Session4: AI for Continual Evolution

- **CMT: A Memory Compression Method for Continual Knowledge Learning of Large Language Models**
  - 通过记忆压缩、存储与聚合，在冻结LLM参数的情况下，实现高效的 LLM 持续学习。
- **RAP: Retrieval-Augmented Planning with Contextual Memory for Multimodal LLM Agents**
  - 通过存储过往经验并基于与当前情境的相似性进行智能检索，用于增强 LLM 智能体的规划能力。
- **AlphaEvolve: A coding agent for scientific and algorithmic discovery**
  - 通过自动化流程，让 LLM 不断地修改和改进代码，并通过持续的评估反馈，迭代出更优秀的算法。
- **Ubiquitous memory augmentation via mobile multimodal embedding system**
  - 通过模仿人脑“先记要点，按需回忆”机制，在资源受限设备上，高效率构建个人“记忆宫殿”。

东南大学
SOUTHEAST UNIVERSITY

计算机科学与工程学院
School of computer science and engineering

CMT: A Memory Compression Method for Continual Knowledge Learning of Large Language Models

Dongfang Li, Zetian Sun, Xinshuo Hu, Baotian Hu, Min Zhang

Harbin Institute of Technology (Shenzhen)

AAAI'25

汇报人：刘佳伟

2025 年 9 月 20 日

提纲

研究背景与动机

- **LLM的根本性缺陷：静态知识困境**
  - 训练数据是静态的
  - 重新训练成本高

- GPT-3 的训练数据大约截止到 **2021 年**
- GPT-3 拥有 **1760 亿**参数
- 从头训练一次，大约需要 **3640 PF-days** 的算力

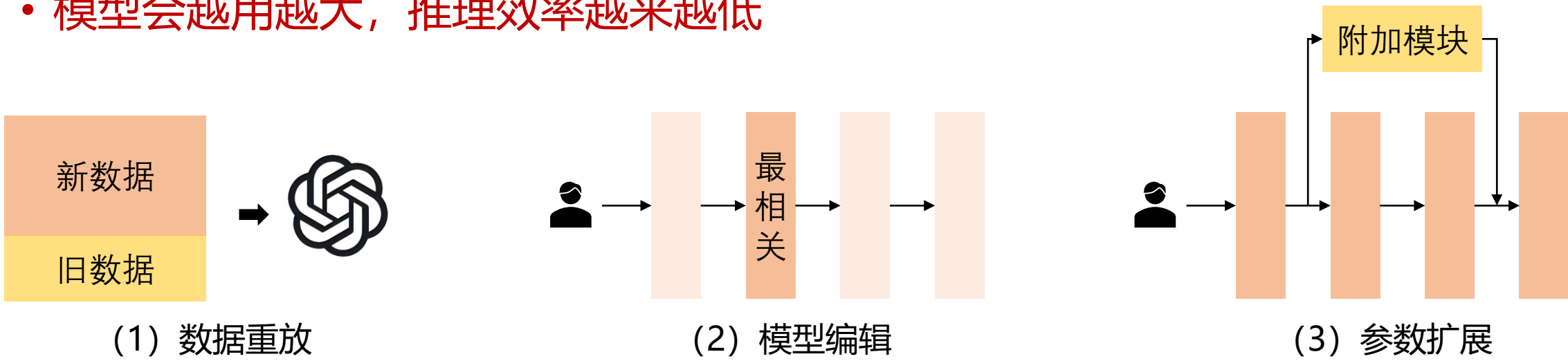
4090（FP32）≈ 82.6 TFLOPS

单精度单卡4090需要计算： $3640 \times 10^{15} \times 86400 \div 82.6 \div 10^{12} \div 86400 \div 365 \approx$ **120.7**年

$1\,P = 1000\,T = 10^{15}$

研究背景与动机

- **现有的静态知识困境解决方案：**
- **（1）数据重放：**利用新数据训练模型的同时，混合一部分旧的训练数据
  - 数据存储和训练成本非常高
- **（2）模型编辑：**直接定位并修改模型中与特定知识相关的极少数参数
  - 见效快，但会引发灾难性遗忘
- **（3）参数扩展：**为模型附加一些小的、可训练的额外参数模块
  - 模型会越用越大，推理效率越来越低

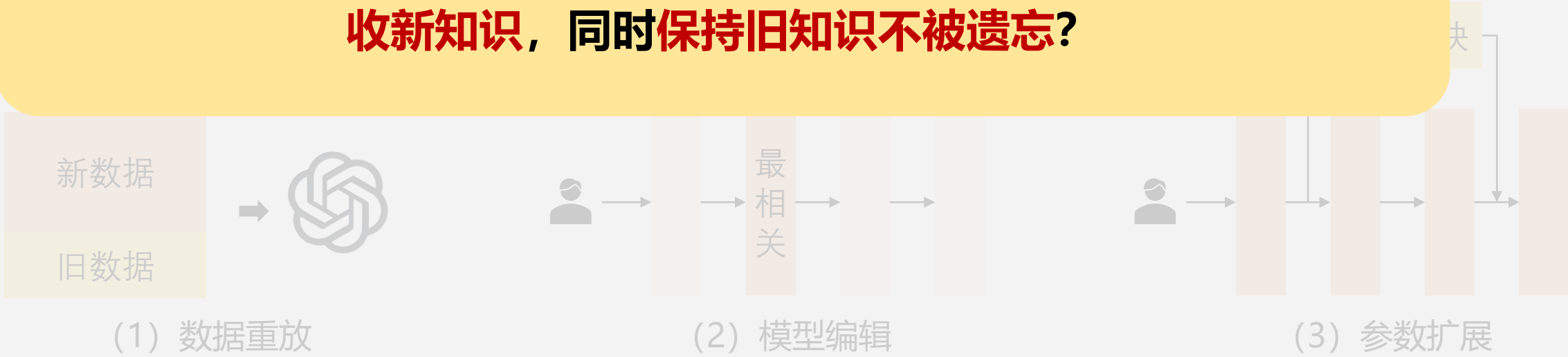


（1）数据重放　　（2）模型编辑　　（3）参数扩展

研究背景与动机

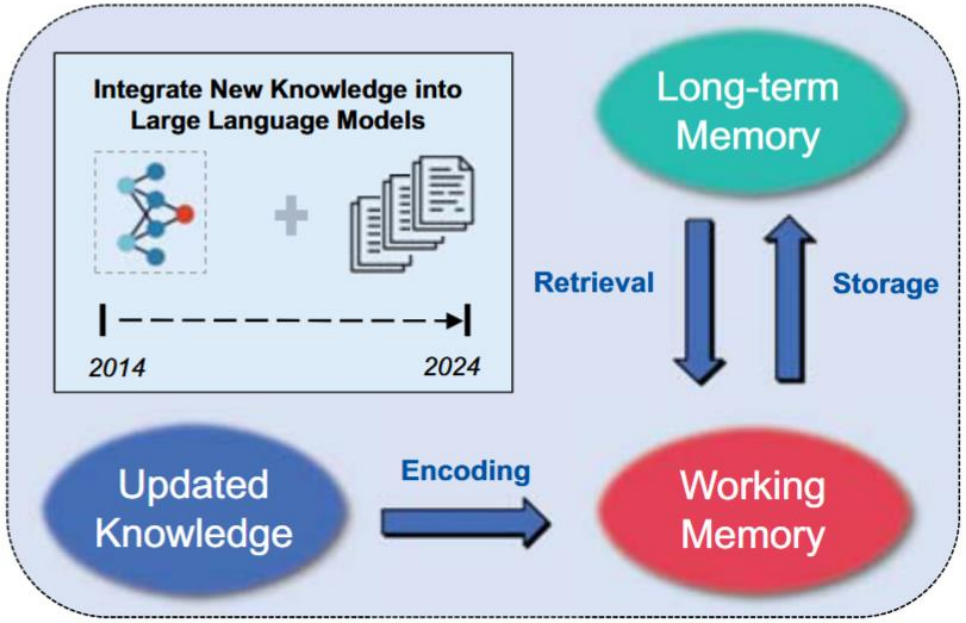
- **现有的静态知识困境解决方案：**
- **（1）数据重放：**利用新数据训练模型的同时，混合一部分旧的训练数据
  - 数据存储和训练成本非常高
- **（2）模型编辑：**直接定位并修改模型中与特定知识相关的极少数参数
-

能不能在不修改模型参数的情况下，让大语言模型像人类一样，逐步吸收新知识，同时保持旧知识不被遗忘？



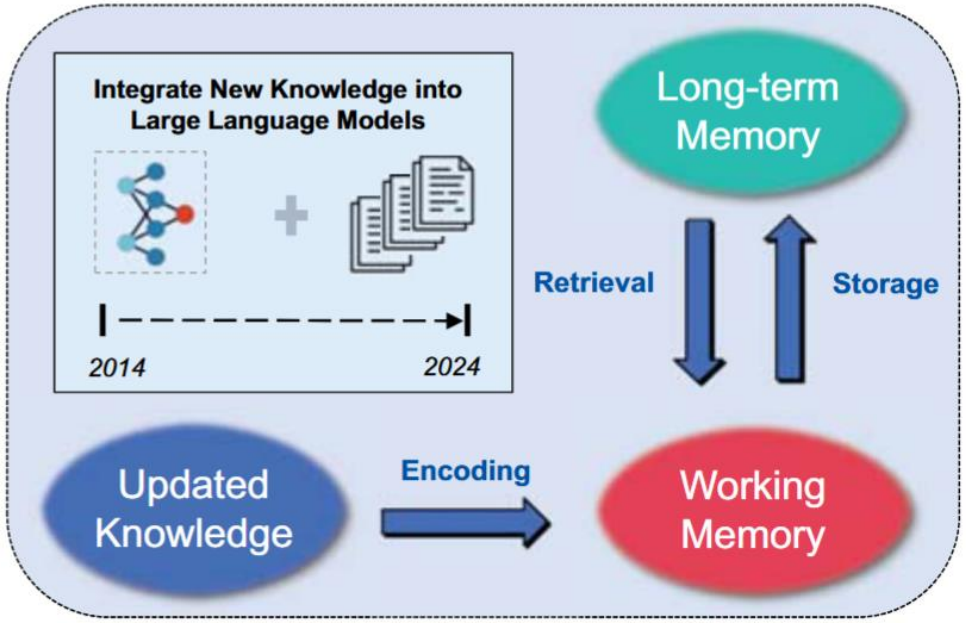
研究背景与动机

- **人类记忆可以分为三个主要阶段：**
  - **编码：**对新信息的理解和抽象，把重要的内容压缩成大脑可以存储的知识结构
  - **存储：**把这些信息放进长期记忆，并且建立一定的结构化组织
  - **检索：**直接从记忆中调用关键事实，并结合上下文来回答问题



研究背景与动机

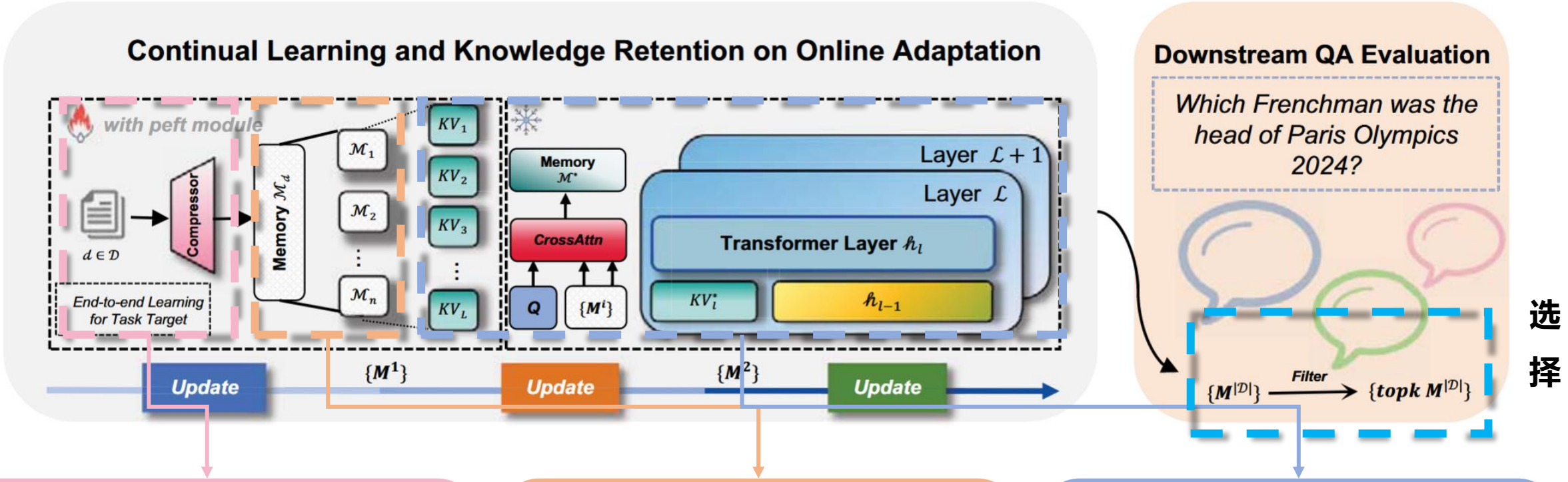
- **人类记忆可以分为三个主要阶段：**
  - **编码：**对新信息的理解和抽象，把重要的内容压缩成大脑可以存储的知识结构
  - **存储：**把这些信息放进长期记忆，并且建立一定的结构化组织
  - **检索：**直接从记忆中调用关键事实，并结合上下文来回答问题



能否把人类记忆机制引入到大语言模型中?

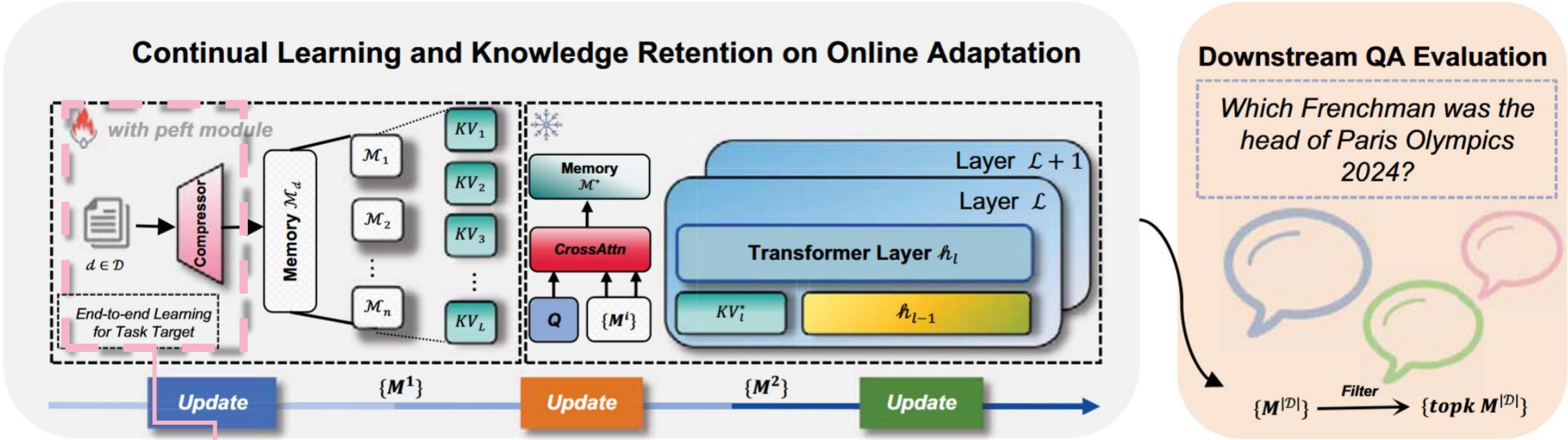
提纲

CMT框架



- 压缩
  - 轻量级压缩模型
  - 长文本->紧凑向量表示
- 存储
  - 可动态扩展
  - 存储开销<<原始文档
- 聚合与对齐
  - 提取最相关信息
  - 对齐LLM，作为KV

压缩



- **压缩**
  - **轻量级压缩模型**
  - **长文本->紧凑向量表示**

- 文档 $d \in \mathcal{D}$ ➡$(w_1, w_2, \ldots, w_n)$ −−> $(c_1, c_2, \ldots, c_k)(k \ll n)$
  原始 压缩
- 压缩 $M = \Theta(d) = (m_1, m_2, \ldots, m_k) \in \mathbb{R}^{k \times d}$
  轻量级压缩模型

压缩

- **压缩与不压缩：数据大小比较**

$$w \in \mathbb{R}^{n\times d}$$

⬇

$$\{w_Q, w_K, w_V\} \in \mathbb{R}^{n\times d}$$

⬇

$$softmax\left(\frac{w_Q {w_K}^{\mathrm{T}}}{\sqrt{\mathrm{d}}}\right) w_V \in \mathbb{R}^{n\times d}$$

原始

$$\theta \in \mathbb{R}^{k\times n}$$

⬇

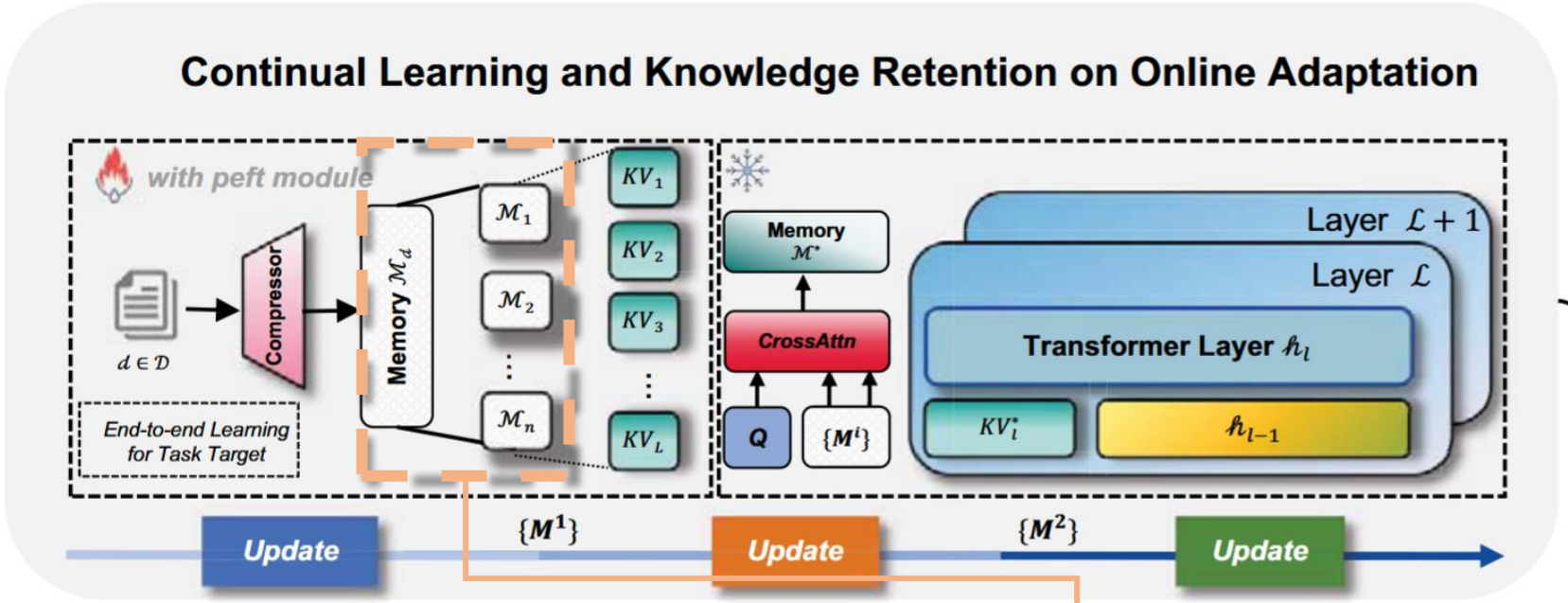
$$\{\theta \cdot w_Q, \theta \cdot w_K, \theta \cdot w_V\} \in \mathbb{R}^{\mathrm{k}\times d}$$

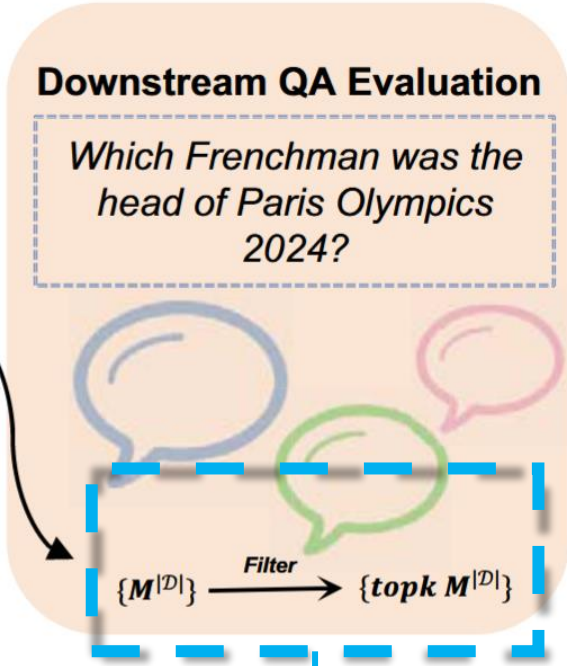
⬇

$$softmax\left(\frac{w_{\mathrm{Q}\theta} {w_{\mathrm{K}\theta}}^{\mathrm{T}}}{\sqrt{\mathrm{d}}}\right) w_{\mathrm{V}\theta} \in \mathbb{R}^{\mathrm{k}\times d}$$

压缩

存储与选择





选择

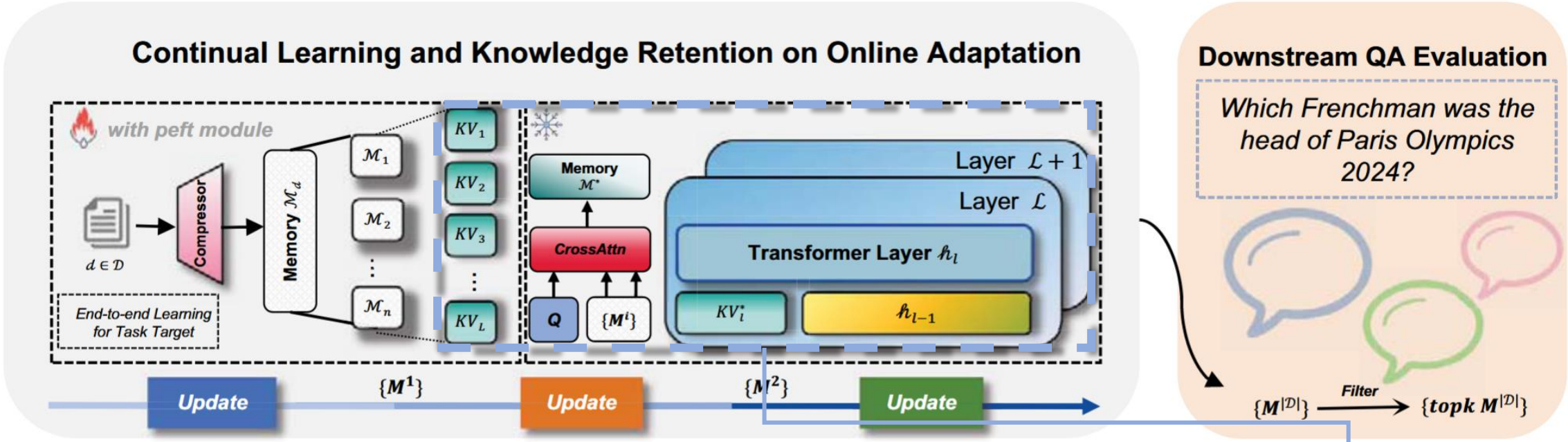
- 训练阶段都是同领域的文档
- 到推理阶段时，涵盖多种领域

←

- 存储
  - 可动态扩展
  - 存储开销<<原始文档

- 选择
  - 计算记忆与问题相似度
  - 选择最相关的K个

聚合与对齐



$$M = (m_1, m_2, \ldots, m_k) \in \mathbb{R}^{k \times d}, \Theta(D) = \{M^i\}_{i=1}^{|D|}$$

$$M^* = \psi\left(\Theta(q_i), \{M^i\}_{i=1}^{|D|}\right) \in \mathbb{R}^{k \times d}$$

- 对于压缩输入q_i，**聚合网络 ψ** 从所有文档（D）中提取最相关信息 M^*

$$KV^* = \pi(M^*)$$

- 利用**对齐网络 π**，将聚合记忆转换为下游LLM所需的KV

- **聚合与对齐**
  - **提取最相关信息**
  - **对齐LLM，作为KV**

聚合与对齐

- **单个文档内，为什么要考虑位置信息?**

  - **RoPE (Rotary Position Embedding)**
    - **顺序不同，计算结果不同**

2月1日南京下雨。2月2日南京下雪。

≠

2月2日南京下雨。2月1日南京下雪。

$$M^* = \psi\left(\Theta(q_i), \{M^i\}_{i=1}^{|D|}\right) \in \mathbb{R}^{k \times d}$$

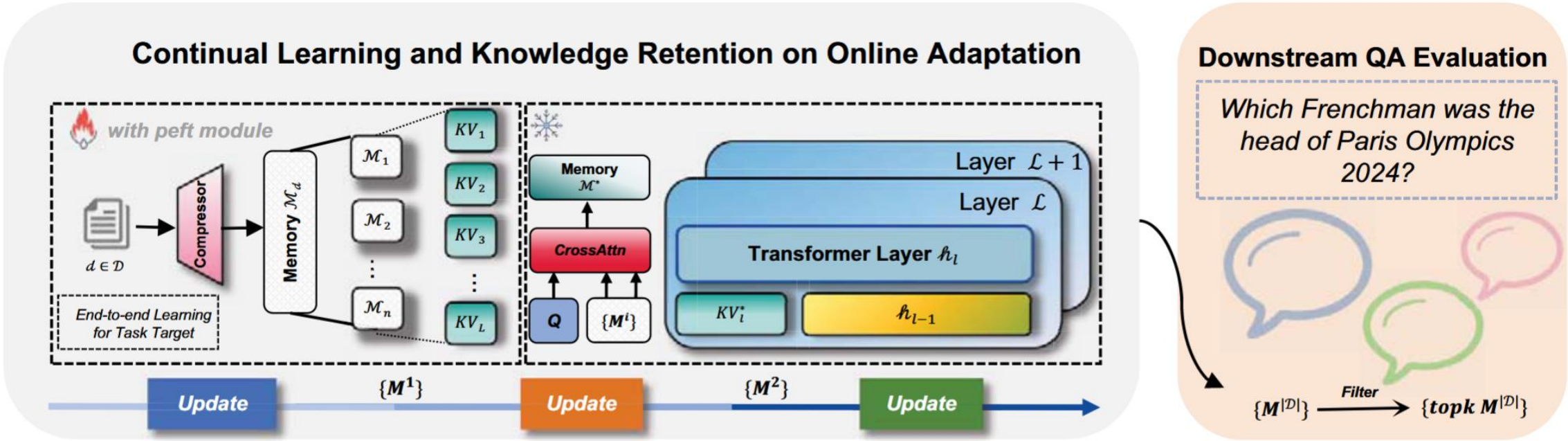
对于D，如何计算?

$$Q \in \mathbb{R}^{k \times d} \qquad \begin{bmatrix} M^1 \\ M^2 \\ M^3 \end{bmatrix} = K^* = V^* \in \mathbb{R}^{3k \times d}$$

$$softmax\left(\frac{w_{Qc} {w_{Kc}}^T}{\sqrt{d}}\right) w_{Vc} \in \mathbb{R}^{k \times d}$$

- **顺序改变，事实也发生了改变**

训练目标



$$Loss = \min_{D_{train}, Q_{train}, Y_{train}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(\mathrm{LLM}(q_i; \pi(M^*)), y_i\right)$$

- **同一领域内，给定文档集合D，问题Q和答案Y，最小化LLM“更新记忆”后的回复与答案Y的差距**

训练目标优化1：新旧记忆的平衡

- **过渡关注记忆会带来负面影响**
  - 原因1：记忆可能与LLM预训练知识冲突
  - 原因2：记忆可能含噪声或偏差
  - 原因3：完全忽略先验会带来灾难性遗忘

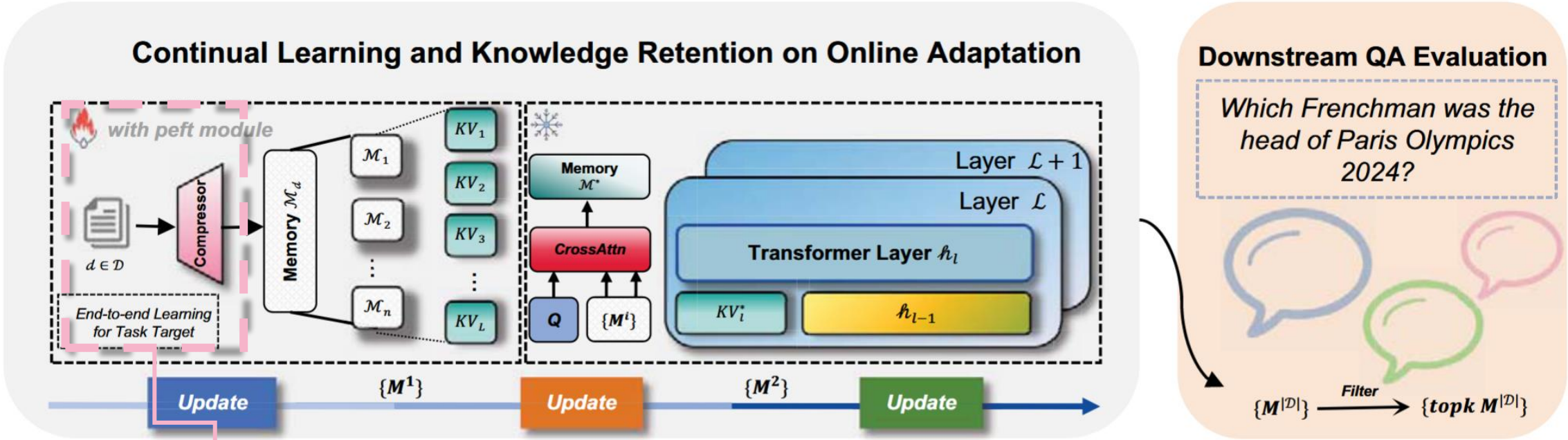
$$Loss = \min_{D_{train}, Q_{train}, Y_{train}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(\mathrm{LLM}(q_i; \pi(M^*)), y_i\right)$$

- **既要利用新记忆，又不能盲信；既要保留老知识，又不能固步自封**

调整 LLM 在 softmax 前的logit (l_θ)：

$$(1+\alpha) l_\theta(y_i \mid M^*, q_i) - \alpha l_\theta(y_i \mid q_i)$$

训练目标优化2：提高压缩记忆的区分度



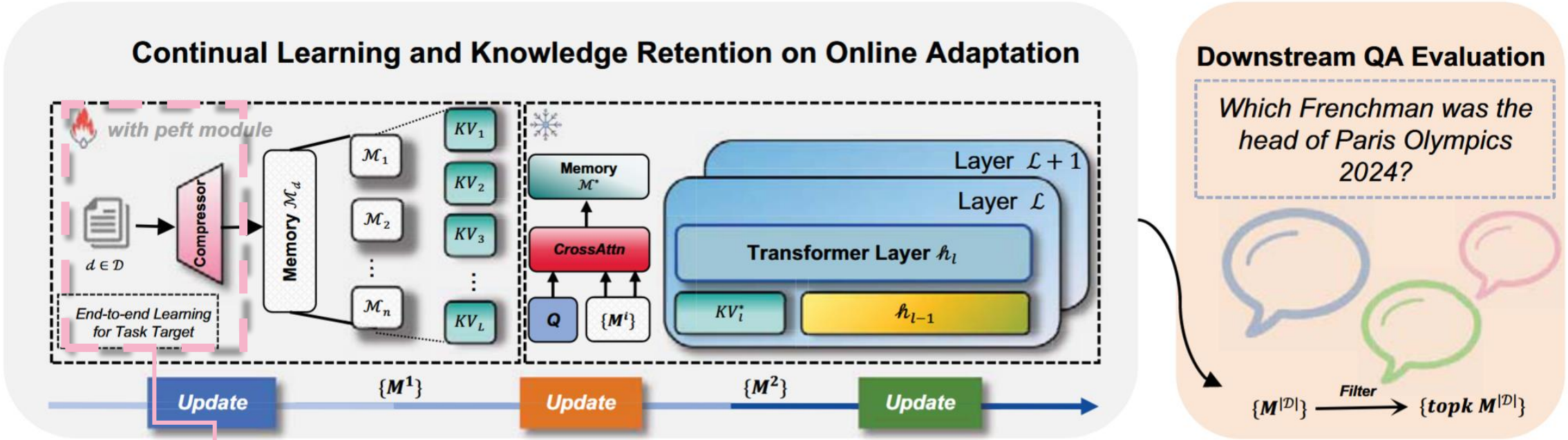
- **压缩**
  - **轻量级压缩模型**
  - **长文本->紧凑向量表示**

- 文档 $d \in \mathcal{D}$ ➡$(w_1, w_2, \ldots, w_n) \dashrightarrow (c_1, c_2, \ldots, c_k)(k \ll n)$
- 压缩 $M = \Theta(d) = (m_1, m_2, \ldots, m_k) \in \mathbb{R}^{k \times d}$

压缩模型Θ偷懒，把整个文档的所有记忆

都压缩成相近甚至完全一致的结果

训练目标优化2：提高压缩记忆的区分度



- **压缩**
  - **轻量级压缩模型**
  - **长文本->紧凑向量表示**

$$\mathcal{L}_{self-match} = \underbrace{-\cos(q_i, M^i)}_{\text{拉近相关记忆}} + \underbrace{\lambda \log \sum_{j \neq i} exp(-|M^i - M^j|^2)}_{\text{推远其他记忆}}$$

- **通过辅助训练目标增强相关记忆的表示，抑制无关记忆，提升模型在持续学习中的知识迁移能力和鲁棒性**

提纲

实验设置

数据集

- **StreamingQA**：新闻事件问答。
- **SQuAD**：维基百科问答。
- **ArchivalQA**：纽约时报的历史档案问答。

模型

- **DistilGPT2-88M**
- **GPT2-Large-774M**
- **GPT2-XL-1.5b**
- **Llama-2-7B**

基线

- **Uniform**：对新文档中的 token 一视同仁，均匀地进行采样和存储，不加任何权重。
- **Salient Spans**：只对包含关键词的 token 赋予权重，而其他 token 不纳入记忆。
- **CaMeLS**：通过额外的神经网络来预测每个 token 的重要性，从而决定哪些内容值得被存入记忆。
- **MAC**：与CMT流程相似，具体实现细节存在差异，如（不压缩记忆，不过滤无关文档等）

评价指标

Exact Match (EM)：预测答案和标准答案是否一模一样（严格，且苛刻）。

F1-Score：Precision-所有预测中，正确的比例（准）；Recall-标准答案中，正确预测的比例（全）。

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

实验结果

- **对比实验**

Datasets	Method	DistilGPT2		GPT2-Large		GPT2-XL		Llama-2	
		EM	F_1	EM	F_1	EM	F_1	EM	F_1
StreamingQA	Uniform	1.62	3.76	4.74	7.00	5.11	7.48	12.43	13.54
	Salient Spans	1.44	4.67	4.86	8.54	5.40	9.42	13.33	18.97
	CaMeLS	1.62	5.79	5.35	10.60	6.55	11.67	-	-
	MAC	5.59	10.18	7.25	13.31	8.99	15.38	14.29	21.79
	CMT (ours)	**6.43**	**12.32**	**7.32**	**13.43**	**9.61**	**16.48**	**18.36**	**25.98**
SQuAD	Uniform	1.24	2.54	3.64	4.97	6.10	6.78	13.25	17.01
	Salient Spans	1.03	2.47	4.03	6.48	4.55	6.74	13.74	18.66
	CaMeLS	1.47	3.08	4.97	8.63	6.70	10.15	-	-
	MAC	2.01	6.85	6.43	11.42	7.10	12.55	15.07	21.14
	CMT (ours)	**3.12**	**7.59**	**7.15**	**12.45**	**9.81**	**12.85**	**19.54**	**25.50**
ArchivalQA	Uniform	4.86	4.08	7.66	8.71	8.61	10.78	18.53	21.35
	Salient Spans	4.52	3.76	9.75	11.19	11.81	14.11	18.97	22.75
	CaMeLS	4.62	6.19	9.92	12.41	13.87	15.74	-	-
	MAC	7.55	10.58	11.84	15.26	14.01	17.12	20.12	23.90
	CMT (ours)	**8.15**	**11.03**	**12.28**	**16.12**	**14.55**	**18.01**	**21.73**	**25.40**

- **表现最出色（EM 提升约 4 个点，F1 提升 4.2 个点）**
- **不仅能吸收新知识，模型规模越大，CMT 的提升越明显**

实验结果

- **消融实验**

#	Method	StreamingQA		SQuAD		ArchivalQA	
		EM	F_1	EM	F_1	EM	F_1
(1)	**CMT**	18.36	**25.98**	**19.54**	**25.50**	21.73	25.40
(2)	w/o Memory-Aware Objective	**18.54**	23.71	15.38	22.77	20.89	24.18
(3)	w/o Self-Matching	17.87	22.54	17.97	23.40	**22.43**	**25.68**
(4)	w/o Top-k Aggregation	16.43	20.13	18.35	24.12	21.09	23.99

- **Top-k Aggregation：下降最明显，如果不筛选，记忆库里的噪声信息会干扰模型。**
- **Self-Matching：不同记忆之间会过度相似，导致检索精度下降。**
- **Memory-Aware Objective：模型学新知识时会出现遗忘加速的情况。**

实验结果

- **能否保留基本能力**

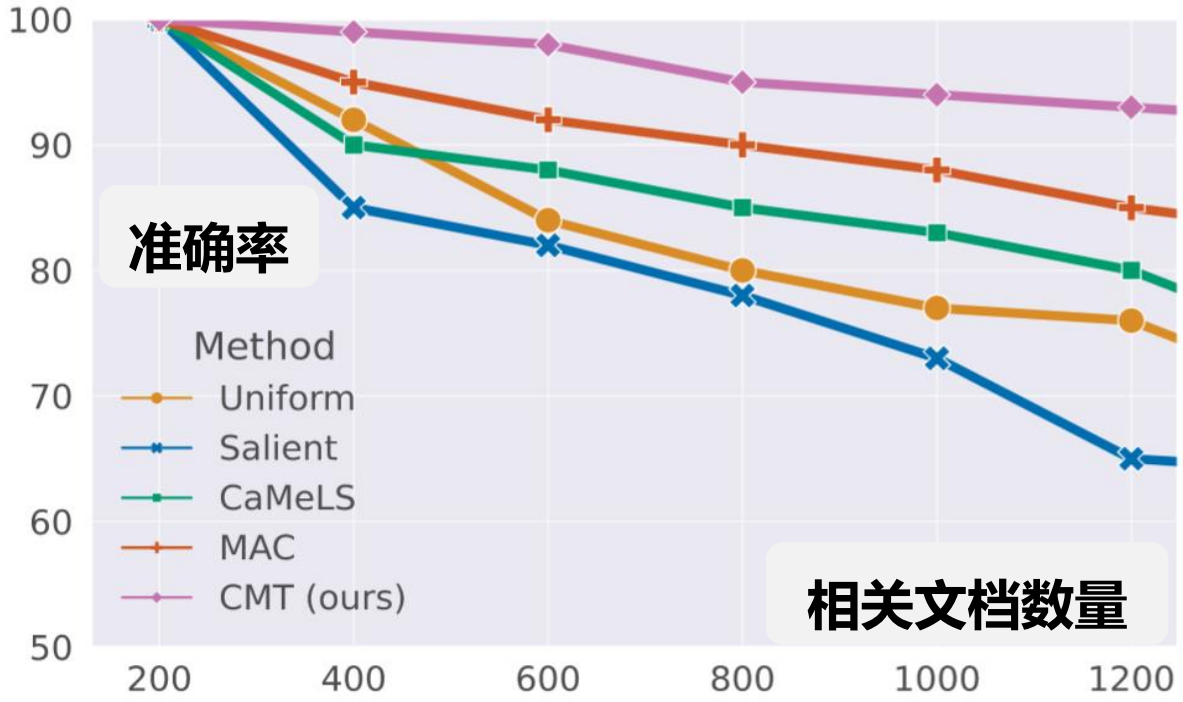


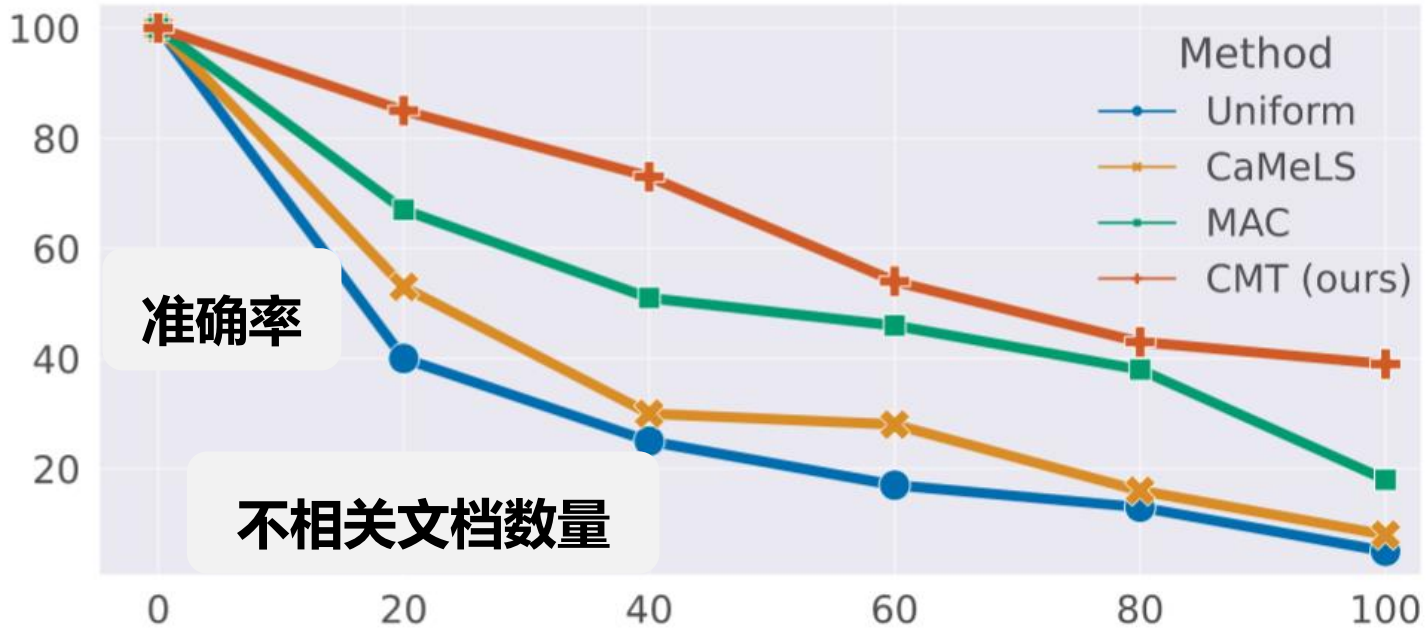
Figure 3: Knowledge retention analysis under `Llama-2-7b` trained on StreamingQA dataset.

不断给模型输入新文档，同时测试它在最早 200 篇文档上的准确率。

- **CMT的准确率下降最慢**
- **MAC次之**

实验结果

- **在“嘈杂环境”中的鲁棒性**



- **CMT的准确率下降最慢**
- **MAC次之**
- **Uniform 和 CaMeLS很快就被误导**

往文档流里插入无关的噪声文档，比例从 0% 到 100%。这些文档和任务完全无关，看看模型会不会被干扰。

提纲

总结

1. 提出了 CMT 框架，通过压缩记忆机制实现 LLM 的持续学习。

2. 设计了三项关键机制 —— Memory-Aware Objective、Self-Matching 和 Top-k Aggregation，它们确保了记忆有效、分散、无噪声。

3. 在多个数据集和不同规模模型上进行了系统实验，结果表明 CMT 在效果和鲁棒性上都显著优于现有方法。

思考

- **能否进一步提高?**
  - **跨任务**能力
  - 动态**记忆管理**
- **能否用到我们场景?**
  - 可以，只要我们也构建基于代码的QA数据集（正确、效率等都可以问），完全可以**用于增强LLM对新代码分析的准确性**
- **能否泛化?**
  - 这是一种**通用的范式**，旨在将相关的外部知识以KV的形式传给LLM。至于选择相关知识、压缩知识、聚合知识、如何传给LLM，都可以自行设计。
  - 因此，这种方法**不限制任务形式，数据模态**，泛化性很高。