Pay Attention to MLPs

Liu, Hanxiao, et al. "Pay Attention to MLPs." *arXiv preprint arXiv:2105.08050* (2021). 정재영 발표

1. Introduction

Motivation

・It remains an open question whether the inductive bias in self-attention is essential to the remarkable effectiveness of Transformers.

・Study the necessity of self-attention modules in key language and vision applications

MLP	Self-Attention
MLPs with static parameterization can represent arbitrary functions	**The attention mechanism** introduces the inductive bias that the model can be dynamically parameterized based on the input representations

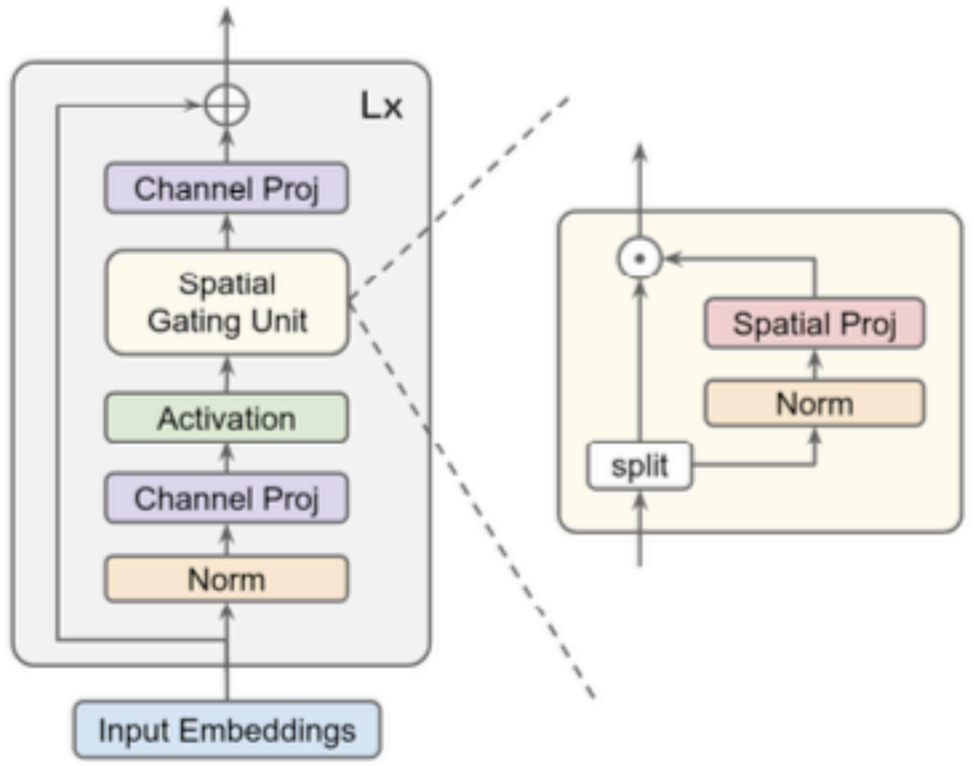
・Propose gMLP, and show experiments [image classification, Masked Language Model]

→ both pretraining and finetuning metrics for gMLPs improve as quickly as for Transformers

・Transformers can be more practically advantageous over gMLPs on tasks that require cross-sentence alignment (e.g., by 1.8% on MNLI), even with similar capacity and pretraining perplexity.

・**Overall, our results suggest that self-attention is not a necessary ingredient for scaling up machine learning models**

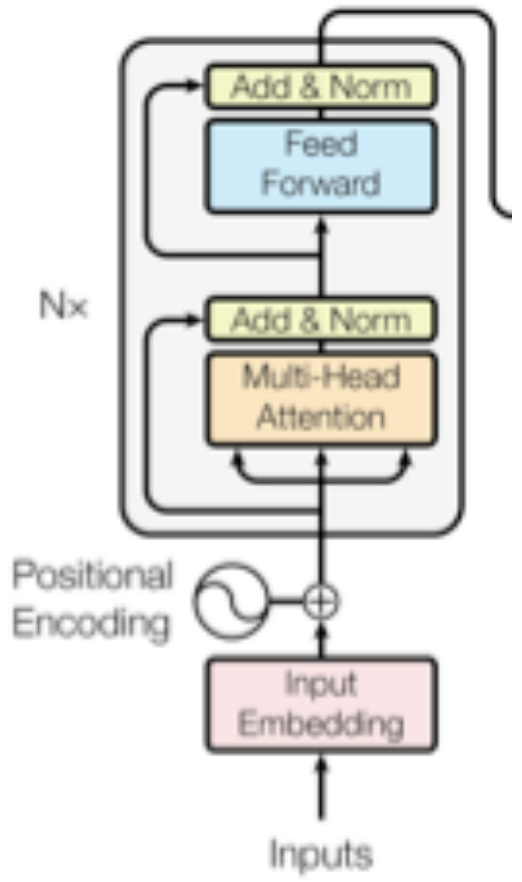
2. Model



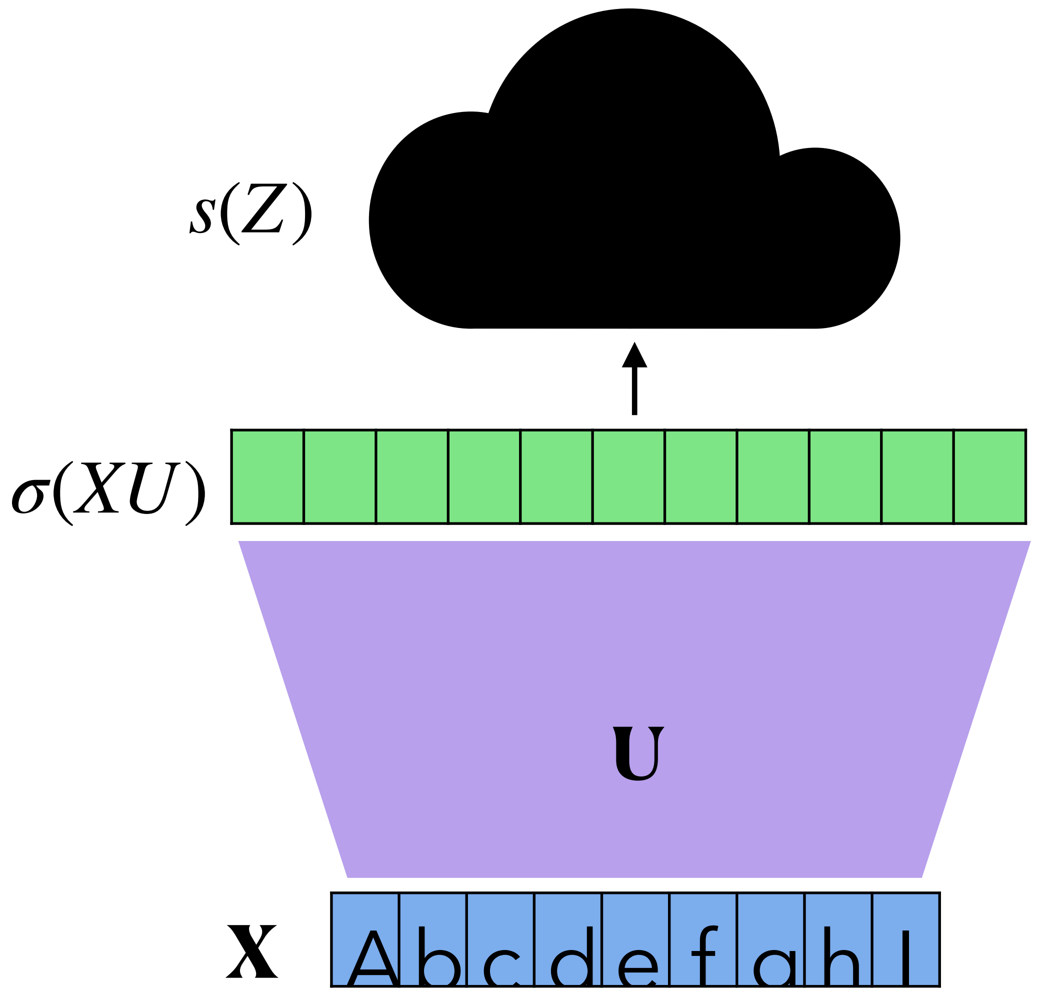
Pseudo-code for the gMLP block

```
def gmlp_block(x, d_model, d_ffn):
  shortcut = x
  x = norm(x, axis="channel")
  x = proj(x, d_ffn, axis="channel")
  x = gelu(x)
  x = spatial_gating_unit(x)
  x = proj(x, d_model, axis="channel")
  return x + shortcut

def spatial_gating_unit(x):
  u, v = split(x, axis="channel")
  v = norm(v, axis="channel")
  n = get_dim(v, axis="spatial")
  v = proj(v, n, axis="spatial", init_bias=1)
  return u * v
```



- Unlike Transformers, gMLPs do not require positional encodings, nor is it necessary to mask out the paddings during NLP finetuning.



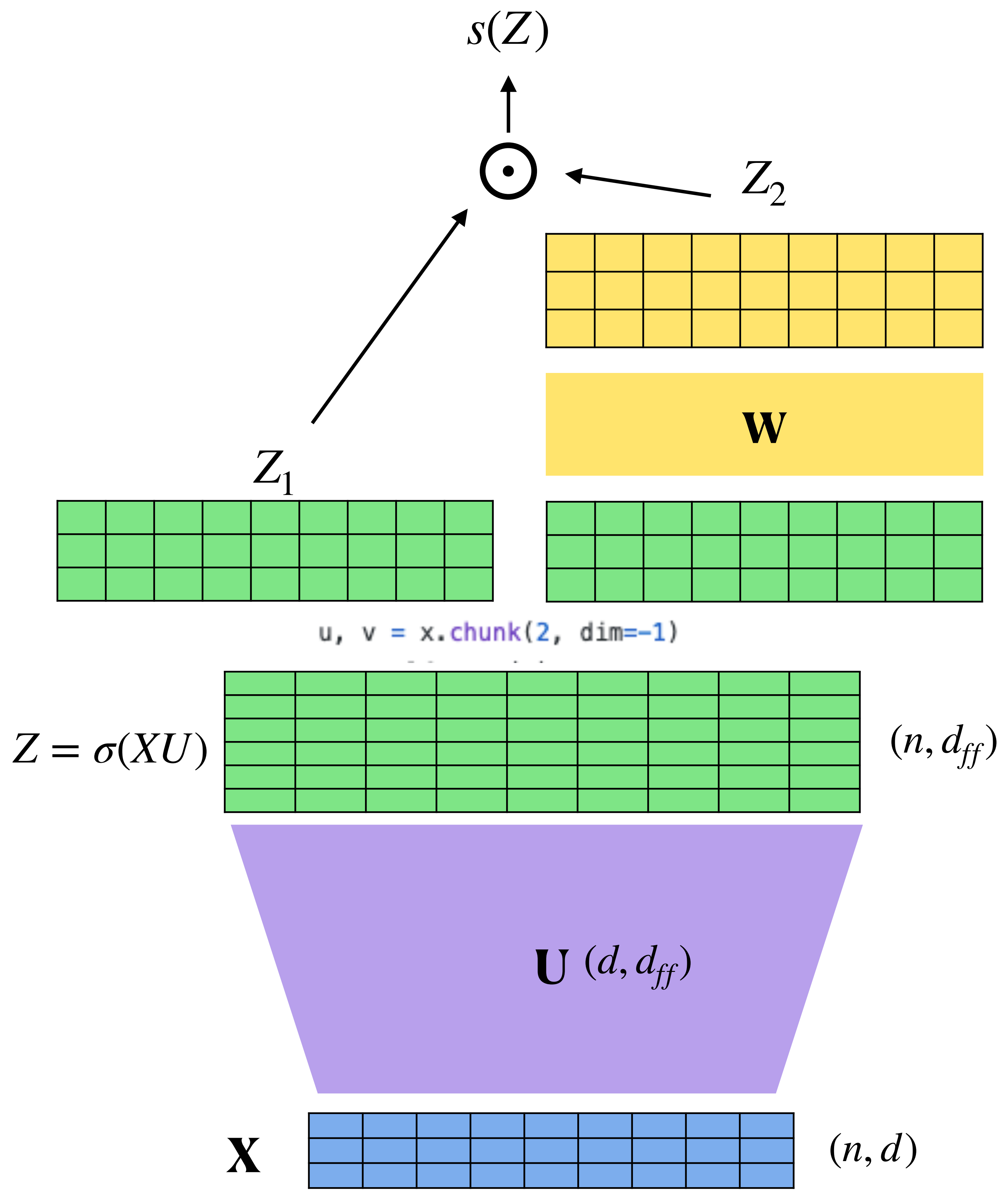
$$Z = \sigma(XU)$$
$$\tilde{Z} = s(Z)$$
$$Y = \tilde{Z}V$$

- **Spatial Gating Unit**: a layer which captures spatial interactions

→ our major focuses is therefore to design a good s capable of capturing complex spatial interactions across tokens

→ our model *does not require position embeddings* because such information will be captured in s(·)

2.1 Spatial Gating Unit

Equations	Plot
$$f_{W,b}(Z) = WZ + b$$ $$s(Z) = Z_1 \odot f_{W,b}(Z_2)$$	

- For training stability, we find it critical to initialize W as near-zero values and b as ones, meaning that s(·) is approximately an identity mapping at the beginning of training.
- the magnitude for each element in Z can be rapidly tuned according to the gating function $f_{w,b}(\cdot)$

3. Image Classification

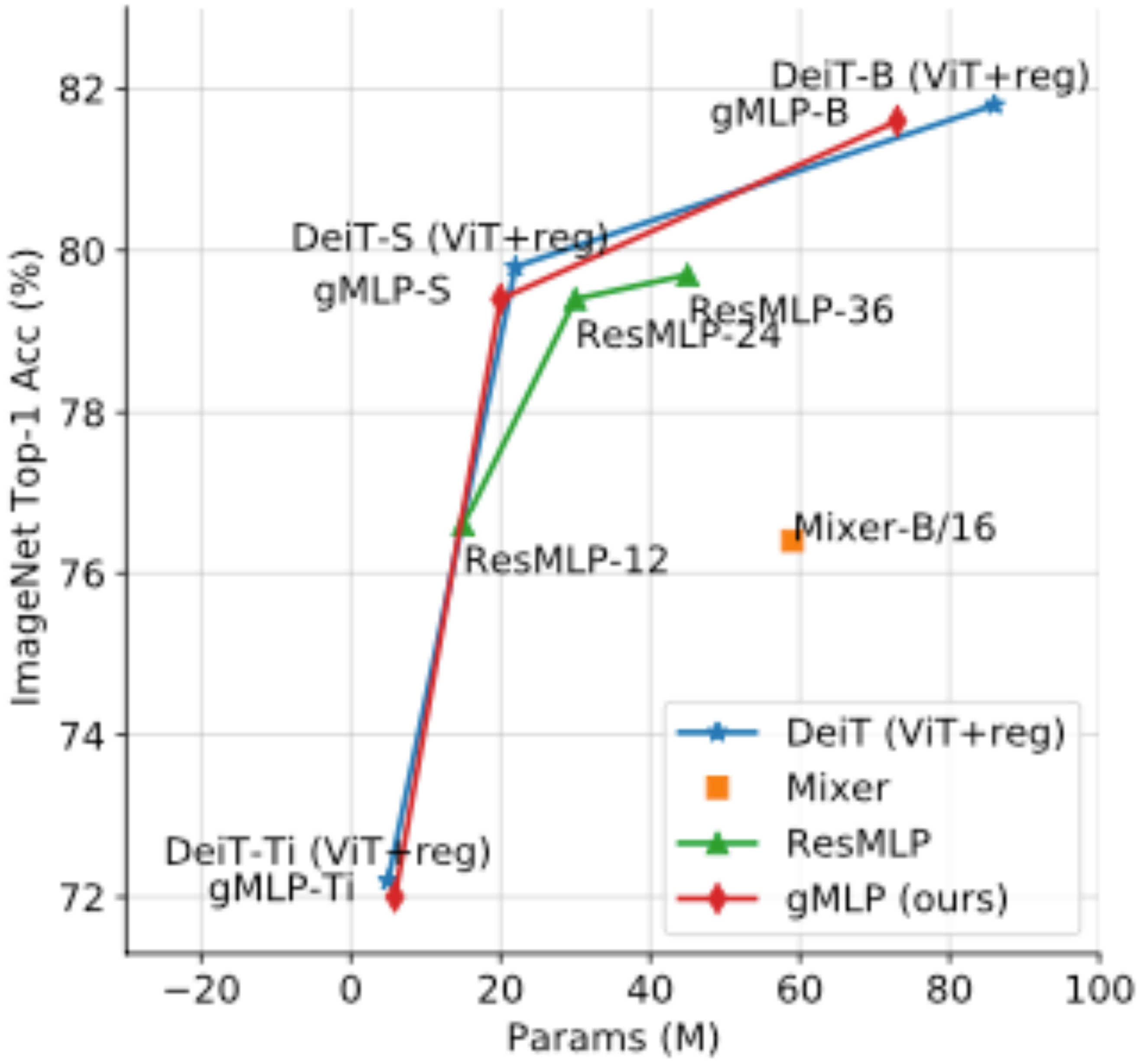


Figure 2: ImageNet accuracy vs model capacity.

・We compare our attention-free models with recent attentive models based on vanilla Transformers, including Vision Transformer (ViT) [7], DeiT [8] (ViT with improved regularization), and several other representative convolutional networks.

・The accuracy-parameter/FLOPs tradeoff of gMLPs surpasses all concurrently proposed MLP-like architectures , which we attribute to the effectiveness of our Spatial Gating Unit

3. Image Classification

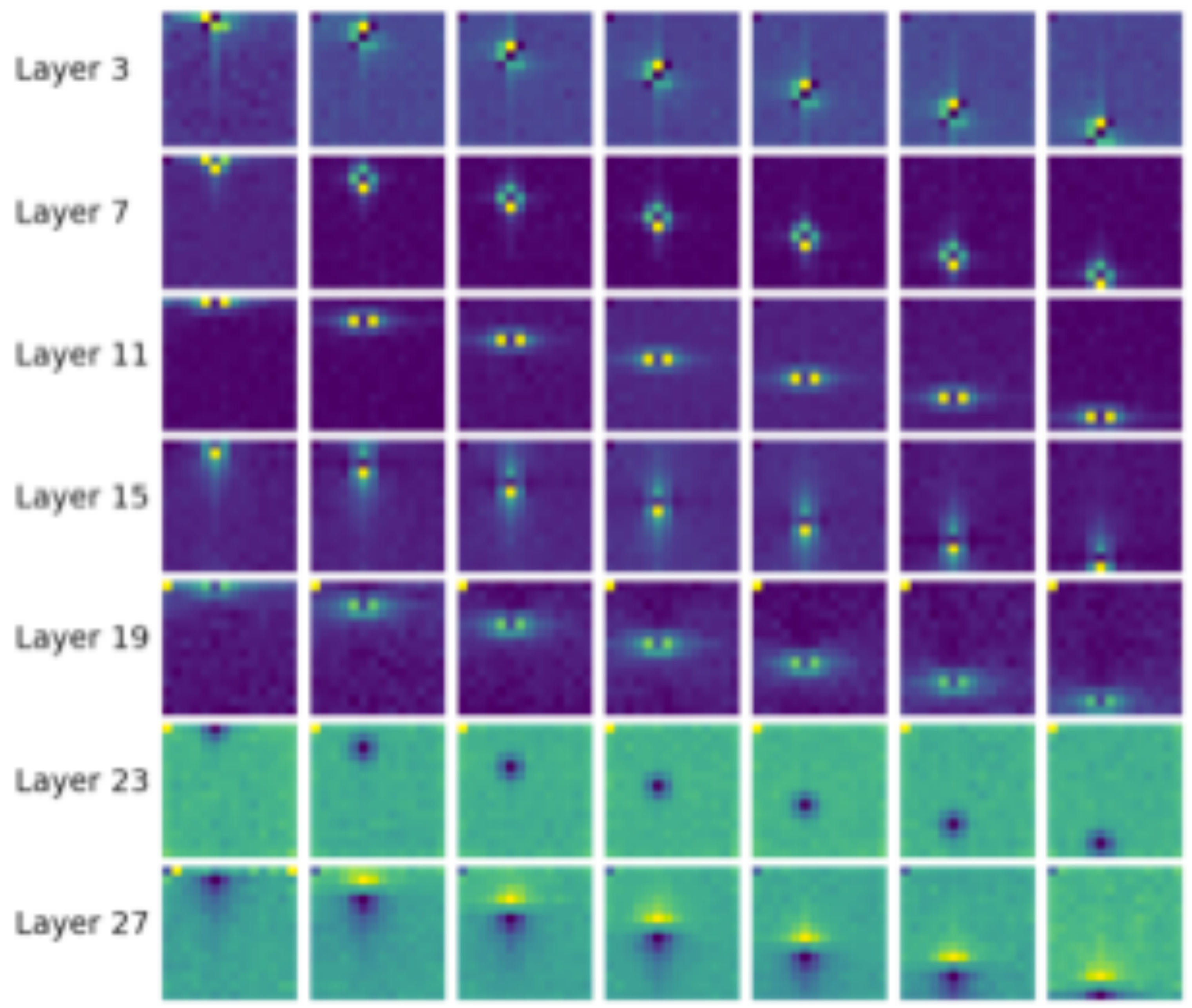


Figure 3: Spatial projection weights in gMLP-B. Each row shows the filters (reshaped into 2D) for a selected set of tokens in the same layer.

・The spatial weights after learning exhibit both locality and spatial invariance. In other words, each spatial projection matrix effectively learns to perform convolution with a data-driven, irregular (non-square) kernel shape.

4. Masked Language Modeling with BERT

- We do not use positional encodings.
- We also find it unnecessary to mask out <pad> tokens in gMLP blocks during finetuning as the model can quickly learn to ignore them.

4.1 Ablation: The Importance of Gating in gMLP for BERT's Pretraining

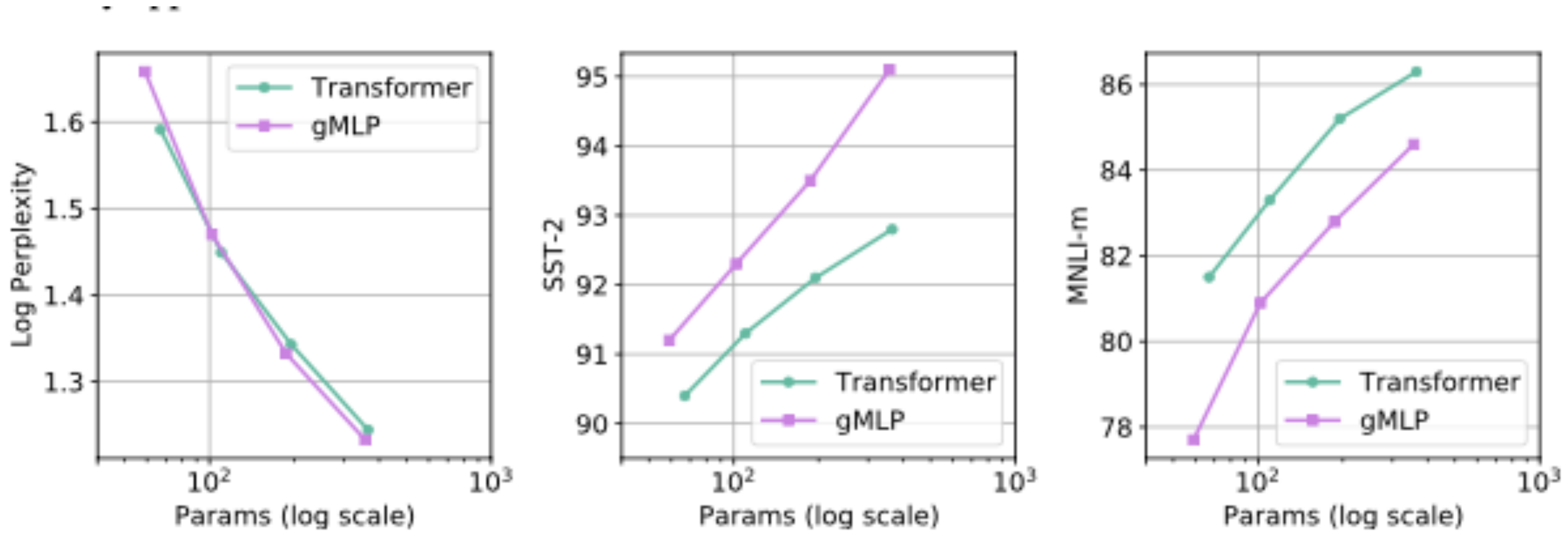
Model	Perplexity	Params (M)
$\text{BERT}_{\text{base}}$ BERT with a Transformer architecture and learnable absolute position embeddings.	4.37	110
$\text{BERT}_{\text{base}}$ + rel pos BERT with a Transformer architecture and T5-style learnable relative position biases	4.26	110
$\text{BERT}_{\text{base}}$ + rel pos - attn	5.64	96
Linear gMLP, $s(Z) = f(Z)$	5.14	92
Additive gMLP, $s(Z) = Z + f(Z)$	4.97	92
Multiplicative gMLP, $s(Z) = Z \odot f(Z)$	4.53	92
Multiplicative, Split gMLP, $s(Z) = Z_1 \odot f(Z_2)$, $Z = Z_1 \| Z_2$	4.35	102

1. SGU outperforms other variants in perplexity
2. gMLP with SGU also achieves perplexity comparable to Transformer.

4.2 Case Study: The Behavior of gMLP as Model Size Increases

Model	#L	Params (M)	Perplexity	SST-2	MNLI-m
Transformer	6+6	67	**4.91**	90.4	81.5
gMLP	18	59	5.25	91.2	77.7
Transformer	12+12	110	**4.26**	91.3	83.3
gMLP	36	102	4.35	92.3	80.9
Transformer	24+24	195	3.83	92.1	85.2
gMLP	72	187	**3.79**	93.5	82.8
Transformer	48+48	365	3.47	92.8	86.3
gMLP	144	357	**3.43**	95.1	84.6

・The results above show that a deep enough gMLP is able to match and even outperform the perplexity of Transformers with comparable capacity

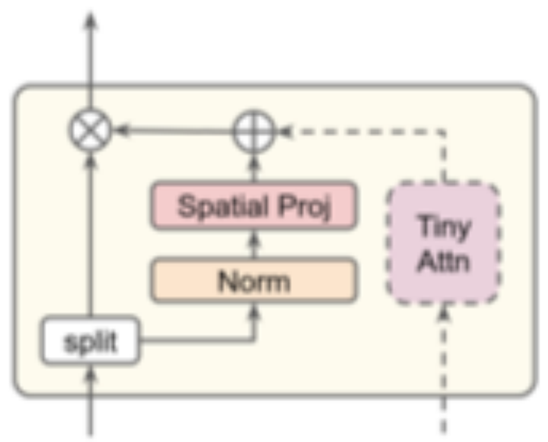


・Our attention-free model is advantageous on SST-2 but worse on MNLI is particularly informative—the former is a single-sentence task whereas

이진 분류(감성 분석)　{전제, 결론} 쌍으로 이루어진 자연어 추리

the latter involves sentence pairs (premise and hypothesis)

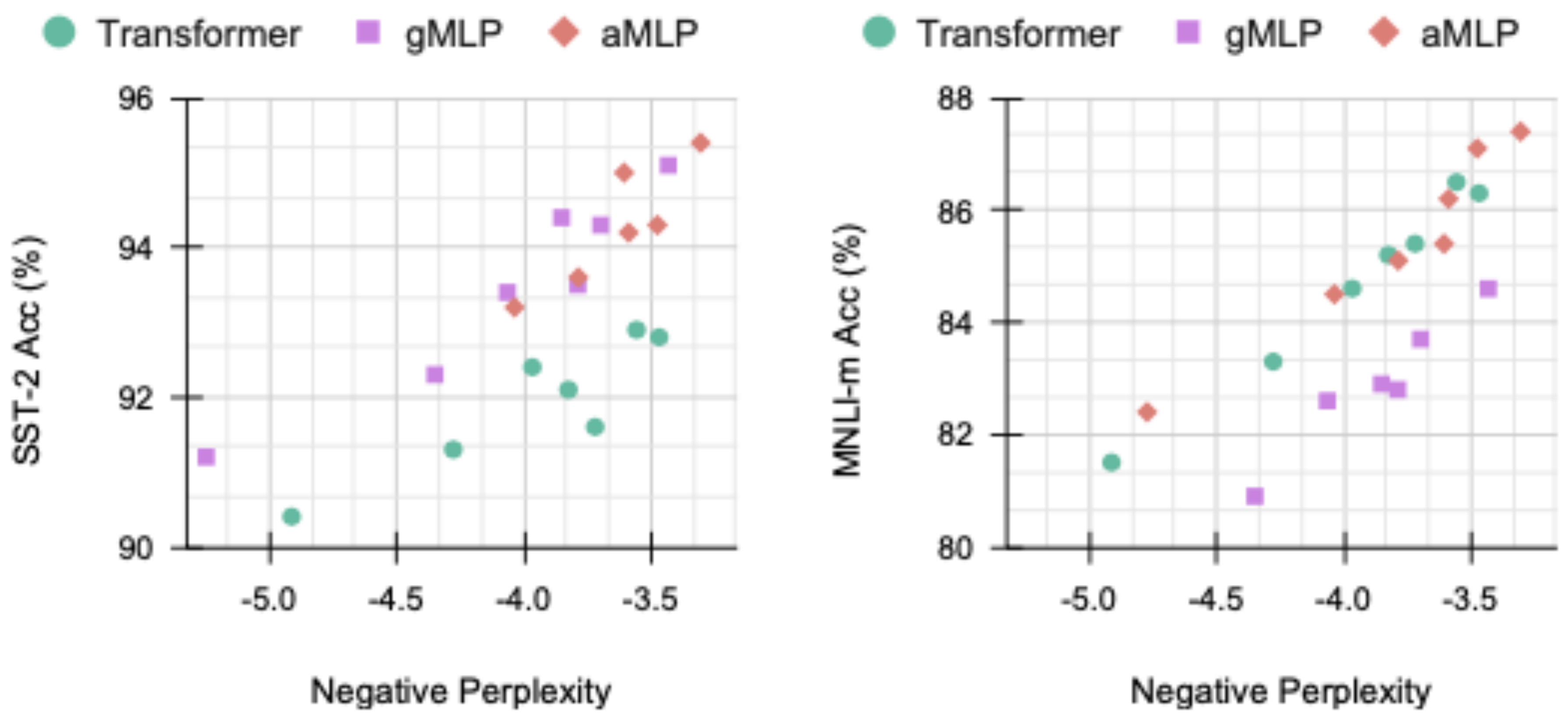
4.3 Ablation: The Usefulness of Tiny Attention in BERT's Finetuning



Pseudo-code for the tiny attention module

```
def tiny_attn(x, d_ffn, d_attn=64):
  qkv = proj(x, 3 * d_attn, axis="channel")
  q, k, v = split(qkv, 3, axis="channel")
  w = einsum("bnd,bmd->bnm", q, k)
  a = softmax(w * rsqrt(d_attn))
  x = einsum("bnm,bmd->bnd", a, v)
  return proj(x, d_ffn, axis="channel")
```

- To isolate the effect of attention, we experiment with a hybrid model where a tiny self-attention block is attached to the gating function of gMLP (Figure 6)



4.4 Main Results for MLM in the BERT Setup

	Perplexity	SST-2	MNLI (m/mm)	SQuAD v1.1	SQuAD v2.0	Attn Size	Params (M)
$\text{BERT}_{\text{base}}$ [2]	–	92.7	84.4/-	88.5	76.3	768 (64×12)	110
$\text{BERT}_{\text{base}}$ (ours)	4.17	93.8	85.6/85.7	90.2	78.6	768 (64×12)	110
$\text{gMLP}_{\text{base}}$	4.28	94.2	83.7/84.1	86.7	70.1	–	130
$\text{aMLP}_{\text{base}}$	3.95	93.4	85.9/85.8	90.7	80.9	64	109
$\text{BERT}_{\text{large}}$ [2]	–	93.7	86.6/-	90.9	81.8	1024 (64×16)	336
$\text{BERT}_{\text{large}}$ (ours)	3.35	94.3	87.0/87.4	92.0	81.0	1024 (64×16)	336
$\text{gMLP}_{\text{large}}$	3.32	94.8	86.2/86.5	89.5	78.3	–	365
$\text{aMLP}_{\text{large}}$	3.19	94.8	88.4/88.4	92.2	85.4	128	316

Table 6: Pretraining perplexities and dev-set results for finetuning. "ours" indicates models trained using our setup. We report accuracies for SST-2 and MNLI, and F1 scores for SQuAD v1.1/2.0.

- our $\text{gMLP}_{\text{large}}$ achieves 89.5% F1 on SQuAD-v1.1 without any attention or dynamic parameterization mechanism
- our hybrid model $\text{aMLP}_{\text{large}}$ achieves 4.4% higher F1 than Transformers on the more difficult SQuAD-v2.0 task.

5 Conclusion

- We show that gMLPs, a simple variant of MLPs with gating, can be competitive with Transformers in terms of BERT's pretraining perplexity and ViT's accuracy.