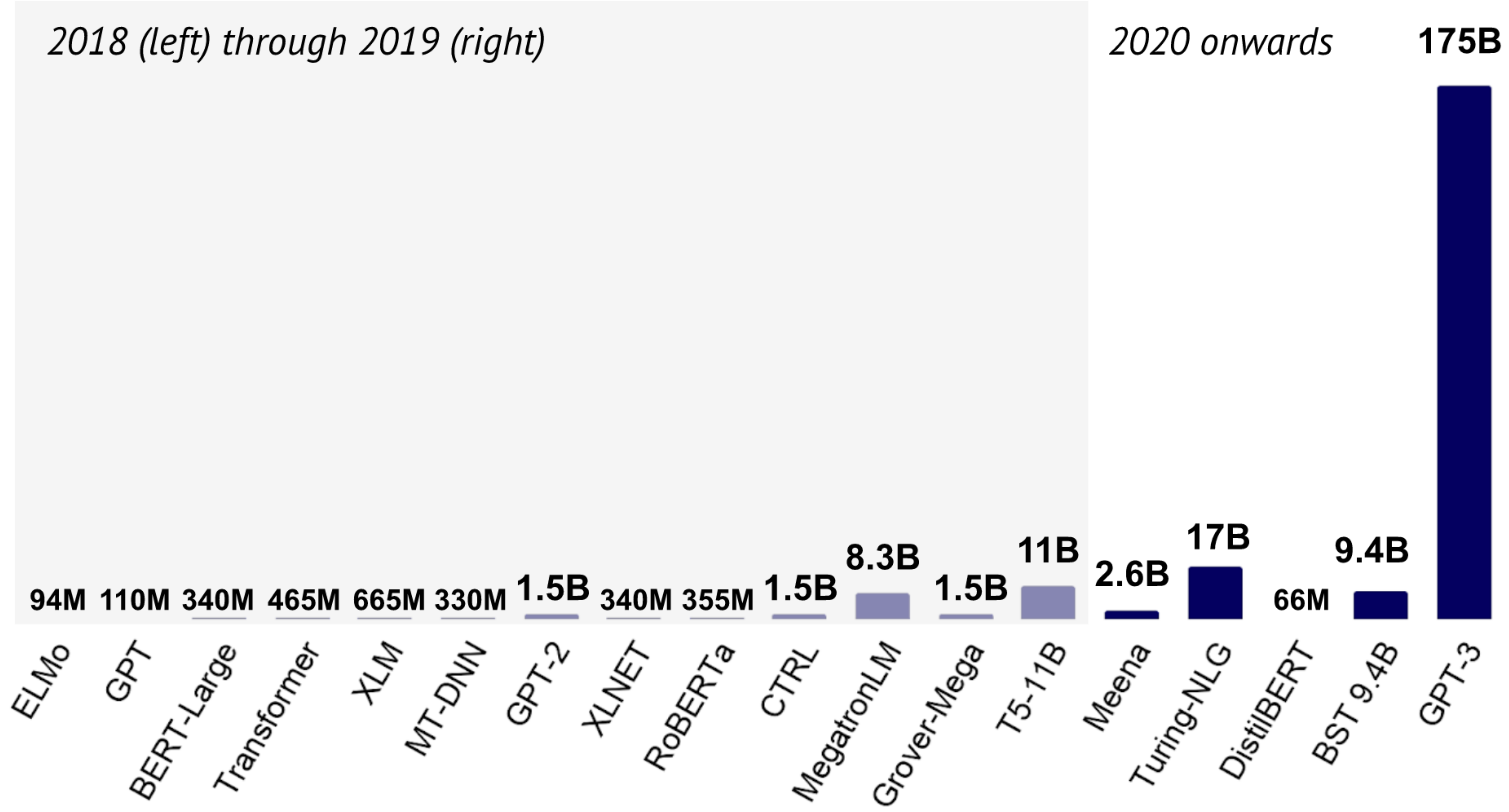
Large Product Key Memory for Pretrained Language Models

QRAFT | AXE
GUIJIN SON

Recent NLP Trends

Large Product Key Memory for Pretrained Language Models



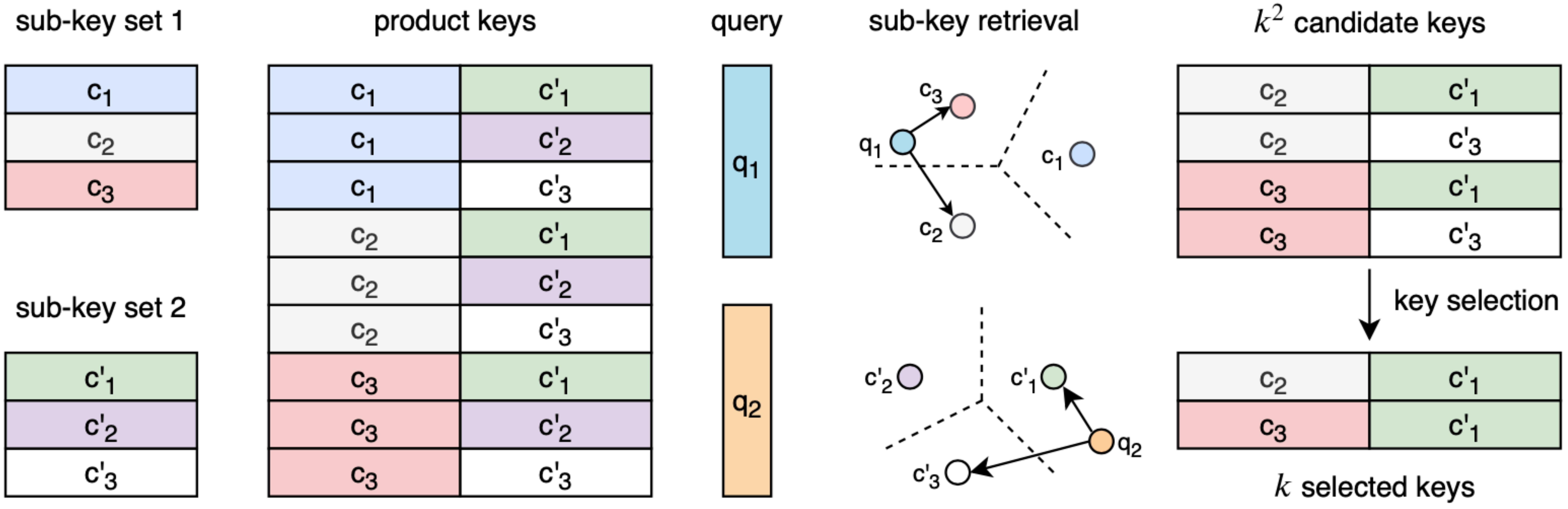
Decoupling Parameters & Performance

Large Product Key Memory for Pretrained Language Models

Model	# Layers	# Params	Inference Speed (batch/sec)
$\text{BERT}_{\text{BASE}}$	12	110M	79.8
$\text{BERT}_{\text{BASE}}$ +PKM	12	506M	61.4
$\text{BERT}_{\text{BASE}}$ +ResM	12	**515M**	**59.3**
$\text{BERT}_{\text{LARGE}}$	24	**340M**	**43.1**
$\text{BERT}_{\text{LARGE}}$ +PKM	24	860M	37.2
$\text{BERT}_{\text{LARGE}}$ +ResM	24	876M	36.1

Product Key Memory

Large Product Key Memory for Pretrained Language Models



Product Key Memory : Pseudo Code

Large Product Key Memory for Pretrained Language Models

```
import torch
import torch.nn as nn
import torch.nn.functional as F

def product_key_memory(query, dim, d_model, d_query, n_head):

  b , t , _ , h = query.shape , n_head

  query = nn.Linear(d_model,d_query)(query)
  query = nn.Dropout(self.MaskedBatchNorm(query))
  query_list = query.chunk(2, dim = -1)

  query = torch.stack(query_list).reshape(2,b,t,h,-1)
  #self.key = nn.parameters(),
  weight = torch.matmul(query_list,self.key,transpose_a = True)
  scores, indices = weight.topk(k=self.topk, dim=-1)
  scores = F.softmax(scores,dim = -1) #normalize

  top_k = sorted(dict((indices,scores)).items(), key=lambda x: x[1],reverse=True)[:10]
  top_k_indice = [indice for indice,score in top_k]

  out = nn.EmbeddingBag(n_key**2,dim,mode='sum')(top_k_indice)

  return nn.Dropout(out)
```

PKM Performance

Large Product Key Memory for Pretrained Language Models

Model	Memory			MLM		
	$\widetilde{MU}$ (4L/8L) (%)	KL_u (4L/8L)	KL_w (4L/8L)	WT-2 (ppl)	WT-103 (ppl)	PG-19 (ppl)
(a) BERT$_{\text{BASE}}$ †	-	-	-	3.49	3.86	6.18
(b) +500k steps	-	-	-	3.40	3.72	5.88
(c) +PKM	2.2/84.1	1.62/0.89	1.99/1.13	3.26	3.39	5.53
(d) +ResM	75.0/81.0	1.50/0.71	1.80/0.92	3.26	3.36	5.45
(e) +Init +PKM	97.4/95.7	0.53/0.69	0.68/0.88	3.14	3.26	5.22
(f) +Init +ResM	**98.2/97.3**	**0.45/0.46**	**0.58/0.60**	**3.10**	**3.20**	**5.14**

Catastrophic Drift

Large Product Key Memory for Pretrained Language Models

Model	QA SQuAD 1.1 (EM/F1)	GLUE MNLI-(m/mm) (Acc)	QQP (Acc)	QNLI (Acc)	SST-2 (Acc)	CoLA (Matt)	Avg -
(a) $\text{BERT}_{\text{BASE}}^{\dagger}$	82.7/89.8	84.3/84.5	91.0	89.3	92.8	60.8	83.8
(b) +500k steps	83.3/90.1	84.8/84.9	91.2	89.2	92.4	61.4	84.0
(c) +PKM	81.9/89.1	84.4/85.0	91.1	89.0	93.6	59.7	83.8
(d) +ResM	81.5/89.4	84.6/84.8	91.0	88.2	93.2	62.8	84.1
(e) +Init +PKM	83.8/90.6	85.8/85.6	91.2	90.0	93.6	63.6	85.0
(f) +Init +ResM	**83.9/90.8**	**86.0/85.8**	**91.4**	**90.4**	**94.0**	**64.1**	**85.3**
(g) $\text{BERT}_{\text{BASE}}^{\star}$	81.1/88.5	83.9/84.4	91.0	88.4	92.9	59.8	83.4
(h) $\text{BERT}_{\text{LARGE}}^{\star}$	83.3/90.6	86.2/86.1	91.4	90.4	93.8	64.1	85.3

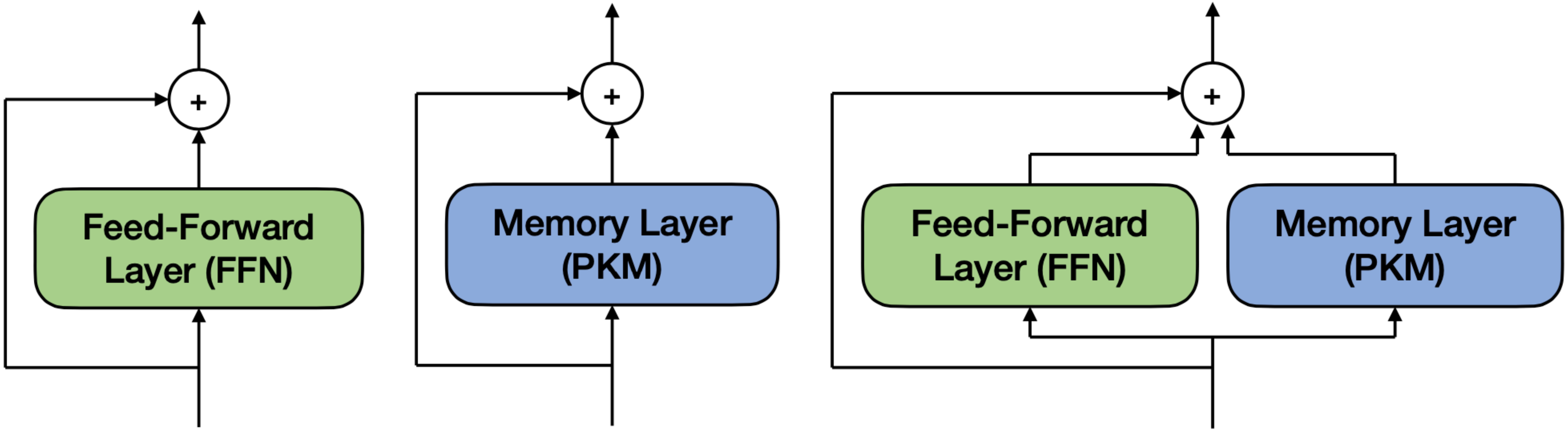
Solutions

Large Product Key Memory for Pretrained Language Models

- Initialization from Pretrained Weights
- Residual Memory Layer

Model Architecture

Large Product Key Memory for Pretrained Language Models



(a) FFN layer (b) PKM layer (c) ResM layer

ResM Performance

Large Product Key Memory for Pretrained Language Models

Model	QA SQuAD 1.1 (EM/F1)	GLUE MNLI-(m/mm) (Acc)	QQP (Acc)	QNLI (Acc)	SST-2 (Acc)	CoLA (Matt)	**Avg** -
(a) $\text{BERT}_{\text{BASE}}{}^{\dagger}$	82.7/89.8	84.3/84.5	91.0	89.3	92.8	60.8	83.8
(b) +500k steps	83.3/90.1	84.8/84.9	91.2	89.2	92.4	61.4	84.0
(c) +PKM	81.9/89.1	84.4/85.0	91.1	89.0	93.6	59.7	83.8
(d) +ResM	81.5/89.4	84.6/84.8	91.0	88.2	93.2	62.8	84.1
(e) +Init +PKM	83.8/90.6	85.8/85.6	91.2	90.0	93.6	63.6	85.0
(f) +Init +ResM	**83.9/90.8**	**86.0/85.8**	**91.4**	**90.4**	**94.0**	**64.1**	**85.3**
(g) $\text{BERT}_{\text{BASE}}{}^{\star}$	81.1/88.5	83.9/84.4	91.0	88.4	92.9	59.8	83.4
(h) $\text{BERT}_{\text{LARGE}}{}^{\star}$	83.3/90.6	86.2/86.1	91.4	90.4	93.8	64.1	85.3

Further Implications

Large Product Key Memory for Pretrained Language Models

Model	MLM PG-19 (ppl)	QA SQuAD 1.1 (EM/F1)	GLUE MNLI-m (Acc)	GLUE SST-2 (Acc)
DistilBERT*	20.61	77.4/85.7	82.0	91.6
+Init +ResM	5.75	80.4/88.3	84.1	93.3
$\text{BERT}_{\text{BASE}}$*	11.82	81.1/88.5	83.9	92.9