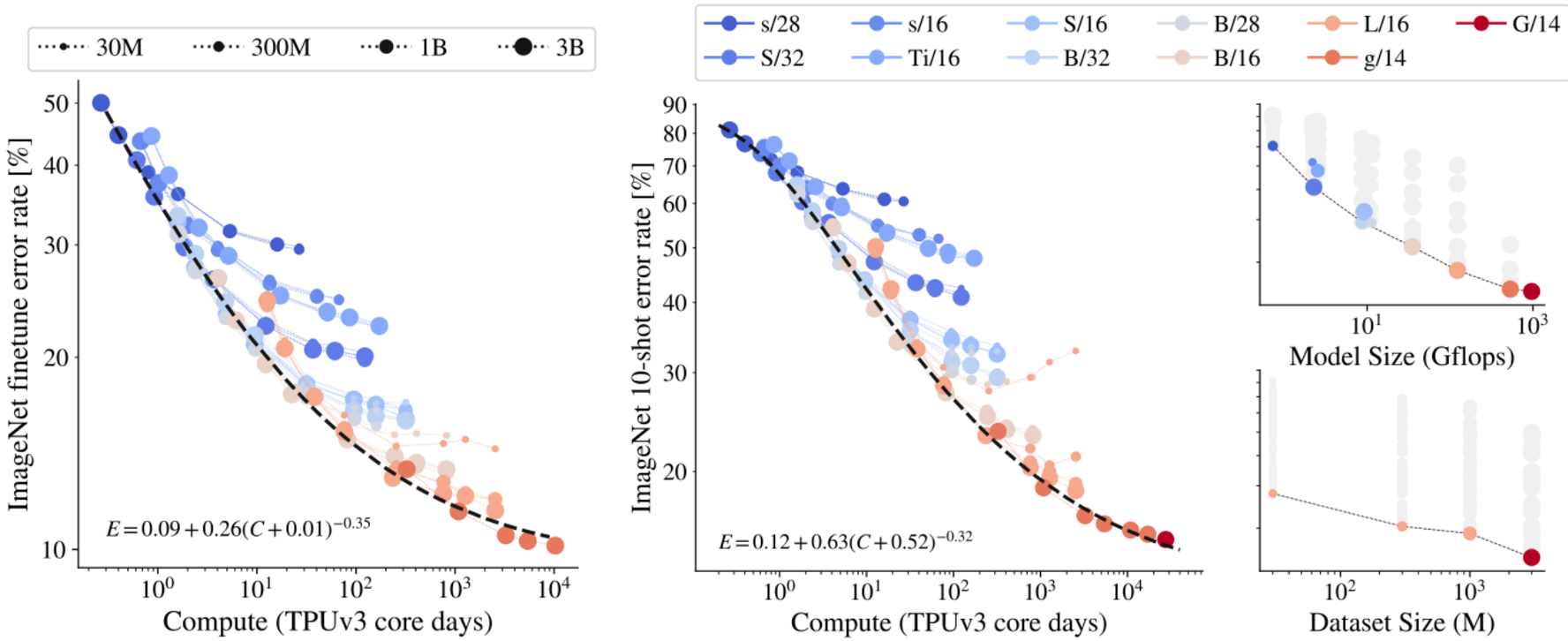
ViTのスケーリング則とその飽和



[Google Research, Brain Team, Zürich, Scaling Vision Transformers, 2021]

- 大きいモデル(赤)ほどパーレトフロンティア(破線)から離れない
- 「ベキ則の破れ＝ImageNet自体の**削減不能な誤差**」が見えている
- データ(モデルサイズ)と計算量を固定すると、モデルサイズ(データサイズ)がボトルネック

ViTのスケーリング則とその飽和



生成モデルの場合 **[OpenAI, Scaling Laws for Autoregressive Generative Modeling, 2020]**

$$\mathbb{E}_{x\sim P}\left[\log \frac{1}{Q(x)}\right] = \mathbb{E}_{x\sim P}\left[\log \frac{P(x)}{Q(x)}\right] + \mathbb{E}_{x\sim P}\left[\log \frac{1}{P(x)}\right]$$

クロスエントロピー損失
（ViTはエラー率なので違うが）

$$= D_{KL}\left(P \| Q\right) + S\left(P\right)$$

削減不能損失（データ分布P自身のエントロピー）

3. MLP-Mixer

2021年5月第一週のパラダイムシフト

Googleチームの発表を皮切りに、世界四カ所で独立に(?)行われていた研究が堰を切ったようにarXiv上に現れる

Google Research, Brain team 5月4日(火) 投稿

[Google Research, Brain Team, Mlp-mixer: An all-mlp architecture for vision. arXiv:2105.01601]

清華大学 5月5日(水) 投稿

[M.-H. Guo et al., Beyond self- attention: External attention using two linear layers for visual tasks. arXiv:2105.02358]

Oxford大学 5月6日(木) 投稿

[L. Melas-Kyriazi. Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet. arXiv:2105.02723]

Facebook AI, Sorbonne大学 etc. 5月7日(金) 投稿

[H. Touvron et al., Resmlp: Feedforward networks for image classification with data-efficient training. arXiv:2105.03404]

2021年5月第一週のパラダイムシフト

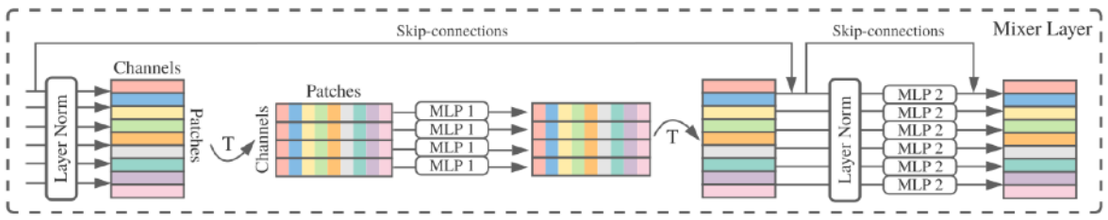
'In this paper we show that while convolutions and attention are both sufficient for good performance, neither of them are necessary.’ [arXiv:2105.01601]

2021年5月第一週のパラダイムシフト

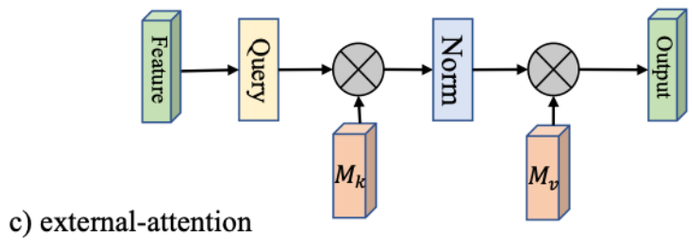
'In this paper we show that while convolutions and attention are both sufficient for good performance, neither of them are necessary.' [arXiv:2105.01601]

自己アテンションを単純なMLPの組み合わせに置き換え、SOTA CNNと競争力のある結果を得た。

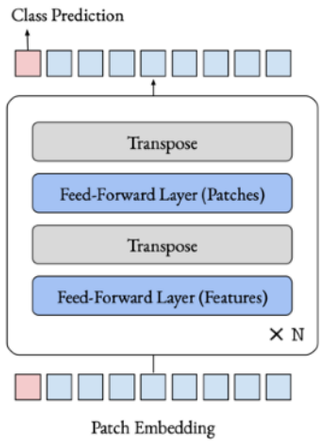
MLPの能力は、現代的な大規模データと計算資源において十分には再検討されていなかったとも言える。



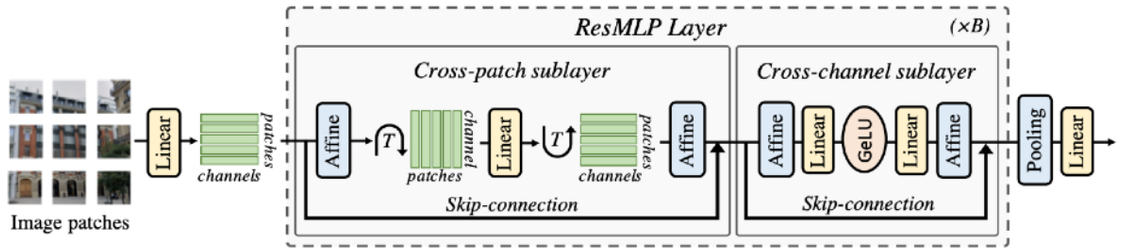
source : arXiv:2105.01601



source : arXiv:2105.02358

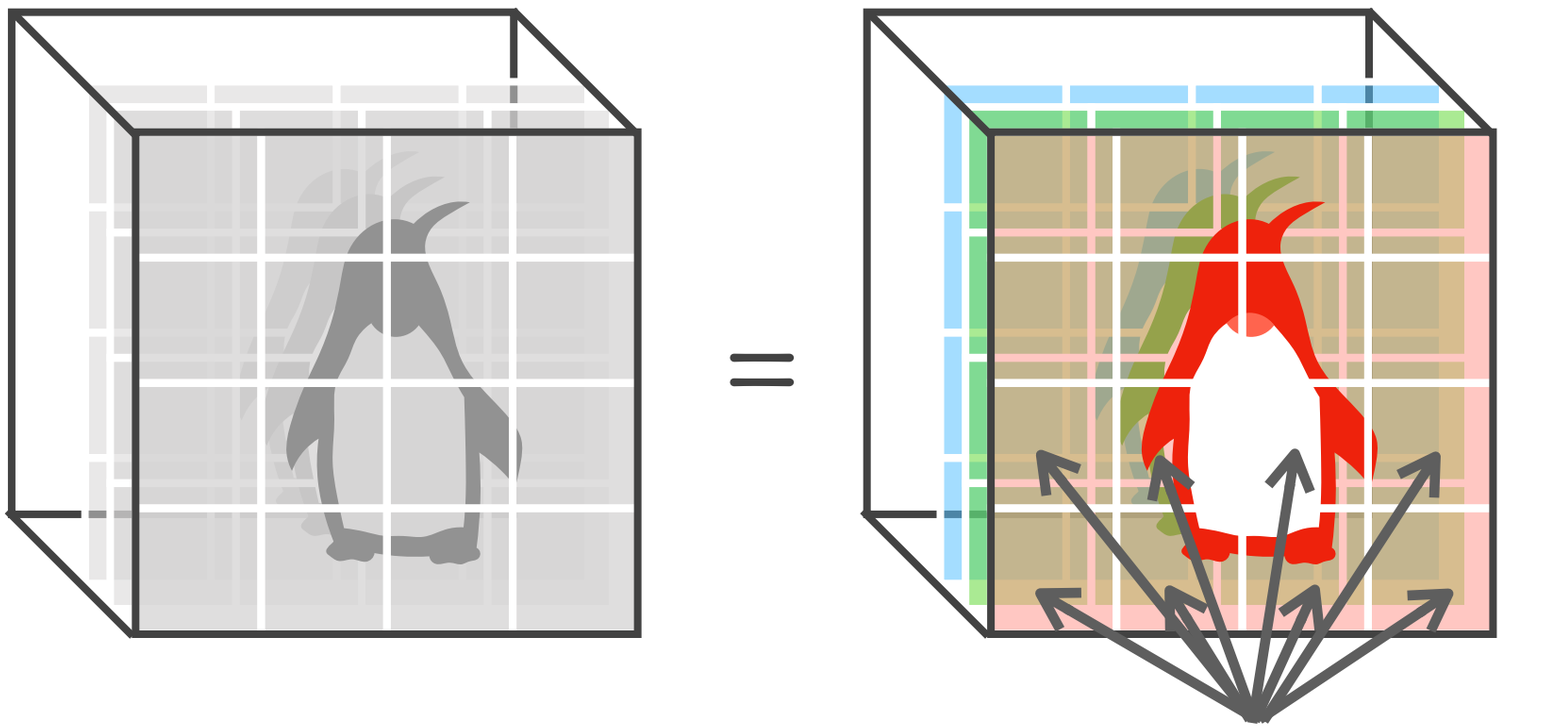


source : arXiv:2105.02723

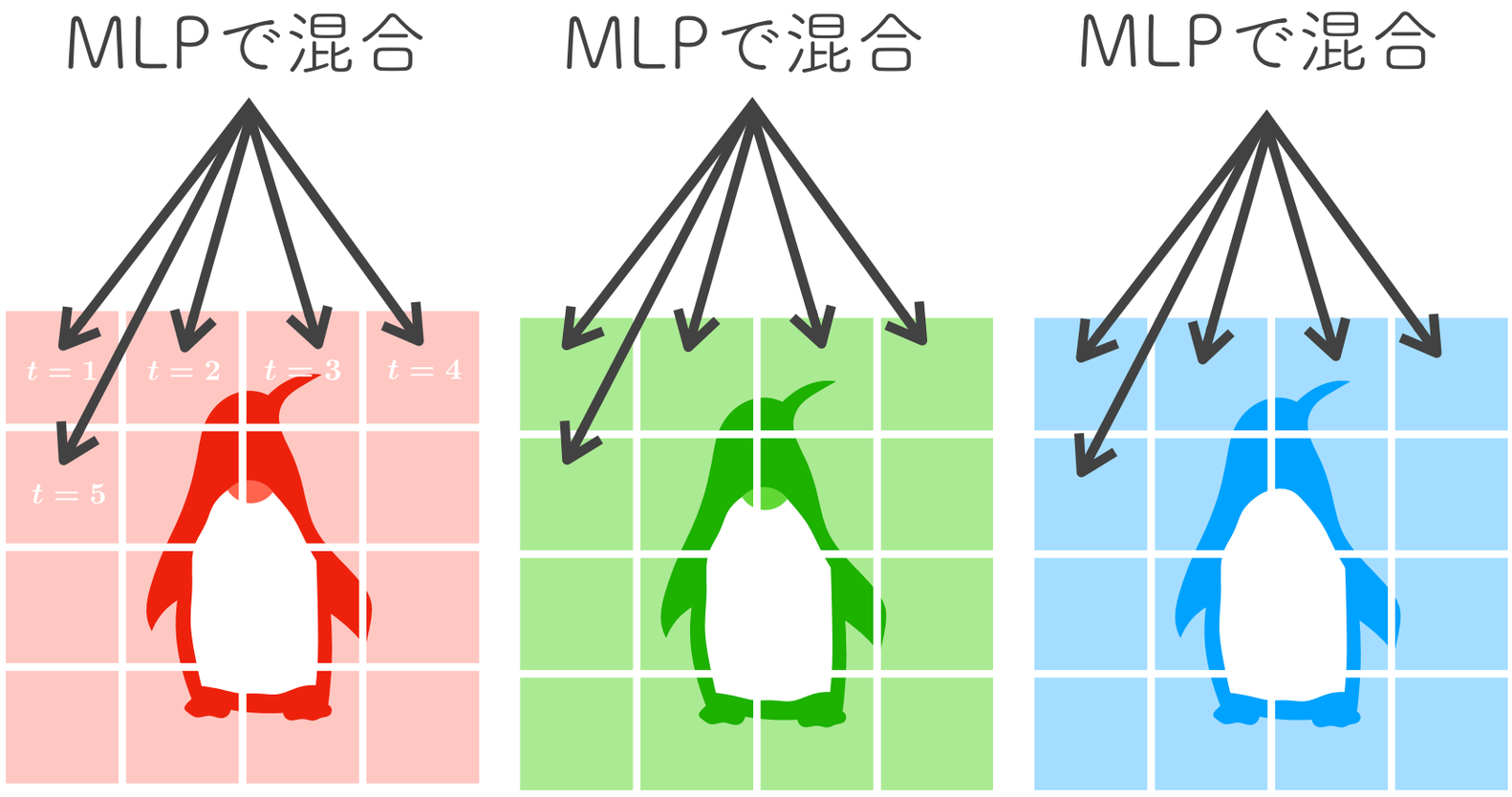


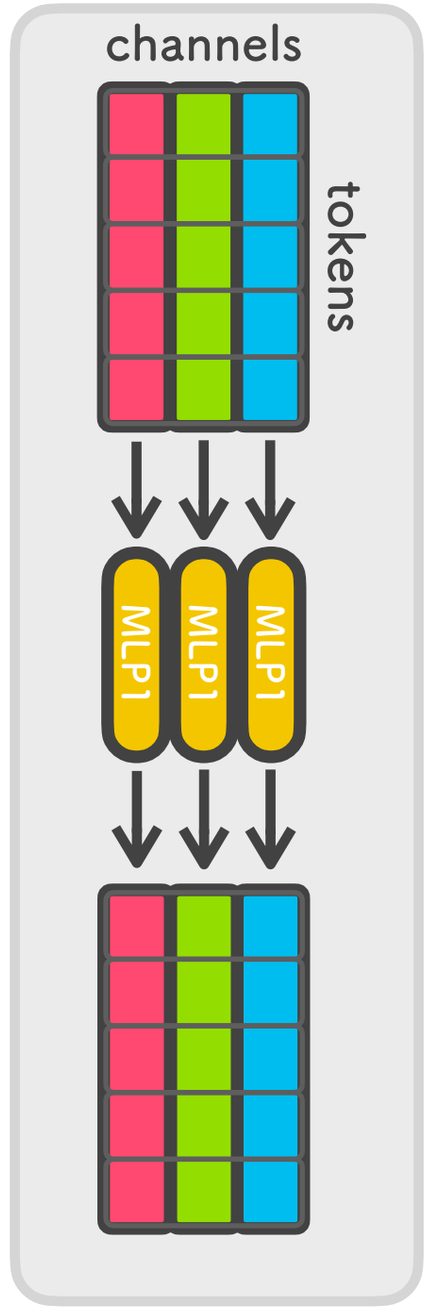
source : arXiv:2105.03404

MLP-Mixer



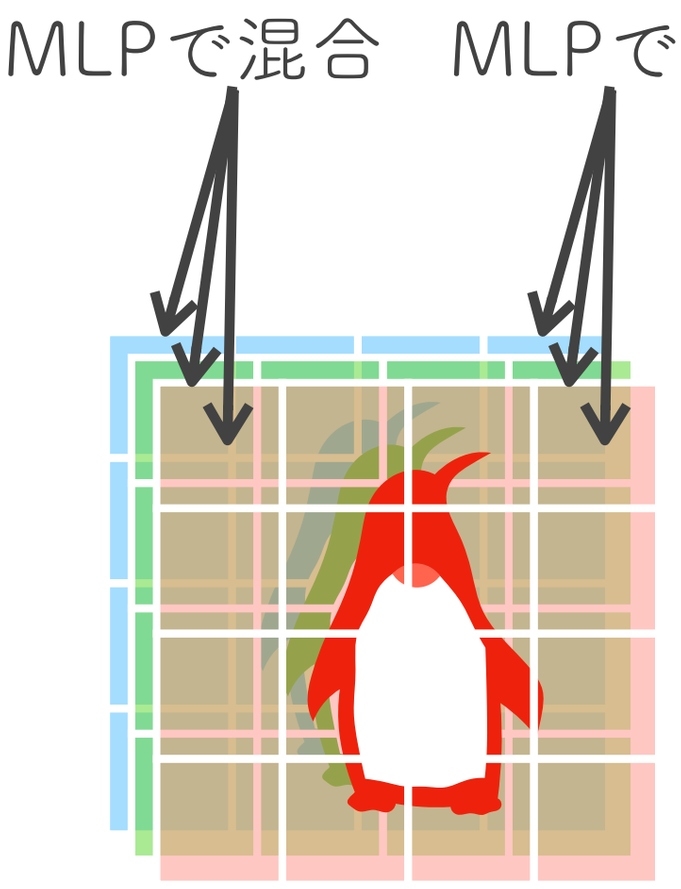
MLP-Mixer : Token Mixing（自己アテンションの代わり）

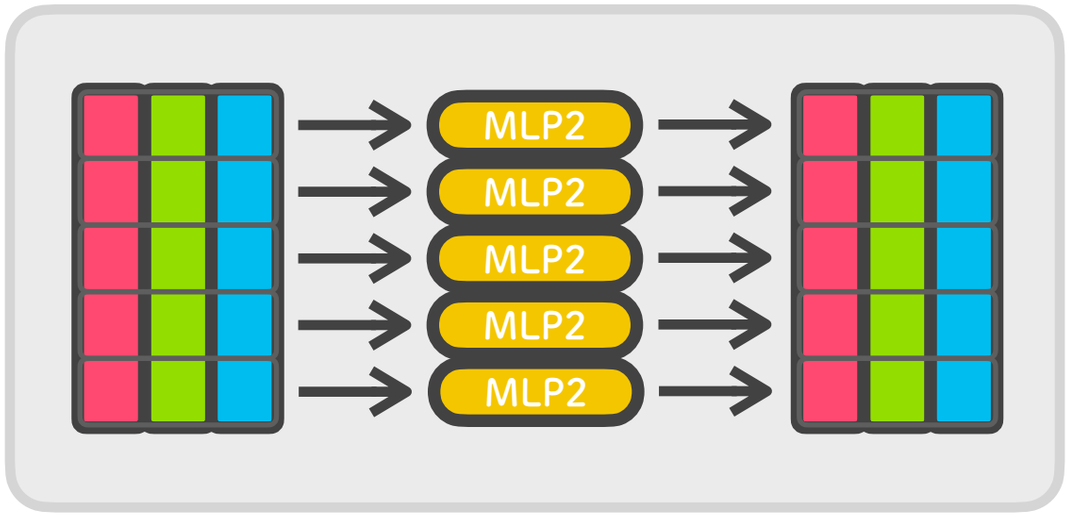




$$X'_{t'c} = \sum_t W^{(1)}_{t't} X_{tc}$$

MLP-Mixer : Channel Mixing





$$X'_{tc'} = \sum_t W^{(2)}_{c'c} X_{tc}$$

＊Transformerのpoint-wise fully-connectedそのもの

MLP-Mixer：擬似コード

```
class MLPBlock(layers.Layer):
    def __init__(self, mixing, d_ff):
        super(MLPBlock, self).__init__()
        if mixing != 'token' and mixing != 'channel':
            raise ValueError("undefiend mixing")
        self.mixing = mixing
```

⋮

```
    def call(self, inputs):
        '''
        inputsは(batches, tokens, channels)
        Denseは最後の軸に作用
        '''
        x = self.norm(inputs)
        x = x if self.mixing!='token' else tf.transpose(x, perm=[0, 2, 1])
        x =self.dense_1(x)
        x = tf.keras.activations.gelu(x, approximate=True)
        x = self.dense_2(x)
        x = x if self.mixing!='token' else tf.transpose(x, perm=[0, 2, 1])
        return x + inputs
```

MLP-Mixerの現状

5～9月の進展：清華大学チームのサーベイ論文

[R. Liu et al., ARE WE READY FOR A NEW PARADIGM SHIFT? A SURVEY ON VISUAL DEEP MLP. arXiv: 2111.04060]

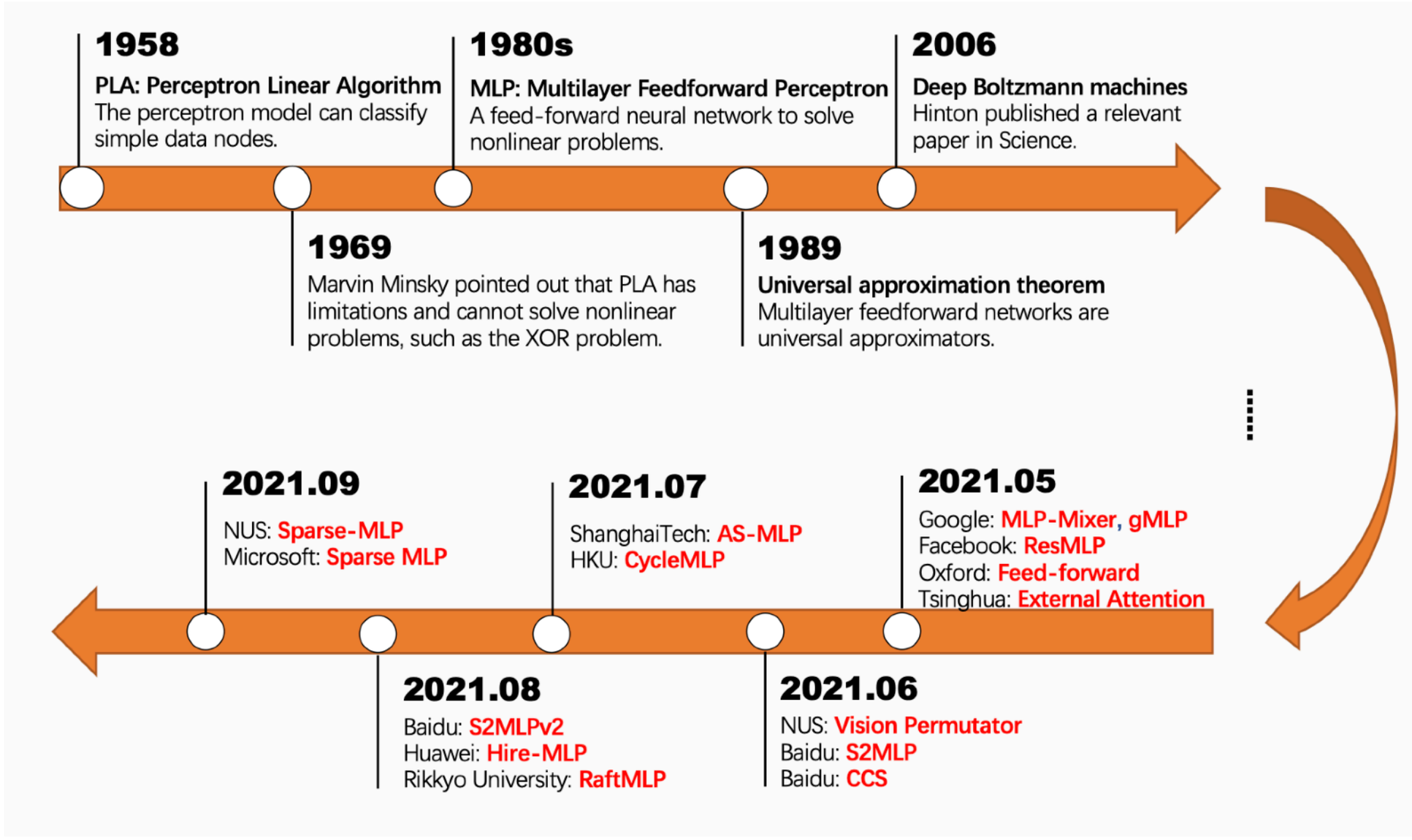
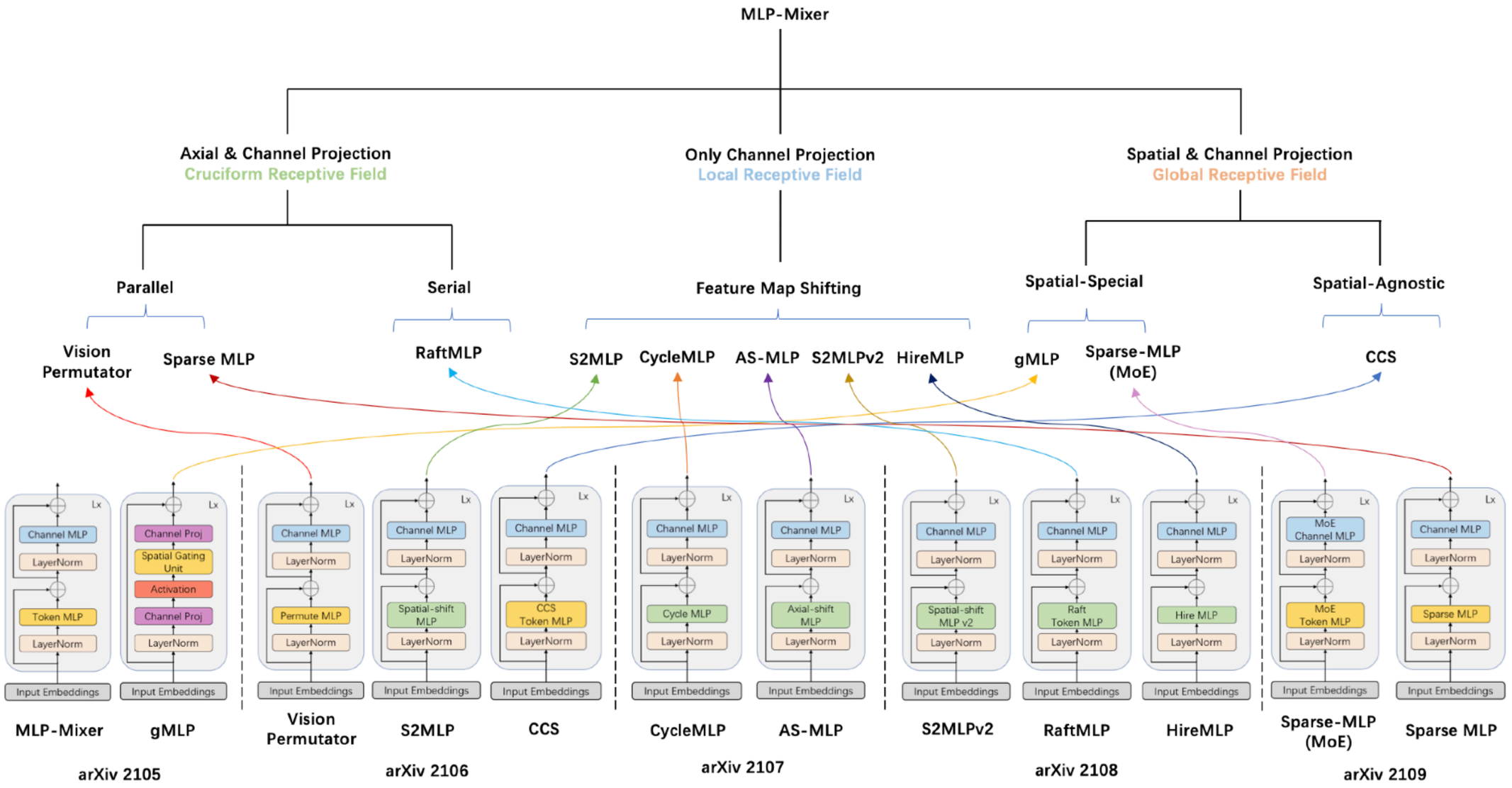


Figure 1: Key milestones in the development of MLPs. The new vision MLP models are marked in red.

MLP-Mixerの現状

5〜9月の進展：清華大学チームのサーベイ論文

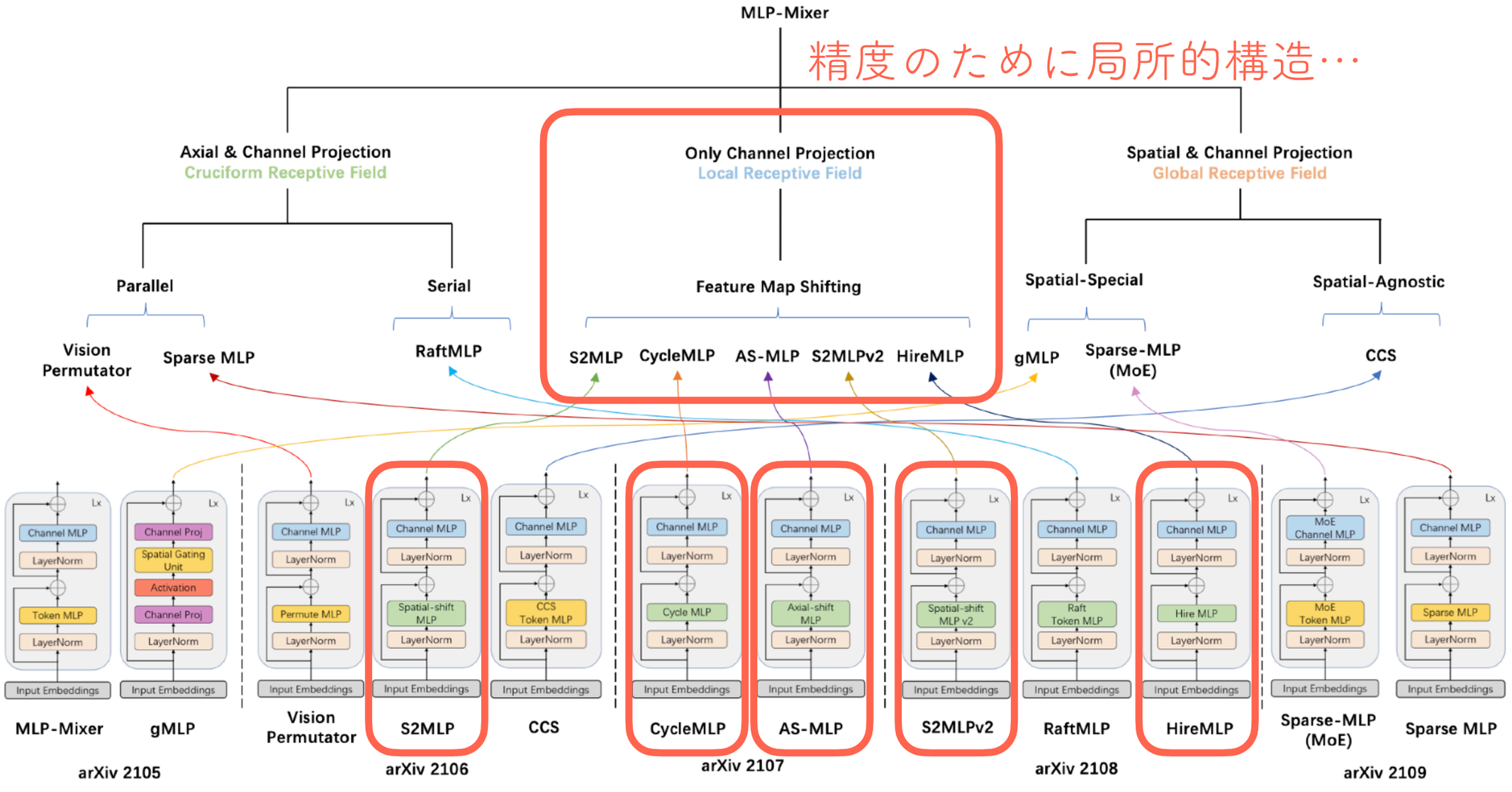
[R. Liu et al., ARE WE READY FOR A NEW PARADIGM SHIFT? A SURVEY ON VISUAL DEEP MLP. arXiv: 2111.04060]



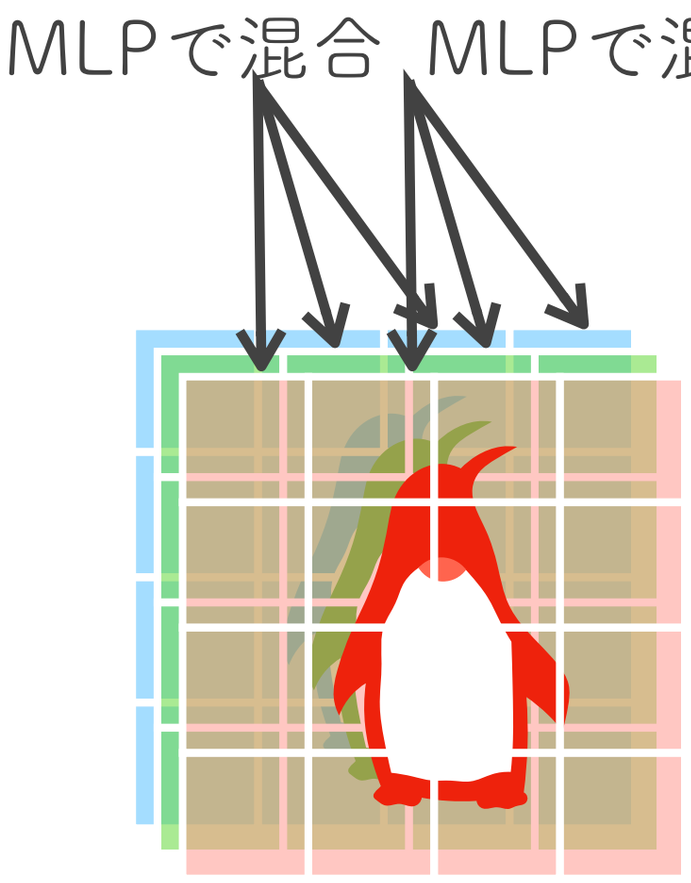
MLP-Mixerの現状

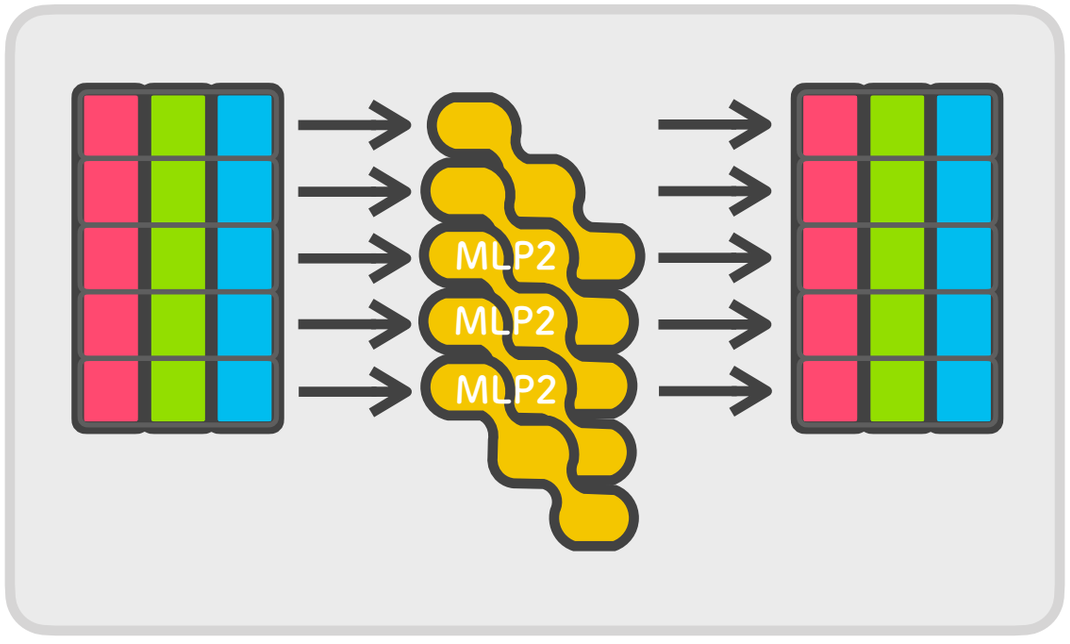
5～9月の進展：清華大学チームのサーベイ論文

[R. Liu et al., ARE WE READY FOR A NEW PARADIGM SHIFT? A SURVEY ON VISUAL DEEP MLP. arXiv: 2111.04060]



CycleMLP（local approachの例）





カーネルの作用範囲をずらしてゆく

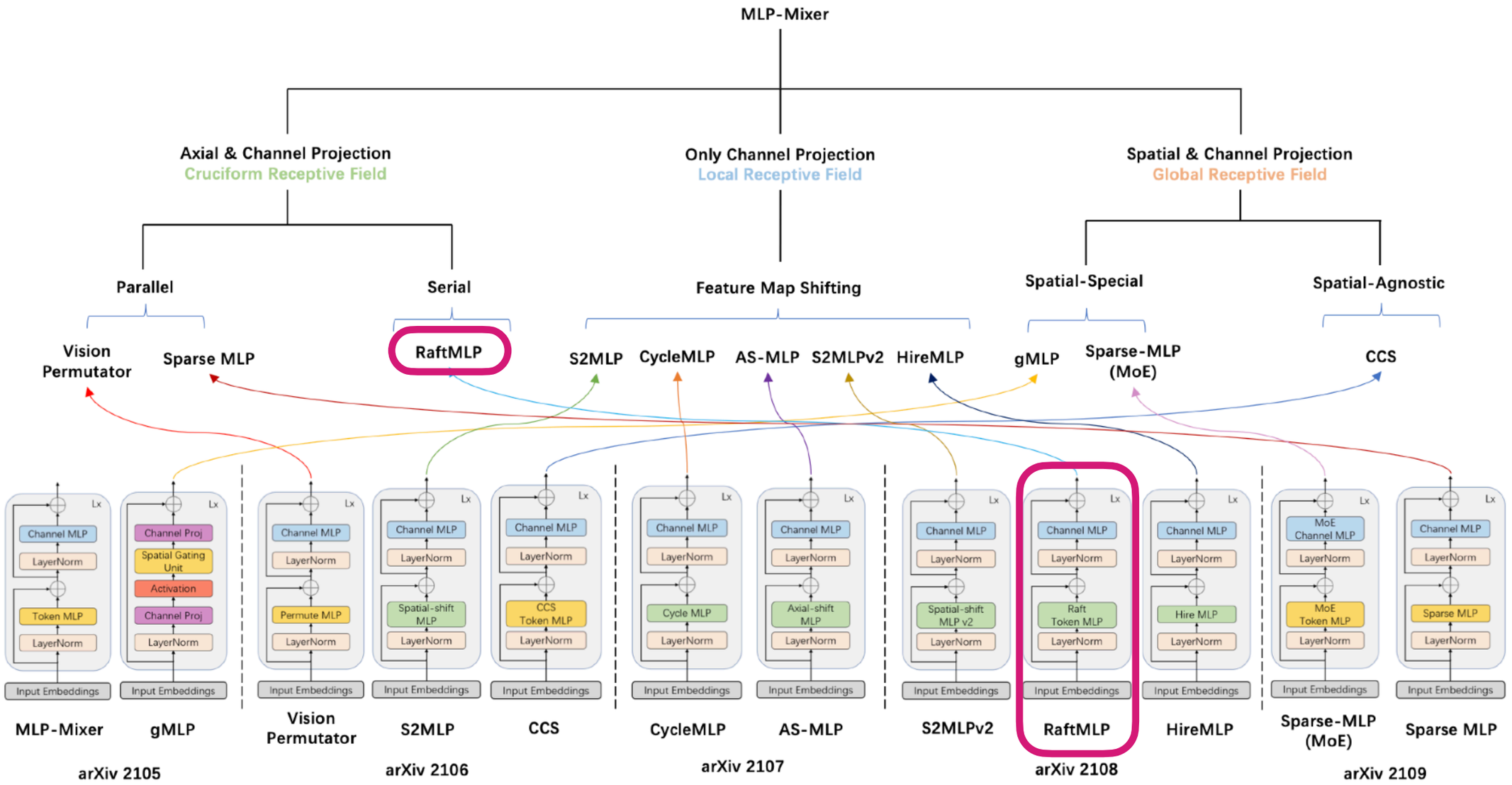
→ 畳み込みのように、受容野に局所性を課し精度向上

→ 精度は良いが、そもそものMixerのresearch question
がどっかに行ってしまう・・・

MLP-Mixerの現状

5〜9月の進展：清華大学チームのサーベイ論文

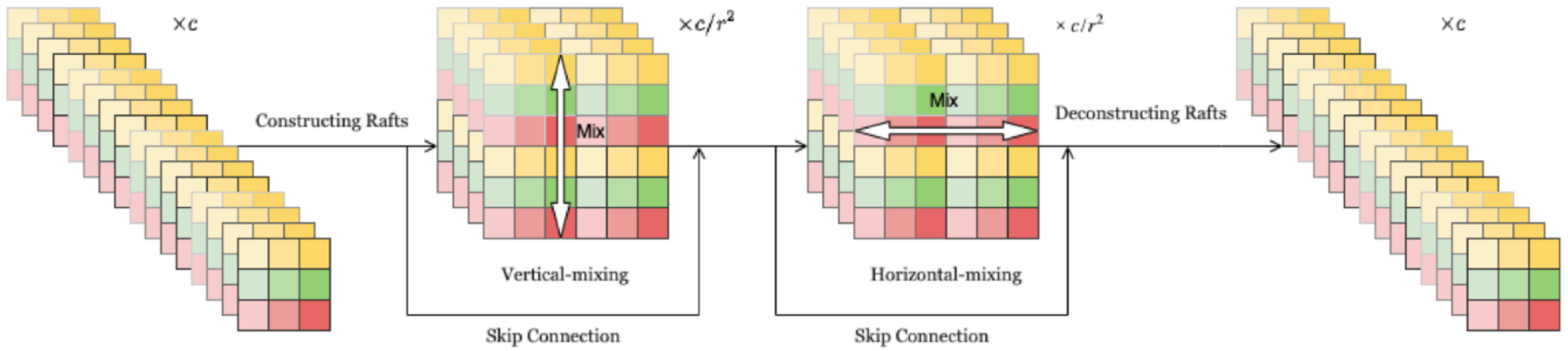
[R. Liu et al., ARE WE READY FOR A NEW PARADIGM SHIFT? A SURVEY ON VISUAL DEEP MLP. arXiv: 2111.04060]



RaftMLP https://arxiv.org/abs/2108.04384

M2の立浪祐貴さんとの論文

最小限の帰納バイアスで、MLP-Mixerの軽量化・高性能化



<u>問題意識</u>

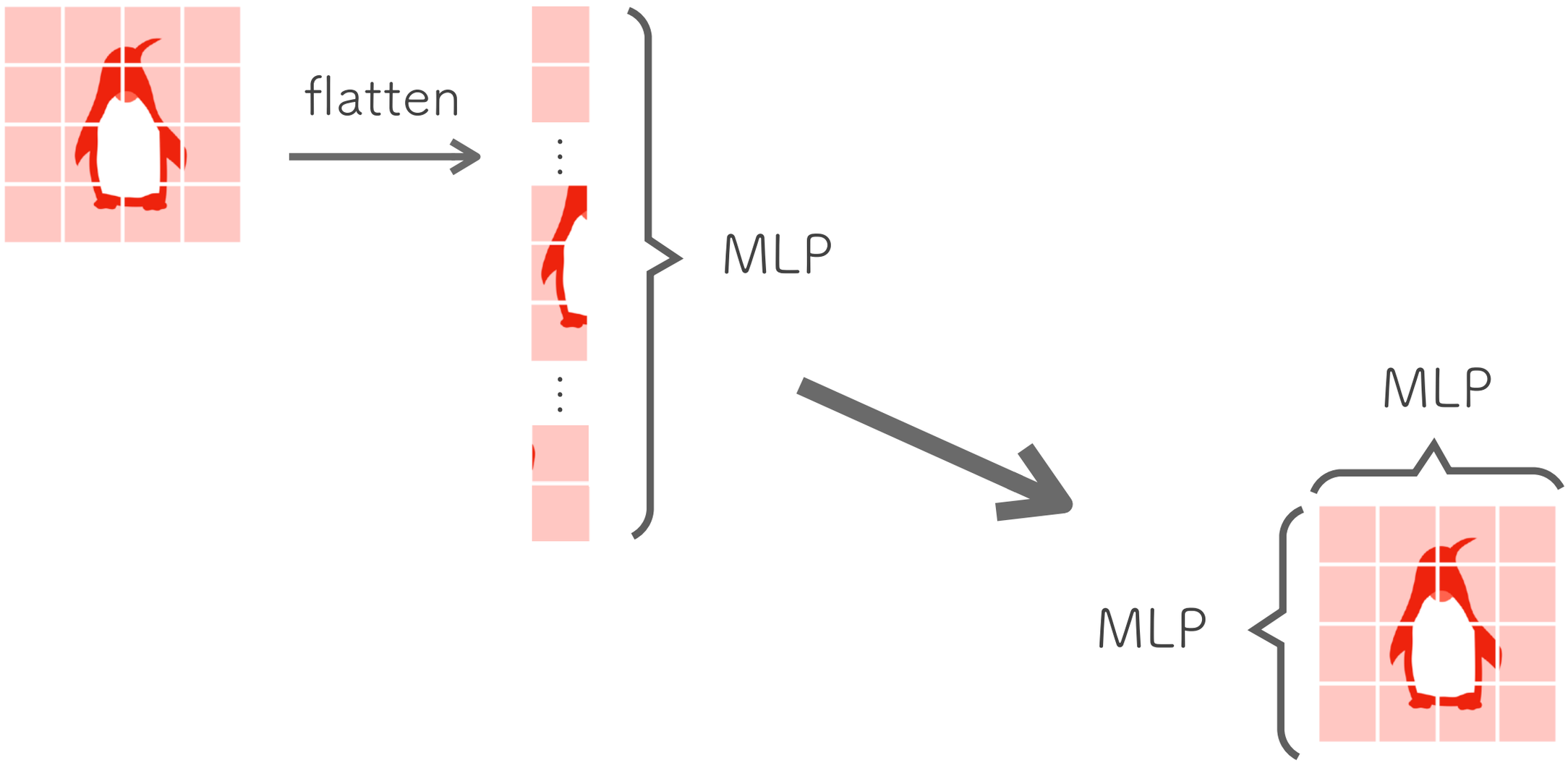
小〜中規模データでは汎化が難

→ 流石に帰納バイアスが不足

→ Mixerの良さを損なわない、ミニマルな帰納バイアスは？

RaftMLP https://arxiv.org/abs/2108.04384

Token混合を水平・垂直方向の変換に分解することで、Mixerに大域的な２次元性の帰納バイアスを持たせる



RaftMLP https://arxiv.org/abs/2108.04384

Token混合を水平・垂直方向の変換に分解することで、Mixerに大域的な２次元性の帰納バイアスを持たせる

`Raft化'により、チャネルの冗長性を空間方向に反映

トークン（パッチ）埋め込みの多スケール階層化

ImageNet-1kで、8基のRTX Quadro 8000を使い訓練

Backbone	Model	#params (M)	FROPs (G)	Top-1 Acc.(%)	Top-5 Acc.(%)
Low-resource Models (#params × FLOPs less than 50P)					
CNN	ResNet-18 [17]	11.7	1.8	69.8	89.1
	MobileNetV3 [20]	5.4	0.2	75.2	-
	EfficientNet-B0 [40]	5.3	0.4	77.1	-
Local MLP	CycleMLP-B1 [6]	15.2	2.1	78.9	-
	ConvMLP-S [27]	9.0	2.4	76.8	-
Global MLP	ResMLP-S12 [43]	15.4	3.0	76.6	-
	gMLP-Ti [31]	6.0	1.4	72.3	-
	RaftMLP-S (**ours**)	9.9	2.1	76.1	93.0
Middle-High-resource Models (#params × FLOPs more than 150P and less than 500P)					
CNN	ResNet-152 [17]	60.0	11.0	77.8	93.8
	EfficientNet-B5 [40]	30.0	9.9	83.7	-
	EfficientNetV2-S [41]	22.0	8.8	83.9	-
Transformer	PVT-M [48]	44.2	6.7	81.2	-
	Swin-S [32]	50.0	8.7	83.0	-
	Nest-S [55]	38.0	10.4	83.3	-
Local MLP	S^2-MLP-deep [51]	51.0	9.7	80.7	95.4
	CycleMLP-B3 [6]	38.0	6.9	82.4	-
	AS-MLP-S [28]	50.0	8.5	83.1	-
	ConvMLP-L [27]	42.7	9.9	80.2	-
Global MLP	Mixer-B/16 [42]	59.9	12.6	76.4	-
	ResMLP-S24 [43]	30.0	6.0	79.4	-
	RaftMLP-L (**ours**)	36.2	6.5	79.4	94.3

4. Conclusion & Discussion

これから？

'We hope that these results spark further research beyond the realms of well established CNNs and Transformers.' [arXiv:2105.01601]

果たして一過性の発展か？ beyond ImageNetスケールでのアーキテクチャへと結実か？

よりソフトで、より汎用な帰納バイアスが次のディープラーニングの発展の鍵の一つだと思われる。

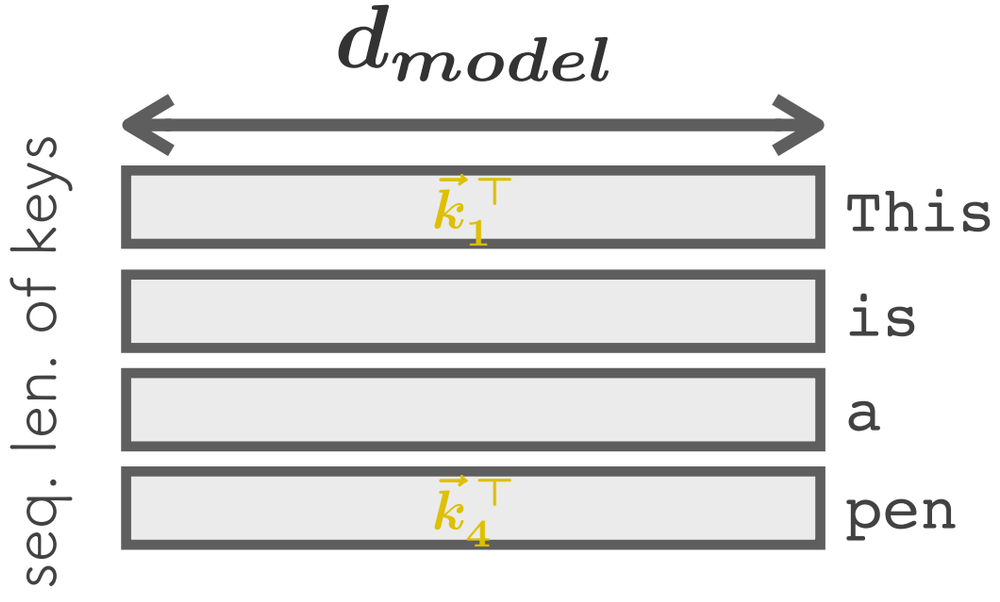
いずれにせよ汎用アーキテクチャとスケーラビリティが、今後も大規模な成果を出し続けるのは間違いない(Google、Microsoft …)

今後ドメインをまたぐマルチタスクモデルが本格的に重要になるとTransformerの流れが真価を発揮するのではないか？

Appendix

自己アテンション層

$$X \quad (seq\_len, d_{model})$$

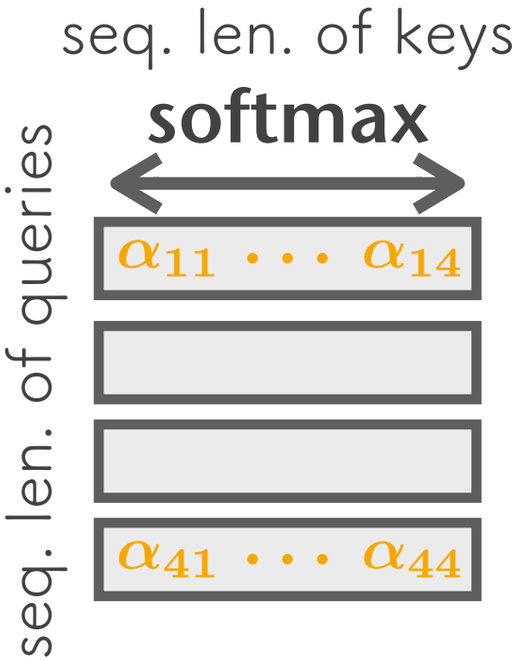


$$K = XW^K$$

$$\rightarrow \quad Q = XW^Q$$

$$V = XW^V$$

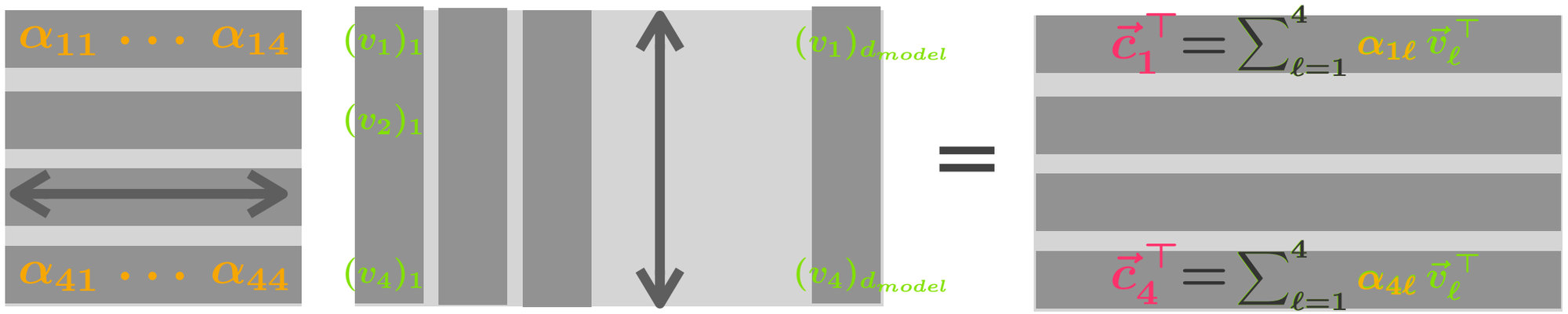
$$\rightarrow \quad Logit = \frac{1}{\sqrt{seq\_len}} QK^\top$$



$$\rightarrow \quad Att = Softmax(Logit,\ axis = -1)$$

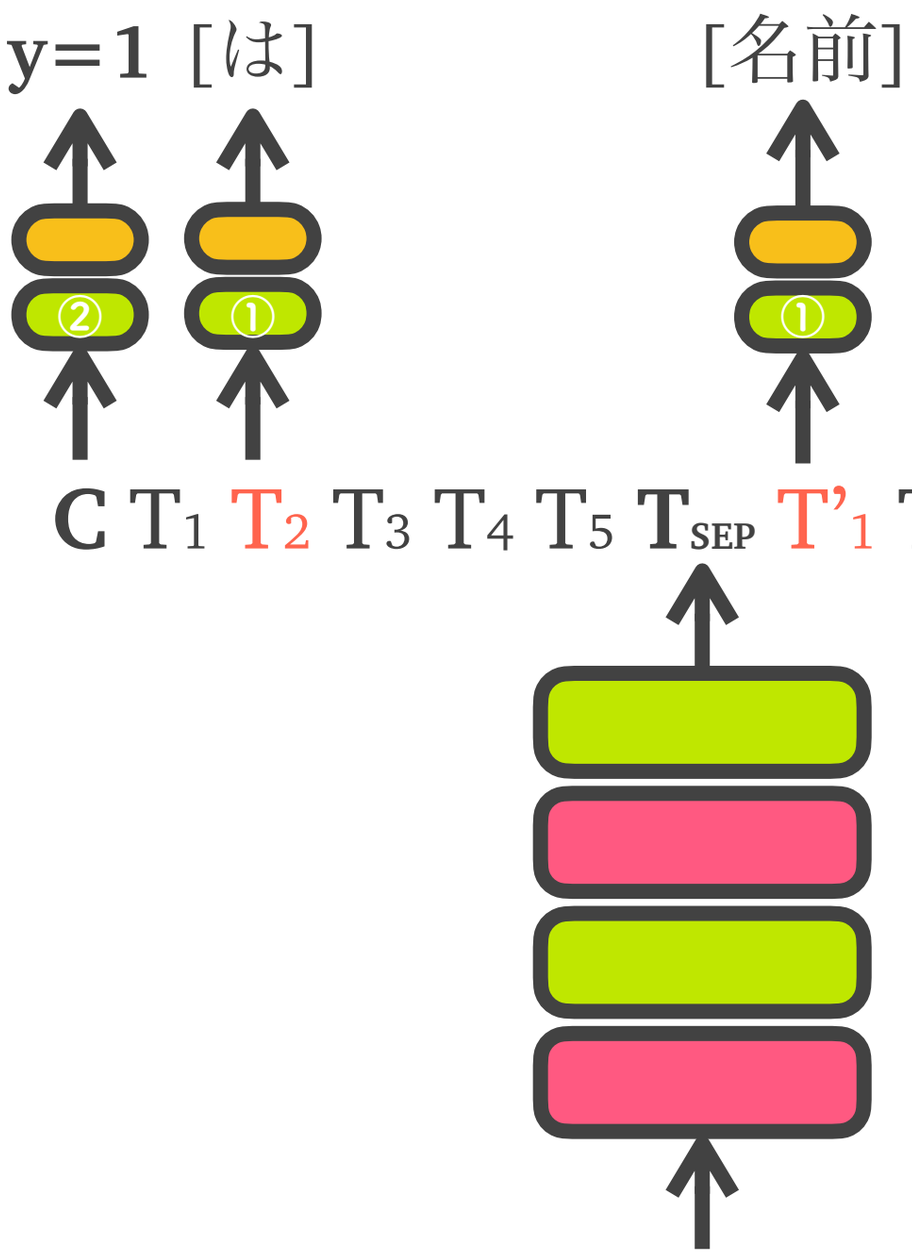
自己アテンション層

$$\longrightarrow \; C = Att\, V$$

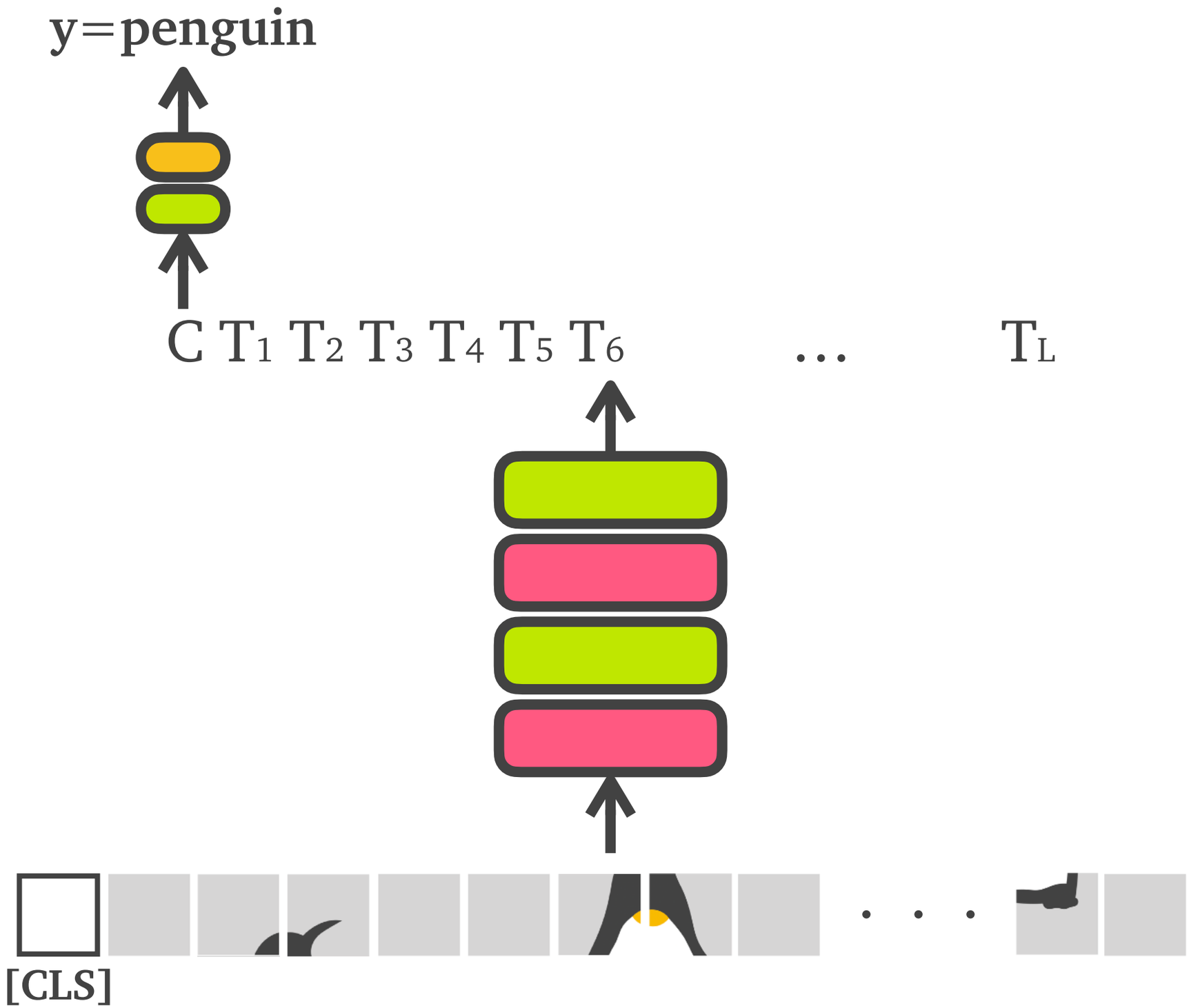


$$\longrightarrow \; O = C\, W^O$$

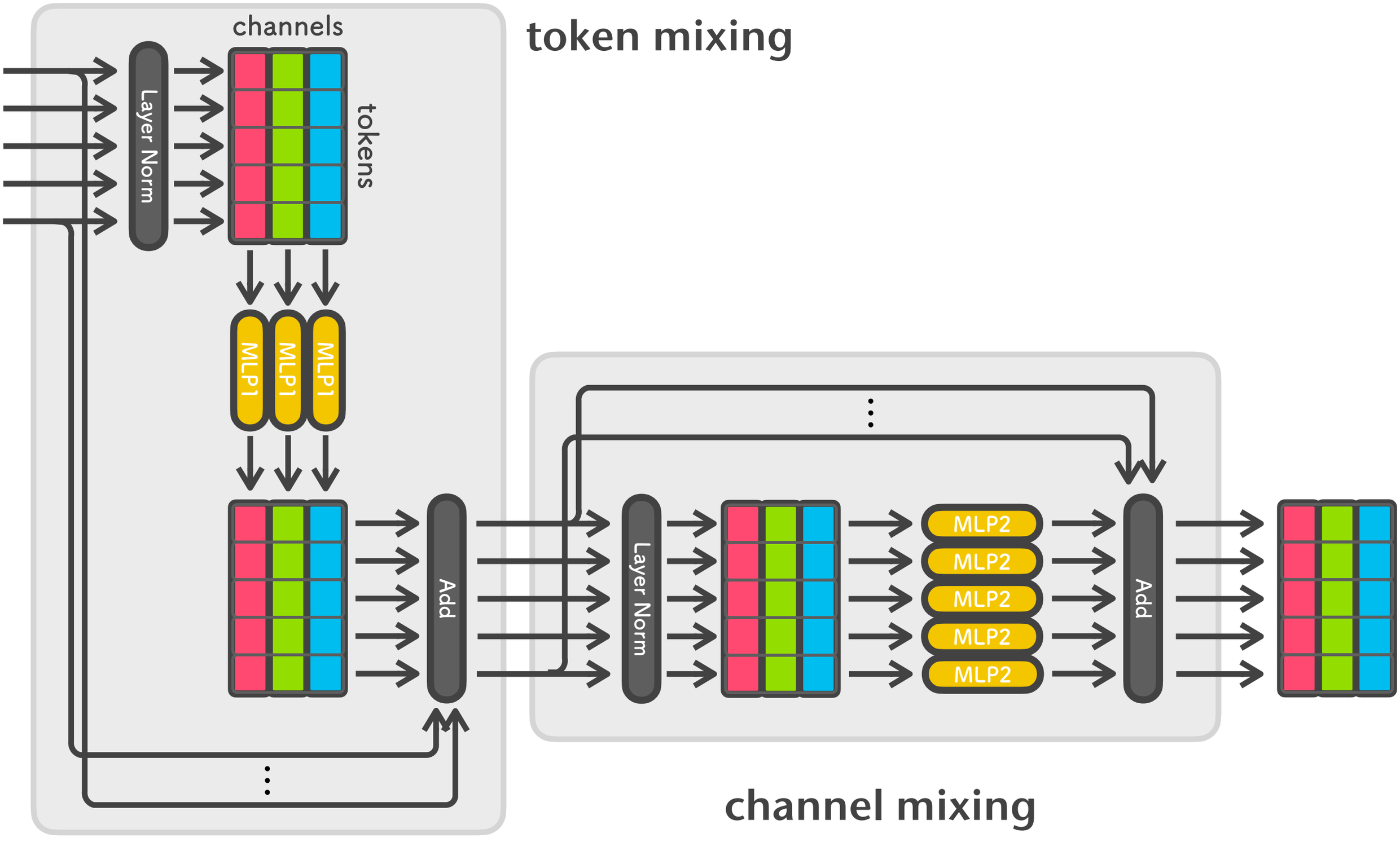
BERT：MLM & NSP事前学習に基づくTransformer



Vision Transformer (ViT)



MLP-Mixer


channels
Layer Norm
tokens
MLP1
MLP1
MLP1
Add
token mixing
Layer Norm
MLP2
MLP2
MLP2
MLP2
MLP2
Add
channel mixing