KV 嵌入：在仅解码器大语言模型中通过内部键值重路由实现免训练文本嵌入

Yixuan Tang 和 Yi Yang香港科技大学
ytangch@connect.ust.hk, imyiyang@ust.hk

arXiv:2601.01046v1 [cs.CL] 3 Jan 2026

摘要

尽管大语言模型是强大的嵌入骨干网络，但在免训练设置下应用它们仍面临两项结构性挑战：因果注意力限制早期词元获取后续上下文，而下一词元预测目标使表示偏向生成，而非语义压缩。为解决这些局限，我们提出 KV 嵌入，这是一种能够激活冻结的大语言模型潜在表示能力的框架。我们利用这样一个观察：每一层最终 token 的键值（KV）状态编码了序列的压缩视图。通过将这些状态重新路由为前置前缀，我们使所有词元能够在一次前向传播中获取序列级上下文。为确保模型无关适用性，我们提出一种基于内在维度的自动层选择策略。在通义千问、Mistral 和 Llama 骨干网络上进行的 MTEB 评测表明，KV 嵌入的表现优于现有免训练基线方法，最高提升 10%，且在长度达 4,096 个词元的序列上仍保持稳健表现。这些结果表明，内部状态操控为修改输入提供了一种高效替代方案；我们希望本研究能推动人们进一步探索大语言模型内部机制，以开展表示学习。

1 引言

文本嵌入支撑着广泛的自然语言处理应用，从语义搜索到检索增强生成（Lewis 等人，2020年；Song 等人，2025年）。近期研究表明，经过对比目标微调后，仅含解码器的大语言模型可作为强大的嵌入主干网络（Wang 等人，2024年；BehnamGhader 等人，2024年）。然而，这些方法针对每种新的主干网络都需要大量计算资源和大规模数据集，限制了其在快速演进的模型架构中的可扩展性。

一种颇具吸引力的替代方案是无需训练的嵌入提取：直接获取表示，即直接从冻结的大语言模型中获取表示。这种方式无需优化，可立即应用于任何新模型。然而，仅解码器大语言模型的两项结构特性带来了重大挑战。首先，因果注意力将每个词元限制为只能关注前面的位置，造成信息不对称：早期词元无法获知后续上下文（Springer 等人，2025年）。其次，下一个词元预测目标会使最终 token 倾向于预测未来内容，而非提炼输入的语义本质（Jiang 等人，2024年）。因此，标准池化策略要么对不完整的表示取均值（均值池化），要么依赖受预测偏差影响的向量（末词元池化）。例如，考虑短语“河岸”：由于因果掩码，“河岸”中的“岸”在计算其表示时并不知道“河”，因此天然含义模糊。由于均值池化对所有词元状态一视同仁地进行聚合，这种模糊的表示会降低最终嵌入的质量。

现有的免训练方法通过多种策略应对这些局限。为缓解预测偏差，PromptEOL（Jiang 等人，2024年）采用专门设计的提示词，引导最终词元形成语义表示。为解决信息不对称问题，Echo（Springer 等人，2025年）采用输入重复，而令牌前置（Fu 等人，2025年）则插入特殊令牌；二者都旨在为每个词元提供全局上下文。然而，Echo 和令牌前置都依赖输入级修改，因此各自也存在局限：输入重复会使序列长度加倍，导致注意力复杂度呈二次增长，并可能产生“中间信息丢失”效应（Liu 等人，2024年）；令牌前置依赖于模型词汇表之外的特殊令牌，因而会产生不可预测的表示。

我们采用了一种不同的方法。我们不通过修改输入层来重建全局上下文，

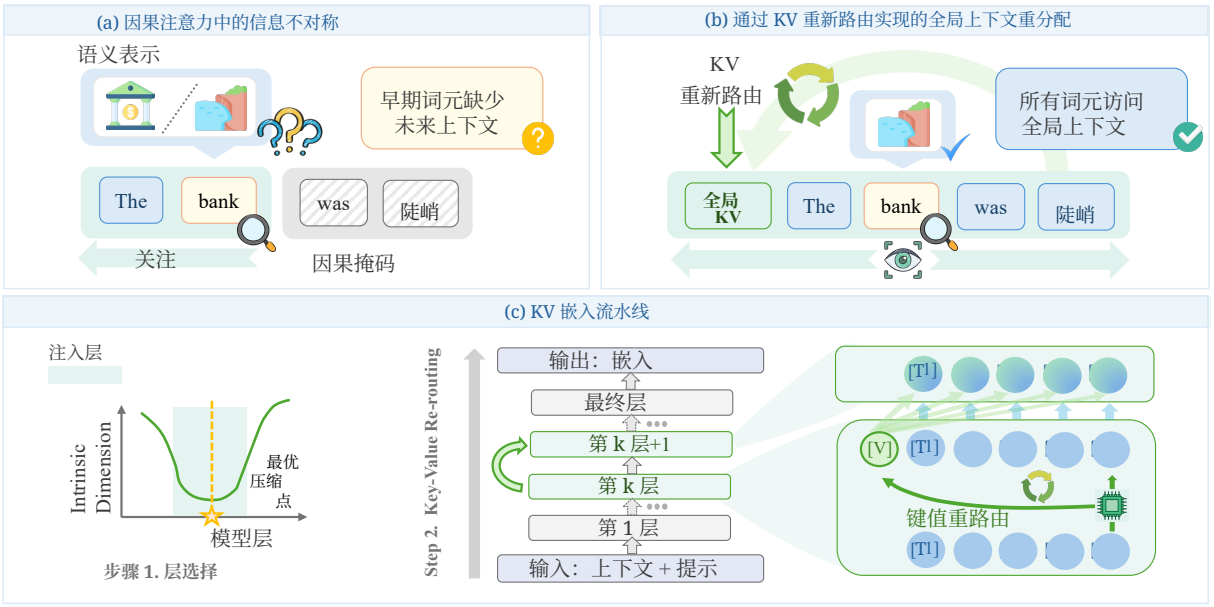

图 1：KV 嵌入概览。(a) 标准因果注意力会造成信息不对称，因为词元只能关注前面的上下文。(b) KV 嵌入将最后一个 KV 对重路由为全局前缀，使模型能够在一次前向传播中访问序列级上下文。(c) 该流程通过定位内在维度最小的层来确定最优重路由锚点，从而确保与模型无关的稳定性，无需手动调参。

而是观察到，模型已经在内部计算了此类全局信息。具体而言，由于因果注意力结构，Radford 等人，2019年）中每一层的最终 token 隐藏状态都会聚合所有前序位置的信息。这引出了一个自然的问题：我们能否直接利用这些在内部计算出的全局表示，来生成质量更高的文本嵌入？

为此，我们提出了**KV 嵌入**，这是一种免训练框架，将最终 token 的键和值（KV）状态重路由为内部前缀。如图 1所示，每一层中最终 token 的 KV 对编码了该层对序列的压缩视图（Katharopoulos 等，2020年；Geva 等，2021年）。通过将该 KV 对置于注意力机制之前，我们使所有词元都能够在单次前向传播中访问全局摘要。我们还采用基于提示的策略，以减轻最终 token 中的预测偏差。这一设计无需将序列长度加倍或引入词表外词元，即可实现序列级上下文访问。

我们进一步提出一种基于内在维度的层选择策略，以确保其适用于不同模型。我们不在每一层都执行重路由，而是识别表示压缩程度达到最大的层，这些层对应于潜在流形维度最小的点。将这些层作为重路由锚点，可以避免干扰早期层的特征提取或晚期层的预测头，从而确保重新分配的上下文与模型的内部几何结构相一致。

在 MTEB（Muennighoff 等人，2023年）和长上下文检索基准 Lo- CoV1（Saad-Falcon 等人，2024年）上的评估证明了 KV 嵌入的有效性。在 MTEB 上，所提出的框架在三个主干网络上均优于现有的免训练方法，最高提升达 10%：通义千问 3-4B（Yang 等人，2025年）、Mistral-7B（Jiang 等人，2023年）以及 Llama-3.1-8B（Dubey 等人，2024年）。在 LoCoV1 上，KV 嵌入在长度最高达 4,096 个词元的序列上仍保持稳健性能，而基线方法则因上下文稀释而性能下降。进一步分析证实，我们的方法能够生成各向同性更强、对齐性和均匀性更佳的嵌入空间。

总而言之，KV 嵌入通过内部 KV 重新路由，在免训练文本嵌入方面树立了新的最先进水平。该框架通过基于内在维度自动选择层，实现了模型无关适用性，无需手动针对不同架构进行调优。大量评估表明，KV 嵌入为利用大语言模型作为文本嵌入模型提供了一种高效且可扩展的解决方案。

2 相关工作

所提出的框架建立在三个研究方向之上：基于大语言模型的文本嵌入、免训练表示提取，以及注意力机制中的内部状态操控。

基于大语言模型的文本嵌入 文本嵌入已从基于编码器的模型发展到仅解码器大语言模型。早期探索如 SGPT（Muennighoff，2022年）采用位置加权池化，而后续模型如 E5-Mistral（Wang 等人，2024年）和 GTE（Li 等人，2023年）则通过在大规模合成数据集上进行对比微调，取得了最先进的结果。为克服因果因果掩码固有的局限性，LLM2Vec（BehnamGhader 等人，2024年）和 NV-Embed（Lee 等人，2025年）在训练期间修改注意力掩码，以支持双向上下文。GritLM（Muennighoff 等人，2025年）进一步在单一架构中统一了生成与嵌入能力。尽管这些基于训练的方法能够产生强大的表示，但其对大量计算资源和专用数据集的需求，限制了它们在快速扩展的大语言模型领域中的可扩展性。因此，有必要开发无需训练的方法，以便在不进行额外优化的情况下利用新模型。我们的工作属于这一类，重点是通过内部状态操控，释放冻结的大语言模型的表示能力。

免训练大语言模型嵌入免训练方法旨在不更新参数的情况下，从冻结的大语言模型中提取高质量嵌入。然而，这种设置受限于仅解码器架构的结构特性：下一词元预测目标使最终状态倾向于未来预测，而非语义概括（Jiang 等人，2024年），因果注意力造成的信息不对称，使早期词元无法获知后续上下文（Springer 等人，2025年）。针对前者，PromptEOL（Jiang 等人，2024年）和 MetaEOL（Lei 等人，2024年）设计了专门的提示，引导模型进行语义压缩。针对后者，Echo（Springer 等人，2025年）通过重复输入序列来模拟双向性，但代价是序列长度加倍。Token Prepending（Fu 等人，2025年）通过传播辅助词元来降低这一开销，但依赖词表外词元，可能导致表示不可预测。在此相反，我们的方法在模型原生注意力机制内重新分配内部状态，使整个序列都能访问上下文，同时无需修改输入或引入外部词元。

注意力中的 KV 操作 在 Transformer 架构中，查询、键和值之间的交互决定了信息在序列中的流动（Vaswani 等人，2017年）。近期的可解释性研究表明，KV 对可以作为一种联想记忆，其中键编码寻址模式，而值存储语义内容（Geva 等人，2021年）。关于信息流的研究进一步表明，在前向传播过程中，全局上下文会累积在特定锚定词元的 KV 状态中（Wang 等人，2023年）。这为内部状态操作提供了一个自然切入点。前缀调优（Li 和 Liang，2021年）将可学习的 KV 对置于序列前端，作为一种参数高效的微调策略；而 StreamingLLM（Xiao 等人，2024年）和 H2O（Zhang 等人，2023年）等 KV 缓存压缩方法则有选择地保留条目，以优化推理效率。这些方法要么需要训练，要么针对生成吞吐量。相比之下，KV 嵌入利用 KV 操作进行表示提取。通过将最终 token 的 KV 状态重新路由为所选层中的前缀，我们的框架使早期词元能够访问压缩后的序列摘要，从而在单次前向传播中解决因果注意力的局限性。

3 方法

我们提出 KV 嵌入，这是一种无需训练的框架，用于从仅解码器大语言模型中提取高质量文本嵌入。该流程如图所示1。

3.1 预备知识

给定输入序列 $X = (x_1, \ldots, x_n)$，仅解码器大语言模型通过 L 顺序层生成隐藏状态 $\mathbf{h}_i^{(l)} \in \mathbb{R}^d$。对于每一层 l，隐藏状态 $\mathbf{h}_i^{(l)}$ 通过多头因果自注意力计算。为简单起见，我们描述单头案例：

$$\mathbf{h}_i^{(l)} = \text{Attention}\left(\mathbf{q}_i^{(l)}, \mathbf{K}_{\leq i}^{(l)}, \mathbf{V}_{\leq i}^{(l)}\right) \quad (1)$$

其中，$\mathbf{K}_{\leq i}^{(l)} = [\mathbf{k}_1^{(l)}, \ldots, \mathbf{k}_i^{(l)}]$ 和 $\mathbf{V}_{\leq i}^{(l)} = [\mathbf{v}_1^{(l)}, \ldots, \mathbf{v}_i^{(l)}]$ 表示键矩阵和值矩阵

来自前面位置的内容。这种因果约束确保每个 $\mathbf{h}_i$ 的计算都不会看到后续词元 $x_{j>i}$。为了从冻结的大语言模型中提取固定维度的嵌入 $\mathbf{e}$，我们将池化函数应用于最后一层的隐藏状态：

$$\mathbf{e} = \text{Norm}\left(P\left(\mathbf{h}_1^{(L)}, \ldots, \mathbf{h}_n^{(L)}\right)\right) \tag{2}$$

其中，$P(\cdot)$ 表示池化策略，$\text{Norm}(\cdot)$ 表示 ℓ_2 归一化。

3.2 KV 嵌入

KV 嵌入通过三个组件解决因果注意力的结构性限制：面向压缩的提示、内部 KV 重路由机制，以及自动化层选择策略。

3.2.1 面向压缩的提示设计

为缓解固有的下一词元预测偏差，我们将输入文本置于一个旨在触发语义压缩的模板中：

> *"{Context/Query}: {text}"*用一个词压缩*{Context/Query}*:

文档的前缀设为"上下文"，查询的前缀设为"查询"。该模板利用模型的生成能力，引导最终词元表示 $\mathbf{h}_n$ 提炼输入的语义精髓。

3.2.2 键值重路由

提示虽能塑造优化目标，却无法解决单向信息流问题：早期词元仍无法访问后续上下文。为此，我们在每个选定层中，将最终词元的键值状态重路由为全局前缀。

在每一层 $l \in \mathcal{L}$，我们首先计算所有位置的键和值投影，得到完整矩阵 $\mathbf{K}^{(l)}$ 和 $\mathbf{V}^{(l)}$。随后，我们提取最终位置上的键值对（$\mathbf{k}_n^{(l)}$, $\mathbf{v}_n^{(l)}$）；由于因果结构，该键值对编码了该层对整个序列的汇总视图。计算注意力之前，我们将此键值对前置到键值矩阵中：

$$\begin{aligned}\tilde{\mathbf{K}}_{\leq i}^{(l)} &= \left[\mathbf{k}_n^{(l)} \| \mathbf{K}_{\leq i}^{(l)}\right] \\ \tilde{\mathbf{V}}_{\leq i}^{(l)} &= \left[\mathbf{v}_n^{(l)} \| \mathbf{V}_{\leq i}^{(l)}\right]\end{aligned} \tag{3}$$

其中 $\|$ 表示沿序列维度进行拼接。此操作会独立应用于每个注意力头。增强后的矩阵使每个查询 $\mathbf{q}_i$ 都能关注虚拟位置 0 处的全局摘要，从而使早期词元能够在单次前向传播中获取序列级上下文。这两个组成部分都至关重要：键用于基于内容的寻址，而值则承载聚合语义信息。我们通过附录 B中的探测实验，实证验证最终 token 的键值状态能够捕获序列级语义。完整算法见附录 A。我们还引入了注意力偏置 b，在关注重新路由位置时将其加到 Softmax 前的逻辑值上，以控制每个词元关注全局摘要的程度。我们根据消融研究设定 $b = 1.0$。详细公式见附录 G。

3.2.3 基于内在维度的自动化层选择

层的功能专门化要求谨慎选择重路由集合 $\mathcal{L}$。浅层编码表层模式，而最终层则偏向词汇预测（Skeanet al.，2025年）。为识别语义密度较高的层，我们借助内在维度（ID），遵循内在维度达到最小值时语义抽象达到最大值这一观察结果（Va-lerianiet al.，2023年）。我们观察到，内在维度轨迹因架构而异（见附录 I），因此需要采用自适应选择策略。

我们使用 theTwoNN 估计器（Facco 等人，2017年），基于 F2LLM（Zhang 等人，2025年）中的 1,000 个句子计算各层的 ID。对于呈现明显 U 形轨迹的模型，我们找出 l^* ID 最小的层，并定义：

$$\mathcal{L} = \{l \mid l^* \leq l \leq \min(L, l^* + \lfloor 0.1L \rfloor)\} \tag{4}$$

由此得到一个从压缩峰值开始的连续窗口。对于存在多个局部最小值的模型，我们改为选择网络中部至后部低 ID 区域内的层，并排除最前面的 $\lfloor 0.2L \rfloor$ 层，这些层主要编码表层特征。此策略旨在选取表示压缩程度最高的层，同时避免早期层噪声和后期预测偏差。我们在附录 C 中提供实证验证。

最后，通过对最终层隐藏状态进行均值池化和最后词元池化并取平均，得到序列嵌入$\mathbf{e}$，将各位置的分布证据与

全局信息引导的最终状态相结合，随后进行 ℓ_2 归一化。

4 实验

本节在多个基准测试和模型架构上评估 KV 嵌入，以展示其作为一种免训练框架在生成高质量文本表示方面的有效性。

4.1 实验设置

基准测试。 我们在两个互补的基准测试上评估 KV 嵌入。MTEB（Muen- nighoff 等人，2023年）涵盖七个类别，提供多任务评估：语义文本相似度、检索、分类、成对分类、聚类、重排序和摘要。为评估模型在长上下文场景中的鲁棒性，我们使用 LoCoV1（Saad- Falcon 等人，2024年），并将文档截断至 1024、2048 和 4096 个词元。这为 KV 重新路由提供了压力测试，因为因果注意力通常会在长序列中受到上下文稀释的影响。有关这些基准测试的任务详细说明和数据集统计信息，请参见附录 D。

模型。 为展示我们方法与模型无关的特性，我们在三种广泛使用的仅解码器大语言模型上进行实验，涵盖不同的参数规模和模型系列：Qwen3-4B（Yang 等人，2025年）、Mistral-7B-Instruct-v0.1（Jiang 等人，2023年）以及 Llama-3.1-8B-Instruct（Dubey 等人，2024年）。所有评估均在零样本、免训练的设置下进行。

基线方法。 我们将 KV 嵌入与具有代表性的免训练策略进行比较。最后一个词元使用最终隐藏状态 $\mathbf{h}_n$，而均值池化则对所有隐藏状态取平均。PromptEOL（Jiang 等人，2024年）通过模板提示引导语义压缩。Echo（Springer 等人，2025年）重复输入序列以模拟双向性，使序列长度加倍。令牌前置（Fu 等人，2025年）通过特殊令牌传播上下文，从第 8 层开始，并从第 6 层至最终层提取表示。我们还报告了一项消融实验（不使用 *KV 重新路由*），该实验使用我们的提示和池化策略，但禁用键值重新路由组件。除非另有说明，所有方法均从最终层提取表示。

实现。我们使用 TwoNN（Facco 等人，2017年）对来自 F2LLM 的 1,000 个句子（Zhang 等人，2025年）。对于在中间层呈现明显内在维度极小值的 Qwen3-4B 和 Mistral-7B-Instruct-v0.1，我们对连续区域应用 KV 重新路由：Qwen3-4B 的第 12–21 层，以及 Mistral-7B-Instruct-v0.1 的第 13–19 层。Llama-3.1-8B 的轨迹更加复杂，存在多个局部最小值；我们从低内在维度区域中选取层，但排除较浅的极小值（第 10–11、20、26–31 层）。MTEB 的最大序列长度为 512。实验在 4 块 NVIDIA H800 GPU 上进行，批量大小为 64，随机种子为 42。

4.2 结果

表1 和表 2 分别展示了 MTEB 和 LoCoV1 上的平均结果。详细数据见附录 E和F。

MTEB 结果。 如表1所示，KV-Embedding 在三个骨干网络上始终优于所有基线方法。与 PromptEOL 和 Echo 相比，我们的方法在保持原始序列长度不变的同时取得了更高的性能。为区分各组件的贡献，我们在保留面向压缩的提示的同时，对 KV 重新路由进行消融实验。消融实验（不使用 KV 重新路由）导致所有模型的性能均有所下降：在 Mistral-7B 上，平均分数从 0.534 降至 0.450。这证实了提示虽能缓解预测偏差，但 KV 重新路由才是性能提升的主要驱动因素。

不同任务类别上的影响。 性能提升因任务类型而异。需要整体语义理解的任务提升显著；尤其是检索，增幅最大，在 Qwen3-4B 上达到 0.2765，而 PromptEOL 为 0.1857。这表明，文档级匹配能够受益于获取整个序列中的全局信息。消融实验也证实了这一规律：移除重新路由同样会导致检索任务的性能降幅最大。语义文本相似度和聚类也呈现出稳定的提升，而摘要任务在不同方法间的差异最小，所有方法的表现都处于相近范围内。

长上下文评测。表2展示了 LoCoV1 在不同序列长度下的结果。KV 嵌入在所有测试长度下均显著优于所有基线方法。在 Mistral-7B 上，我们的方法在所有

方法	STS	检索	分类	配对	聚类	重排序	摘要	Avg.
				Qwen3-4B				
最后一个词元	0.2818	0.0722	0.4243	0.3054	0.2395	0.3476	0.2341	0.2721
均值池化	0.4571	0.0294	0.4656	0.5269	0.3305	0.3272	0.1949	0.3331
PromptEOL	0.6741	0.1857	0.6138	0.5637	0.3651	0.4924	0.2396	0.4478
Echo	0.6634	0.1727	0.5761	0.6367	0.3530	0.4315	0.2294	0.4375
令牌前置	0.6370	0.1622	0.6283	0.5206	0.3372	0.4901	0.2483	0.4320
KV 嵌入	**0.7141**	**0.2765**	**0.6375**	**0.6800**	**0.3903**	**0.5007**	**0.2566**	**0.4937**
不使用 *KV* 重新路由	0.5631	0.2023	0.5094	0.5296	0.3769	0.4600	0.2257	0.4096
				Mistral-7B-Instruct-v0.1				
最后一个词元	0.3915	0.1036	0.5314	0.3440	0.2516	0.3673	0.2823	0.3245
均值池化	0.4885	0.0886	0.5157	0.5681	0.3472	0.3790	0.2557	0.3775
PromptEOL	0.6953	0.1746	0.6571	0.5629	0.3158	0.4835	0.2863	0.4537
Echo	0.7333	0.2414	0.6398	0.7560	0.3681	0.4751	0.2922	0.5008
令牌前置	0.6775	0.1619	0.6580	0.5764	0.2825	0.4790	0.2921	0.4468
KV 嵌入	**0.7720**	**0.3014**	**0.6951**	**0.7564**	**0.3902**	**0.5145**	**0.3088**	**0.5341**
不使用 *KV* 重新路由	0.6478	0.2014	0.6008	0.5931	0.3611	0.4804	0.2664	0.4502
				Llama-3.1-8B-Instruct				
最后一个词元	0.3513	0.0887	0.4769	0.3554	0.2652	0.3608	0.2526	0.3073
均值池化	0.4702	0.0713	0.4848	0.5176	0.3577	0.3537	0.1996	0.3507
PromptEOL	0.6919	0.2017	0.6250	0.5911	0.4035	0.4991	0.2788	0.4702
Echo	0.7124	0.2941	0.6261	0.7125	0.4189	0.5074	0.2527	0.5034
令牌前置	0.6968	0.1817	0.6577	0.5734	0.3365	0.4992	**0.2880**	0.4619
KV 嵌入	**0.7398**	**0.3079**	**0.6602**	**0.7332**	**0.4448**	**0.5237**	0.2796	**0.5270**
不使用 *KV* 重新路由	0.6187	0.2088	0.5611	0.5839	0.4076	0.5044	0.2746	0.4513

表1：MTEB 上的结果。每项数据报告相应任务类别的主要指标，并对各数据集的结果取平均值。KV 嵌入在所有骨干网络上均取得最佳整体性能。不使用 *KV* 重新路由表示在保留压缩提示的同时消融重新路由。每个模型的最高平均分数以**粗体**标出。

设置下的得分均高于 0.18，而基线方法无论序列长度如何，得分都低于 0.10。通义千问 Qwen3-4B 和 Llama-3.1-8B 也呈现出类似趋势：KV 嵌入始终比表现最好的基线方法高出 1.3–3.5× 倍。这些结果表明，基线方法普遍难以胜任长上下文检索，这可能是因为在因果注意力机制下，最终 token 无法有效聚合来自远距离位置的信息。KV 嵌入通过显式地将全局摘要重新路由至所有位置，实现了更有效的长距离信息整合。

5 分析与消融实验

在本节中，我们开展了一系列实验，以探索 KV 嵌入框架、验证其核心机制，并分析由此产生的嵌入空间质量。

5.1 机制验证

简单移除掩码的风险 一个自然的问题是，单纯移除因果掩码是否就能实现全局上下文访问。如表4所示，这种方法行不通：应用双向注意力会导致性能持续下降，在 Qwen3-4B 和 Mistral-7B-Instruct-v0.1 上的得分甚至低于简单的最后一个 token 基线。出现这种情况，是因为仅解码器大语言模型在预训练期间针对单向模式进行了优化；强制词元关注未来位置会引入分布外状态，导致模型无法有效处理。相比之下，KV 嵌入会重新路由现有的 KV 状态，而非改变注意力结构，因此在解决信息不对称的同时保留了模型的内部流形。更多分析见附录K。

5.2 嵌入分析

注意力模式可视化 表3 比较了不同方法下最终 token 的注意力分布。Echo 表现出明显的近因偏差，主要关注最后几个词元。PromptEOL 和令牌前置能够捕捉特定关键词，但经常遗漏“Looking”等表达初始意图的词语。在

方法	1024	2048	4096
Qwen3-4B			
PromptEOL	0.0413	0.0438	0.1286
Echo	0.0961	0.0658	0.1318
令牌前置	0.0414	0.0422	0.1255
KV 嵌入	**0.1289**	**0.1192**	**0.1822**
Mistral-7B-Instruct-v0.1			
PromptEOL	0.0414	0.0407	0.0442
Echo	0.0627	0.0549	0.0591
令牌前置	0.0416	0.0422	0.0992
KV 嵌入	**0.2165**	**0.1838**	**0.2068**
Llama-3.1-8B-Instruct			
PromptEOL	0.0463	0.0410	0.1285
Echo	0.1128	0.0385	0.1191
令牌前置	0.0421	0.0422	0.1397
KV 嵌入	**0.2191**	**0.1817**	**0.2404**

表2：不同上下文长度下 Lo-CoV1 上的检索性能（NDCG@10）。无论序列长度如何，KV 嵌入始终优于所有基线方法。

这些关键词方面。相比之下，KV 嵌入与查询展现出更加结构化的对齐；它同时为表达意图的词语“Looking”以及“examples”和“beginners”等功能性锚点赋予更高权重。这种模式表明，重新路由全局上下文能够使最终状态从不同位置聚合关键语义成分，从而克服因果注意力的局限。

嵌入空间质量表5 展示了 Mistral-7B-Instruct 的对齐度和均匀性指标（Wang 和 Isola，2020年）。对齐度评估相似句子对的嵌入之间的接近程度，而均匀性则评估特征在超球面上的分布。与基线方法相比，KV 嵌入在这两项指标上均取得了最佳值。均匀性的提升表明，内部 KV 重新路由有助于缓解仅解码器大语言模型固有的各向异性问题：其表示往往聚集在狭窄的锥形区域内。通过重新分配全局上下文，KV 嵌入促成了更具区分性且各向同性的嵌入空间。关于几何分析的详细实验设置见附录 H。

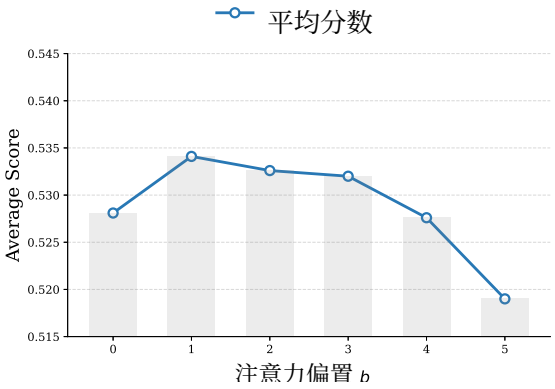

图2：Mistral-7B-Instruct 使用 KV 嵌入时在不同注意力偏置值下的平均性能。偏置值为 1.0 时对应最高平均分数。

5.3 关键组件消融实验

注意力偏置的影响我们在注意力作用于重新路由位置时，向 Softmax 前的逻辑值添加注意力偏置 b，以控制每个词元关注全局摘要的强度。如图2所示，性能从$b = 0$ 提升至 $b = 1.0$，证实重新路由的上下文能够直接提升嵌入质量。在较大的取值范围内，该曲线保持稳定，表明注入的 KV 对能够自然地融入预训练注意力机制。然而，当 $b > 3.0$时，性能会下降，说明模型过度依赖全局摘要。我们将 $b = 1.0$ 设为该值，以平衡全局上下文访问与局部特异性。详细公式和完整结果见附录 G。

基于ID的层选择的有效性我们将基于ID的策略与均匀选择网络早期、中间或后期三分之一部分的层进行比较。正如附录J中所详述的，在早期层进行重路由的性能最低（通义千问 Qwen3-4B上为0.409），这证实了浅层表示缺乏有效重新分配上下文所需的语义密度。基于ID的策略取得了最高分数（通义千问 Qwen3-4B上为0.494，Mistral-7B上为 0.534），同时使用的层数少于均匀选择方案。在Mistral-7B上，基于ID的选择（7层）优于选择中间三分之一部分的10层方案，表明ID最小值能够有效识别语义抽象程度最高的层。

池化策略 我们评估了不同聚合方法的影响，包括最后词元、均值和混合池化。如 Ap-

方法	带有注意力权重的关键词元
Echo	正在 适合
PromptEOL	正在 适合
令牌前置	正在 适合
KV 嵌入	正在 深度

表 3：搜索查询的注意力分布：“寻找适合初学者的全面机器学习教程，涵盖神经网络、决策树和深度学习算法，并提供实用的 Python 代码示例。”颜色表示注意力权重（白色： <0.012，浅色：0.012-0.055，中等：0.055-0.15，深色：>0.15）。图中显示了关键内容词。

模型	方法	STS	检索	分类	配对	聚类	重排序	摘要	Avg.
Qwen3-4B	最后一个词元 + 双向注意力	0.1794	0.0248	0.3764	0.2214	0.1901	0.3146	0.2530	0.2228
	最后一个词元 + 因果注意力	0.2818	0.0722	0.4243	0.3054	0.2395	0.3476	0.2341	0.2721
	KV 嵌入	**0.7141**	**0.2765**	**0.6375**	**0.6800**	**0.3903**	**0.5007**	**0.2566**	**0.4937**
Mistral-7B-Instruct	最后一个词元 + 双向注意力	0.2255	0.0216	0.4133	0.2748	0.2308	0.3458	0.2746	0.2552
	最后一个词元 + 因果注意力	0.3915	0.1036	0.5314	0.3440	0.2516	0.3673	0.2823	0.3245
	KV 嵌入	**0.7720**	**0.3014**	**0.6951**	**0.7564**	**0.3902**	**0.5145**	**0.3088**	**0.5341**

表 4：注意力约束的影响。移除因果掩码（双向注意力）会导致性能崩溃，而 KV 嵌入能够在不违反因果先验的情况下实现全局上下文访问。

方法	对齐(↓)	均匀性(↓)
PromptEOL	0.7378	-2.2715
Echo	0.6366	-2.0794
令牌前置	0.7505	-2.2233
KV 嵌入	0.6082	-2.3899

表5：使用不同方法对 Mistral- 7B-Instruct-v0.1 进行的对齐和均匀性分析。 ↓ 表示数值越低越好。

附录 M，混合策略对最后词元和均值池化得到的表示取平均，在不同模型和任务中表现出更高的一致性。均值池化往往表现不佳，因为它对所有词元一视同仁，未能优先考虑融合全局信息的最终状态。基于这些发现，我们在主要实验中采用混合方法。

对提示设计的鲁棒性 为评估对提示设计的敏感性，我们测试了五种在前缀和动词选择上各不相同的模板。结果如附录L所示，性能保持在较窄的范围内（0.529–0.540）。这一稳定性表明，KV 嵌入对提示变化相对不敏感，且内部 KV 重新路由可能会降低对特定指令的依赖，从而引出语义表示。

6 结论

我们提出了 KV 嵌入，这是一种免训练框架，通过重新分配仅解码器大语言模型的内部状态来提取文本表示。通过在选定层中将最终 token 的键值状态重新路由为前缀，我们的方法使所有位置都能在一次前向传播中访问压缩的序列摘要，从而有效解决因果注意力固有的信息不对称问题。基于内在维度的自动层选择策略使该方法无需手动调优即可适用于不同模型。在 MTEB 和长上下文检索基准 LoCoV1 上的评估表明，KV 嵌入的表现优于现有免训练方法，最高提升 10%，同时在长度达 4,096 个词元的序列上保持稳健表现。这些结果表明，仅解码器大语言模型蕴含着潜在的表示能力，可以通过操控内部状态而非修改外部输入来释放。我们希望这项工作能推动对大语言模型内部机制的进一步探索，弥合生成式预训练与表示学习之间的差距。

局限性

尽管 KV 嵌入相较于免训练基线方法持续取得改进，仍需注意两个主要局限。首先，与标准池化相比，重新路由机制会增加延迟。其次，作为一种免训练方法，KV 嵌入的表现可能不及在大型数据集上使用对比目标进行微调的监督模型。因此，它可作为资源受限场景中的实用替代方案，而非最先进监督式文本嵌入的全面替代品。

参考文献

Parishad BehnamGhader、Vaibhav Adlakha、Marius Mosbach、Dzmitry Bahdanau、Nicolas Chapados 和 Siva Reddy。2024年。LLM2vec：大语言模型其实是强大的文本编码器。在第一届语言建模会议。

Abhimanyu Dubey、Abhinav Jauhri、Abhinav Pandey、Abhishek Kadian、Ahmad Al-Dahle、Aiesha Letman、Akhil Mathur、Alan Schelten、Amy Yang、Angela Fan 等另外 1 位作者。2024年。Llama 3 模型家族。*arXiv* 电子预印本，第 arXiv–2407 页。

Elena Facco、Maria d'Errico、Alex Rodriguez 和 Alessandro Laio。2017 年。通过最小邻域信息估计数据集的内在维度。《科学报告》，7(1):12140–8。

Yuchen Fu、Zifeng Cheng、Zhiwei Jiang、Zhonghui Wang、Yafeng Yin、Zhengliang Li 和 Qing Gu。2025年。令牌前置：一种无需训练、从大语言模型中提取更优句子嵌入的方法。载于《第*63*届计算语言学协会年会论文集》（第*1*卷：长篇论文），*ACL 2025*，第3168–3181页。计算语言学协会。

Mor Geva、Roei Schuster、Jonathan Berant 和 Omer Levy。2021年。Transformer前馈层是键值记忆。载于《*2021*年自然语言处理实证方法会议论文集，*EMNLP 2021*》，第5484–5495页。计算语言学协会。

Albert Q. Jiang、Alexandre Sablayrolles、Arthur Mensch、Chris Bamford、Devendra Singh Chaplot、Diego de las Casas、Florian Bressress、Gianna Lengyel、Guillaume Lample、Lucile Saulnier、Lélio Renard Lavaud、Marie-Anne Lachaux、Pierre Stock、Teven Le Scao、Thibaut Lavril、Thomas Wang、Timothée Lacroix 和 William El Sayed。2023年。Mistral 7b。预印本，arXiv:2310.06825。

Ting Jiang、Shaohan Huang、Zhongzhi Luan、Deqing Wang 和 Fuzhen Zhuang。2024年。利用大语言模型扩展句子嵌入。在计算语言学协会论文集：*EMNLP*，第 3182–3196 页。计算语言学协会。

Angelos Katharopoulos、Apoorv Vyas、Nikolaos Pap- pas 和 François Fleuret。2020年。Transformer 就是循环神经网络：采用线性注意力的快速自回归Transformer。载于第*37*届国际机器学习会议论文集，*ICML 2020*，机器学习研究论文集第119卷，5156–5165页。PMLR。

Chankyu Lee、Rajarshi Roy、Mengyao Xu、Jonathan Raiman、Mohammad Shoeybi、Bryan Catanzaro 和 Wei Ping。2025年。NV-Embed：将大语言模型训练为通用嵌入模型的改进技术。载于第十三届国际学习表征会议，*ICLR 2025*。OpenReview.net。

Yibin Lei、Di Wu、Tianyi Zhou、Tao Shen、Yu Cao、Chongyang Tao 和 Andrew Yates。2024年。元任务提示可从大语言模型中提取嵌入。载于第*62*届计算语言学协会年会论文集（第*1*卷：长篇论文），*ACL 2024*，10141–10157页。计算语言学协会。

Patrick Lewis、Ethan Perez、Aleksandra Piktus、Fabio Petroni、Vladimir Karpukhin、Naman Goyal、Hein- rich Küttler、Mike Lewis、Wen-tau Yih、Tim Rock- täschel、Sebastian Riedel 和 Douwe Kiela。2020年。面向知识密集型自然语言处理任务的检索增强生成。载于《神经信息处理系统进展》第*33*卷：*2020*年神经信息处理系统年会，*NeurIPS 2020*，*2020*年*12*月*6*日至*12*日，线上。

Xiang Lisa Li 和 Percy Liang。2021年。前缀调优：优化用于生成的连续提示。载于第*59*届计算语言学协会年会和第*11*届自然语言处理国际联合会议论文集，*ACL/IJCNLP 2021*，（第*1*卷：长篇论文），第4582–4597页。计算语言学协会。

Zehan Li、Xin Zhang、Yanzhao Zhang、Dingkun Long、Pengjun Xie 和 Meishan Zhang。2023年。通过多阶段对比学习实现通用文本嵌入。*CoRR*，摘要/2308.03281。

Nelson F. Liu、Kevin Lin、John Hewitt、Ashwin Paranjape、Michele Bevilacqua、Fabio Petroni 和 Percy Liang。2024年。迷失在中间：语言模型如何使用长上下文。计算语言学协会会刊，12:157–173。

Niklas Muennighoff。2022年。SGPT：用于语义搜索的 GPT 句子嵌入。*CoRR*，摘要/2202.08904。

Niklas Muennighoff、Hongjin Su、Liang Wang、Nan Yang、Furu Wei、Tao Yu、Amanpreet Singh 和 Douwe Kiela。2025年。生成式表示指令微调。载于第十三届国际学习表征会议，*ICLR 2025*。OpenReview.net。

Niklas Muennighoff、Nouamane Tazi、Loic Magne 和 Nils Reimers。2023年。MTEB：大规模文本嵌入基准测试。载于第*17*届欧洲计算语言学协会分会会议论文集，第2014–2037页。计算语言学协会。

Alec Radford、Jeffrey Wu、Rewon Child、David Luan、Dario Amodei、Ilya Sutskever 等人。2019年。语言模型是无监督的多任务学习者。*OpenAI*博客，1(8):9。

Jon Saad-Falcon、Daniel Y. Fu、Simran Arora、Neel Guha 和 Christopher Ré。2024年。使用 loco 和 m2-bert 对长上下文检索模型进行基准测试和构建。发表于第*41*届国际机器学习会议论文集，ICML'24。JMLR.org。

Oscar Skean、Md Rifat Arefin、Dan Zhao、Niket Nikul Patel、Jalal Naghiyev、Yann LeCun 和 Ravid Shwartz-Ziv。2025年。逐层解析：揭示语言模型中的隐藏表示。发表于第四十二届国际机器学习会议。

Tingyu Song、Guo Gan、Mingsheng Shang 和 Yilun Zhao。2025年。IFIR：用于评估专家领域信息检索中指令遵循能力的综合基准。在计算语言学协会美洲分会*2025*年会议：人类语言技术论文集，*NAACL 2025 - 第1*卷：长篇论文，阿尔伯克基，新墨西哥州，美国，*2025*年*4*月*29*日至*5*月*4*日，第10186–10204页。计算语言学协会。

Jacob Mitchell Springer、Suhas Kotha、Daniel Fried、Graham Neubig 和 Aditi Raghunathan。2025年。重复可改善语言模型嵌入。在第十三届国际学习表示大会，*ICLR 2025*。OpenReview.net。

Lucrezia Valeriani、Diego Doimo、Francesca Cuturello、Alessandro Laio、Alessio Ansuini 和 Alberto Cazzaniga。2023年。大型Transformer模型隐藏表示的几何结构。在神经信息处理系统进展第*36*卷：*2023*年神经信息处理系统年度会议，*NeurIPS 2023*。

Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Lukasz Kaiser 和 Illia Polosukhin。2017年。《注意力就是你所需要的一切》。载于《神经信息处理系统进展》第*30*卷：*2017*年神经信息处理系统年会，第5998–6008页。

Lean Wang、Lei Li、Damai Dai、Deli Chen、Hao Zhou、Fandong Meng、Jie Zhou 和 Xu Sun。2023年。标签词是锚点：从信息流视角理解上下文学习。载于*2023*年自然语言处理实证方法会议论文集，*EMNLP 2023*，第9840–9855 页。计算语言学协会。

Liang Wang、Nan Yang、Xiaolong Huang、Linjun Yang、Rangan Majumder 和 Furu Wei。2024年。《借助大语言模型改进文本嵌入》。载于《计算语言学协会第*62*届年会论文集》（第*1*卷：长篇论文），*ACL 2024*，第11897–11916页。计算语言学协会。

Tongzhou Wang 和 Phillip Isola。2020年。《通过超球面上的对齐与均匀性理解对比表征学习》。载于《第*37*届国际机器学习会议论文集，*ICML 2020*》，机器学习研究会议录第119卷，第9929–9939页。PMLR。

Guangxuan Xiao、Yuandong Tian、Beidi Chen、Song Han 和 Mike Lewis。2024年。具有注意力汇的高效流式语言模型。载于第十二届国际学习表征会议，*ICLR 2024*。OpenReview.net。

An Yang、Anfeng Li、Baosong Yang、Beichen Zhang、Binyuan Hui、Bo Zheng、Bowen Yu、Chang Gao、Chengen Huang、Chenxu Lv 及另外1位作者。2025年。通义千问3技术报告。*arXiv* 预印本 *arXiv:2505.09388*。

Zhenyu Zhang、Ying Sheng、Tianyi Zhou、Tianlong Chen、Lianmin Zheng、Ruisi Cai、Zhao Song、Yuandong Tian、Christopher Ré、Clark W. Barrett、Zhangyang Wang 和 Beidi Chen。2023年。H2O：用于大语言模型高效生成式推理的重击者预言机。载于神经信息处理系统进展第*36*卷：*2023*年神经信息处理系统年度会议，*NeurIPS 2023*。

Ziyin Zhang、Zihan Liao、Hang Yu、Peng Di 和 Rui Wang。2025年。F2LLM 技术报告：使用 600 万条开放数据匹配最先进的嵌入性能。*CoRR*，摘要/2510.02294。

A 算法细节

算法1概述了完整的键值嵌入流程。该方法只需要一次前向传播，并在选定的层应用键值重新路由 $\mathcal{L}$。

B 探测KV状态中的序列信息

为通过实证验证最终 token 的键值状态比更早位置的状态捕获了更丰富的序列级语义，我们在文本分类任务上进行了一项探测实验。

算法 1KV 嵌入

输入： 文本 x，包含 L 层的大语言模型，层集合 $\mathcal{L}$
输出： 嵌入$\mathbf{e}$

将压缩提示应用于 x
对于 $l = 1, \ldots, L$ **执行**
  $\mathbf{K}^{(l)}$，$\mathbf{V}^{(l)} \leftarrow$ 键值投影
  如果 $l \in \mathcal{L}$ **那么**
    $\mathbf{k}_n$，$\mathbf{v}_n \leftarrow$ 最终位置的键值
    $\tilde{\mathbf{K}}^{(l)} \leftarrow [\mathbf{k}_n \| \mathbf{K}^{(l)}]$
    $\tilde{\mathbf{V}}^{(l)} \leftarrow [\mathbf{v}_n \| \mathbf{V}^{(l)}]$
    注意力$(\mathbf{Q}^{(l)}, \tilde{\mathbf{K}}^{(l)}, \mathbf{V}^{(l)})$
  else
    标准注意力
  结束 if
结束 for
$\mathbf{e}_1 \leftarrow \mathbf{h}_n^{(L)}$，$\mathbf{e}_2 \leftarrow$均值$(\mathbf{H}^{(L)})$
返回 归一化$((\mathbf{e}_1 + \mathbf{e}_2)/2)$

设置我们从 MTEB（Muennighoff 等人，2023年）中选取两个分类数据集：IMDb 分类（二分类情感）和推文情感提取分类（三分类情感）。对于每个数据集，我们抽取 1,000 个训练样本和 500 个验证样本。我们对 Mistral-7B-Instruct （Jiang 等人，2023 年）进行一次前向传播，并从三个位置提取键值状态：第一个词元、中间词元和最后一个词元。对于每个位置，我们将最终层所有注意力头上的键向量和值向量拼接起来，得到一个固定维度表示。我们使用这些冻结表示训练逻辑回归分类器，并通过在以下范围内进行交叉验证来选择 L2 正则化系数 $[10^{-4}, 10^{1}]$。

结果 表 6 展示了各个位置的探测准确率。在两个数据集上，最后一个词元始终取得最高准确率。在 IMDb 上，准确率从第一个位置的 51.4% 提升至最后一个位置的 85.2%，绝对增幅超过 33 个百分点。推文情感分类也呈现出类似趋势，最后一个位置的表现比第一个位置高出 15.6 个百分点。这些结果证实，在因果注意力机制下，序列级信息会逐步向序列末尾累积，为我们的键值重路由方法提供了直接的实证支持。

C 逐层表示分析

为了了解表征质量如何随层变化，并验证我们的层选择策略，我们使用 Mistral-7B-Instruct（Jiang 等人，2023年）在语义相似度任务上评估每一层最后一个词元的隐藏状态。

位置	IMDB	推文情感
第一个	51.4	45.2
中间	73.0	52.6
Last	85.2	60.8

表 6：使用 Mistral-7B-Instruct 不同词元位置的键值状态，在分类任务上的探测准确率（%）。

设置 我们提取全部 32 层的隐藏状态，并在两项任务上进行评估：成对分类，通过余弦 AP 衡量细粒度语义匹配；语义文本相似度（STS），通过斯皮尔曼相关系数衡量相似性。这些任务揭示了表征质量的互补方面。

结果表7揭示了两项任务的不同模式。成对分类在第 10–13 层达到峰值，之后有所下降，这表明中间层捕获了用于细粒度语义匹配的最具判别性的特征。相比之下，语义文本相似度在较后层表现更佳，这可能是因为下一个词元预测偏差更强调局部词语关联，而非整体语义。

层范围	成对分类	STS
0–6	0.270	0.221
7–12	0.302	0.282
13–19（按 ID 选择）	0.301	0.290
20–26	0.286	0.297
27–32	0.275	0.339

表 7：Mistral-7B-Instruct 的逐层探测性能。基于 ID 选定的范围在避免后期层偏向生成的同时，保持了较强的平均性能。

我们的基于 ID 的策略选取第 13–19 层，对应语义抽象程度较高、但预测偏差尚未占据主导的过渡区域。该范围避开了语义内容不足的早期层，以及越来越针对下一词元预测进行优化的最终层。成对分类结果支持这一选择，证实基于 ID 选定的范围保留了较强的语义匹配能力。

D 基准测试详情

D.1 MTEB 基准测试

大规模文本嵌入基准测试（MTEB）（Muennighoff 等人，2023年）为评估文本提供了标准化框架

在多样化语义场景中的嵌入。我们开展了全面评估，涵盖七个核心任务类别和 42 个数据集，以全面评估语义表示。

语义文本相似度（STS）结果基于完整的标准语义文本相似度（STS）任务集报告，其中包括 BIOSSES、SICK-R 和 STS12 至 STS16。按照标准做法，我们使用嵌入对之间余弦相似度与人工标注的金标准标签之间的斯皮尔曼相关系数，作为衡量嵌入空间中细粒度对齐情况的核心指标。

重排序和检索重排序性能使用 AskUbuntu 重复问题、MindSmallReranking、SciDocs 重排序 和 StackOverflow 重复问题上的均值平均精度（MAP）进行衡量。对于检索，我们采用一组广泛的基准测试，包括 NFCorpus、FiQA2018、SciFact 和 QuoraRetrieval，并以 NDCG@10 作为评估排序效果的主要指标。

分类和摘要对于分类任务，我们评估嵌入在 ToxicConversations、IMDb、Banking77、Emotion 和 TweetSentimentExtraction 上的判别能力，并报告各项任务的准确率。摘要能力通过 Summeval 数据集进行衡量，我们根据余弦相似度计算斯皮尔曼相关系数，以评估生成的摘要与源文档之间的对齐情况。

聚类和成对分类聚类评估涵盖了来自不同领域的 11 个数据集，包括 Biorxiv 、 Medrxiv 、 Reddit 和 StackExchange，并使用 V-measure 衡量性能。对于成对分类任务，例如 TwitterURLCorpus 和 SprintDuplicateQuestions，我们报告平均精度（AP），以衡量模型对二元语义相似度的敏感度。

D.2 LoCoV1 基准

LoCoV1（Saad-Falcon 等人，2024年）是针对长上下文稳定性的专项压力测试。与 MTEB 中以短文本为主的内容不同，LoCoV1 要求在较长序列中保持语义密度。为分析序列长度对我们的 KV 重新路由机制的影响，我们在 1k、2k 和 4k 词元的受控截断条件下评估性能。对于检索任务，我们采用 NDCG@10，并将结果与不同的基线方法进行比较。

E MTEB 详细结果

本节展示 MTEB 基准测试中各任务类别的详细结果（Muen- nighoff 等人，2023年）。我们报告了三种模型的性能：Qwen3-4B、Mistral-7B-Instruct-v0.1 和 Llama-3.1-8B-Instruct。我们比较了六种不同的嵌入方法：最后一个词元、均值池化、PromptEOL（Jiang 等人，2024年）、Echo（Springer 等人，2025年）、TokenPrepending（Fu 等人，2025年）以及我们提出的 KV 嵌入。

结果的详细分类如下。表8报告了语义文本相似度（STS）任务的 Spearman 相关系数，而表9 则详述了以 NDCG@10 衡量的检索性能。各评估数据集上的分类准确率见表10。对于成对分类，表11给出了平均精度（AP）分数。此外，表12展示了以有效性度量（V-measure）衡量的聚类结果，表13展示了重排序性能（平均准确率均值），最后，表14展示了摘要任务的 Spearman 相关系数。

F LoCoV1 详细结果

LoCoV1 实验的完整结果见表15、16和17。在测试的三种上下文窗口和各类模型架构中，我们的 KV 嵌入方法始终取得最高的平均性能，展现出在长上下文检索任务中更强的鲁棒性。

G 注意力偏置实验

在注意力计算过程中，当关注位置 0 处重新路由的 KV 对时，我们会向预 softmax logit 添加标量偏置b 。对于任意位置 i处的查询 $\mathbf{q}_i$，修改后的注意力分数为：

$$\tilde{s}_{i,0} = \frac{\mathbf{q}_i^\top \mathbf{k}_0}{\sqrt{d}} + b \tag{5}$$

最终的注意力权重 $\alpha_{i,j}$ 计算如下：

$$\alpha_{i,j} = \frac{\exp(\tilde{s}_{i,j})}{\sum_{k=0}^{n} \exp(\tilde{s}_{i,k})} \tag{6}$$

其中 $\tilde{s}_{i,k} = s_{i,k}$ 对于所有 $k > 0$均成立。更高的 b 会促使模型更加关注全局摘要。

模型	方法	Avg.	BIOSSES	SICK-R	STS12	STS13	STS14	STS15	STS16	STS-B
Qwen3-4B	最后一个词元	0.2818	0.2781	0.4939	0.1761	0.2796	0.1560	0.1161	0.4477	0.3068
	均值池化	0.4571	0.5943	0.5148	0.4526	0.3556	0.3286	0.5481	0.5249	0.3378
	PromptEOL	0.6741	0.5997	0.6612	0.5428	0.7460	0.6356	0.7359	0.7541	0.7176
	Echo	0.6634	0.6726	0.6565	0.5796	0.6884	0.5972	0.7275	0.7246	0.6607
	令牌前置	0.6370	0.5648	0.6405	0.5154	0.7040	0.6061	0.6947	0.6924	0.6785
	KV 嵌入	**0.7141**	0.7551	0.6835	0.6247	0.7740	0.6665	0.7622	0.7586	0.6887
Mistral-7B-Instruct-v0.1	最后一个词元	0.3915	0.3089	0.5465	0.2552	0.3657	0.2411	0.2488	0.6033	0.5629
	均值池化	0.4885	0.6425	0.5036	0.3910	0.4536	0.3781	0.5854	0.5695	0.3843
	PromptEOL	0.6953	0.5817	0.6918	0.6554	0.7783	0.6740	0.7575	0.6981	0.7258
	Echo	0.7333	0.7716	0.7245	0.6041	0.7916	0.6822	0.7899	0.7581	0.7443
	令牌前置	0.6775	0.5716	0.6825	0.6210	0.7587	0.6633	0.7343	0.6741	0.7145
	KV 嵌入	**0.7720**	0.7496	0.7683	0.6891	0.8056	0.7308	0.8254	0.7975	0.8097
Llama-3.1-8B-Instruct	最后一个词元	0.3513	0.3302	0.5315	0.1837	0.3846	0.2226	0.2052	0.5462	0.4063
	均值池化	0.4702	0.6253	0.5103	0.3583	0.4200	0.3602	0.5650	0.5403	0.3824
	PromptEOL	0.6919	0.6509	0.7034	0.5765	0.7376	0.6306	0.7383	0.7636	0.7343
	Echo	0.7124	0.7704	0.6956	0.5428	0.7843	0.6514	0.7557	0.7676	0.7311
	令牌前置	0.6968	0.6536	0.6959	0.5947	0.7642	0.6665	0.7416	0.7342	0.7240
	KV 嵌入	**0.7398**	0.7710	0.7254	0.6239	0.7793	0.6946	0.7819	0.7874	0.7546

表 8：M　不同模型和方法下的 TEB STS 斯皮尔曼相关系数结果。最高平均得分　　re
对于每个模 型均以粗体突出显示。

模型	方法	Avg.	ArguAna	NFCorpus	FiQA	SciFact	SCIDOCS	Quora	TREC	Touche	CQADupstack
Qwen3-4B	最后一个词元	0.0722	0.0797	0.0175	0.0000	0.0087	0.0031	0.5140	0.0159	0.0000	0.0111
	均值池化	0.0294	0.1674	0.0142	0.0047	0.0362	0.0034	0.0170	0.0193	0.0000	0.0021
	PromptEOL	0.1857	0.2462	0.0948	0.0519	0.2138	0.0850	0.6500	0.2393	0.0135	0.0767
	Echo	0.1727	0.2336	0.0347	0.0552	0.1146	0.0211	0.7465	0.2239	0.0327	0.0918
	令牌前置	0.1622	0.1912	0.1060	0.0496	0.1294	0.0713	0.5622	0.1949	0.0066	0.1489
	KV 嵌入	**0.2765**	0.4643	0.1069	0.1023	0.4054	0.0608	0.6916	0.3957	0.0869	0.1749
Mistral-7B-Instruct-v0.1	最后一个词元	0.1036	0.0961	0.0347	0.0141	0.0220	0.0027	0.6655	0.0770	0.0022	0.0178
	均值池化	0.0886	0.3972	0.0201	0.0095	0.2365	0.0044	0.0525	0.0669	0.0000	0.0100
	PromptEOL	0.1746	0.1030	0.1404	0.0781	0.2033	0.0934	0.5847	0.2212	0.0359	0.1118
	Echo	0.2414	0.2719	0.1199	0.0909	0.3325	0.0327	0.7676	0.3283	0.0487	0.1797
	令牌前置	0.1619	0.0589	0.1453	0.0768	0.1838	0.0834	0.5520	0.2143	0.0173	0.1257
	KV 嵌入	**0.3014**	0.2543	0.2257	0.2063	0.3723	0.0969	0.7774	0.4387	0.1120	0.2292
Llama-3.1-8B-Instruct	最后一个词元	0.0887	0.1104	0.0210	0.0088	0.0136	0.0037	0.6128	0.0232	0.0007	0.0041
	均值池化	0.0713	0.4113	0.0164	0.0023	0.1201	0.0030	0.0275	0.0593	0.0000	0.0017
	PromptEOL	0.2017	0.2983	0.1317	0.0747	0.2140	0.0740	0.6890	0.2316	0.0066	0.0957
	Echo	0.2941	0.3685	0.1183	0.1255	0.4756	0.0679	0.8026	0.4382	0.0581	0.1920
	令牌前置	0.1817	0.0721	0.1893	0.0833	0.1678	0.1222	0.6488	0.1952	0.0131	0.1434
	KV 嵌入	**0.3079**	0.4965	0.2046	0.1507	0.4452	0.0724	0.7794	0.3993	0.0602	0.1629

表 9：MTEB检索（NDCG@10）结果。每个模型的最高平均分数以粗体突出显示。缩写：*TREC* （TREC-COVID）、*CQADupstack* （CQADupstackEnglishRetrieval）。

模型	方法	Avg.	有毒	Imdb	Banking77	Emotion	推文情感
Qwen3-4B	最后一个词元	0.4243	0.6463	0.5616	0.1850	0.2636	0.4648
	均值池化	0.4656	0.5867	0.6914	0.3730	0.2244	0.4527
	PromptEOL	0.6138	0.6976	0.7377	0.5057	0.5043	0.6236
	Echo	0.5761	0.6740	0.6315	0.6426	0.4049	0.5278
	令牌前置	0.6283	0.6837	0.8281	0.4863	0.5079	0.6357
	KV 嵌入	**0.6375**	0.7088	0.7259	0.6746	0.4859	0.5922
Mistral-7B-Instruct-v0.1	最后一个词元	0.5314	0.6375	0.6324	0.5491	0.2958	0.5424
	均值池化	0.5157	0.6180	0.7334	0.4702	0.2553	0.5019
	PromptEOL	0.6571	0.7202	0.8641	0.5416	0.5042	0.6554
	Echo	0.6398	0.6868	0.7290	0.7270	0.4452	0.6109
	令牌前置	0.6580	0.7094	0.8654	0.5572	0.5034	0.6546
	KV 嵌入	**0.6951**	0.7101	0.8655	0.7204	0.5193	0.6602
Llama-3.1-8B-Instruct	最后一个词元	0.4769	0.6170	0.6163	0.3641	0.2753	0.5120
	均值池化	0.4848	0.5940	0.7286	0.3796	0.2414	0.4801
	PromptEOL	0.6250	0.6880	0.8283	0.5530	0.4381	0.6179
	Echo	0.6261	0.6814	0.7328	0.6958	0.4418	0.5788
	令牌前置	0.6577	0.7065	0.8926	0.5718	0.4796	0.6379
	KV 嵌入	**0.6602**	0.6931	0.8660	0.6876	0.4605	0.5939

表 10：M TEB 分类（准确率）结果。每个模型的最高平均分数以粗体突出显示。 ld.
列标 题缩写如下：有毒 （ToxicConversations）、推文情感 （TweetSentimentExtraction）。

模型	方法	Avg.	TwitterURL	SprintDup	TwitSemEval
Qwen3-4B	最后一个词元	0.3054	0.4003	0.1278	0.3882
	均值池化	0.5269	0.6720	0.5726	0.3361
	PromptEOL	0.5637	0.7592	0.3597	0.5722
	Echo	0.6367	0.6791	0.7071	0.5239
	令牌前置	0.5206	0.7637	0.2008	0.5972
	KV 嵌入	**0.6800**	0.7898	0.7144	0.5357
Mistral-7B-Instruct-v0.1	最后一个词元	0.3440	0.4417	0.1441	0.4462
	均值池化	0.5681	0.7388	0.5881	0.3774
	PromptEOL	0.5629	0.8079	0.1749	0.7059
	Echo	0.7560	0.8059	0.7916	0.6707
	令牌前置	0.5764	0.8113	0.1974	0.7206
	KV 嵌入	**0.7564**	0.8460	0.6939	0.7291
Llama-3.1-8B-Instruct	最后一个词元	0.3554	0.4745	0.1289	0.4630
	均值池化	0.5176	0.7157	0.4670	0.3701
	PromptEOL	0.5911	0.7624	0.3968	0.6140
	Echo	0.7125	0.7749	0.7314	0.6312
	令牌前置	0.5734	0.7842	0.2385	0.6975
	KV 嵌入	**0.7332**	0.8161	0.7237	0.6598

表 11：MTEB 配对分类（平均精度）结果。每个模型的最高平均分数以粗体突出显示。列标题采用缩写：*SprintDup* （SprintDuplicateQuestions）、*TwitSemEval*（TwitterSemEval2015）。

模型	方法	Avg.	BioP2P 生物学序列到序列	医学点对点	医学序列到序列	Reddit	Reddit 点对点	Stack Exchange	Stack 点对点	20 新闻组
Qwen3-4B	最后一个词元	0.2395	0.1783	0.1737	0.2059	0.2238	0.1610	0.3362	0.4269	0.3053	0.1441
	均值池化	0.3305	0.3808	0.2528	0.3423	0.2806	0.1984	0.5372	0.4074	0.4088	0.1661
	PromptEOL	0.3651	0.2855	0.3095	0.2699	0.2989	0.3692	0.4943	0.5488	0.3500	0.3597
	Echo	0.3530	0.3290	0.2688	0.3085	0.2909	0.3228	0.4843	0.5631	0.3786	0.2306
	令牌前置	0.3372	0.2851	0.3020	0.2681	0.3009	0.2799	0.4776	0.4002	0.3449	0.3757
	KV 嵌入	**0.3903**	0.3627	0.3165	0.2967	0.3181	0.3721	0.5526	0.5409	0.3890	0.3642
Mistral-7B-Instruct-v0.1	最后一个词元	0.2516	0.1043	0.1821	0.1824	0.2312	0.2666	0.3505	0.4848	0.2860	0.1766
	均值池化	0.3472	0.3981	0.2696	0.3523	0.2889	0.2258	0.5847	0.4004	0.4078	0.1969
	PromptEOL	0.3158	0.2289	0.2779	0.2253	0.2671	0.3014	0.4622	0.4519	0.3301	0.2978
	Echo	0.3681	0.3407	0.2723	0.3183	0.2804	0.3496	0.5288	0.5566	0.3724	0.2942
	令牌前置	0.2825	0.2124	0.2757	0.2156	0.2580	0.2173	0.4431	0.3192	0.3169	0.2841
	KV 嵌入	**0.3902**	0.3196	0.3308	0.2795	0.3107	0.3901	0.5538	0.5460	0.3685	0.4129
Llama-3.1-8B-Instruct	最后一个词元	0.2652	0.2125	0.1686	0.2169	0.2275	0.2741	0.3376	0.4912	0.2957	0.1630
	均值池化	0.3577	0.4047	0.2820	0.3516	0.2860	0.2463	0.5870	0.4322	0.4237	0.2060
	PromptEOL	0.4035	0.3357	0.3406	0.2916	0.3291	0.4301	0.5354	0.5873	0.3543	0.4269
	Echo	0.4189	0.3574	0.3071	0.3303	0.3122	0.4419	0.5893	0.6279	0.4153	0.3891
	令牌前置	0.3365	0.3037	0.3165	0.2672	0.3019	0.2847	0.4788	0.3781	0.3413	0.3559
	KV 嵌入	**0.4448**	0.3770	0.3654	0.3252	0.3453	0.4908	0.6061	0.6103	0.3992	0.4838

表 12：MTEB 聚类（V-measure）结果。每个模型的最高平均分数以粗体标出。列标题经过缩写以适应页面（例如，*BioP2P* 表示 BiorxivClusteringP2P，*20* 新闻组 表示 TwentyNews- groupsClustering）。

模型	方法	Avg.	AskUbuntu	MindSmall	SciDocs	StackOverflow
Qwen3-4B	最后一个词元	0.3476	0.4255	0.2445	0.4790	0.2415
	均值池化	0.3272	0.4378	0.2823	0.3543	0.2342
	PromptEOL	0.4924	0.5360	0.2947	0.7558	0.3833
	Echo	0.4315	0.5065	0.2791	0.5523	0.3880
	令牌前置	0.4901	0.5263	0.2930	0.7573	0.3838
	KV 嵌入	**0.5007**	0.5672	0.3066	0.7203	0.4088
Mistral-7B-Instruct-v0.1	最后一个词元	0.3673	0.4432	0.2532	0.4964	0.2763
	均值池化	0.3790	0.4760	0.2715	0.4777	0.2906
	PromptEOL	0.4835	0.5402	0.2905	0.7276	0.3759
	Echo	0.4751	0.5390	0.2863	0.6402	0.4347
	令牌前置	0.4790	0.5403	0.2776	0.7120	0.3863
	KV 嵌入	**0.5145**	0.5723	0.3034	0.7533	0.4289
Llama-3.1-8B-Instruct	最后一个词元	0.3608	0.4447	0.2458	0.4866	0.2661
	均值池化	0.3537	0.4510	0.2657	0.4254	0.2729
	PromptEOL	0.4991	0.5360	0.3095	0.7606	0.3902
	Echo	0.5074	0.5338	0.3147	0.7500	0.4313
	令牌前置	0.4992	0.5533	0.2942	0.7458	0.4034
	KV 嵌入	**0.5237**	0.5612	0.3097	0.7841	0.4396

表 13：MTEB 重排序（平均准确率均值）结果。每个模型的最高平均分数以粗体标出。列标题为缩写：*AskUbuntu*（AskUbuntu 重复问题）、*StackOverflow*（StackOverflow 重复问题）、*SciDocs*（SciDocs 重排序）。

模型	方法	SummEval
Mistral-7B-Instruct-v0.1	最后一个词元	0.2823
	均值池化	0.2557
	PromptEOL	0.2863
	Echo	0.2922
	令牌前置	0.2921
	KV 嵌入	**0.3088**
Qwen3-4B	最后一个词元	0.2341
	均值池化	0.1949
	PromptEOL	0.2396
	Echo	0.2294
	令牌前置	0.2483
	KV 嵌入	**0.2566**
Llama-3.1-8B-Instruct	最后一个词元	0.2526
	均值池化	0.1996
	PromptEOL	0.2788
	Echo	0.2527
	令牌前置	**0.2880**
	KV 嵌入	0.2796

表 14：MTEB 摘要任务（斯皮尔曼相关系数）结果。每个模型的最高分以粗体标出。

表 18 展示了改变 b 对 MTEB 性能的影响。从 $b = 0$ 到$b = 1.0$ 的提升证实，重新路由的 KV 对为嵌入质量提供了有用的上下文。性能曲线在较宽的范围内保持稳定，波动幅度在 ±1.5%（0.519 至 0.534）以内，这表明注入的 KV 对能够自然地融入预训练注意力机制。然而，当 b 超过 3.0 时，性能会下降，这说明过度偏置可能导致模型过度依赖全局摘要，而牺牲局部细节。我们将 $b = 1.0$ 设为在全局上下文访问与局部特异性之间取得平衡。

H 嵌入空间的几何结构

为评估表示空间的结构属性，我们采用 Wang 和 Isola（2020年）提出的对齐度和均匀度指标。我们使用 Mistral-7B-Instruct，在 F2LLM 数据集（Zhang 等人，2025年）的 1,000 个句子对的子集上计算这些指标。令 $f(x) \in \mathbb{R}^d$ 为句子 ℓ_2 的归一化嵌入 x。

对齐 对齐用于评估语义相似样本对的嵌入接近程度。对齐分数越低，表示模型越能有效地将相关实例映射到潜在空间中彼此接近的区域，从而保持局部语义一致性：

$$\mathcal{L}_{\text{align}}(f;\alpha) \triangleq \mathbb{E}_{(x,y)\sim p_{\text{pos}}}\left[\|f(x)-f(y)\|_2^{\alpha}\right] \quad (7)$$

按照标准做法，我们在实验中设置了 $\alpha = 2$。

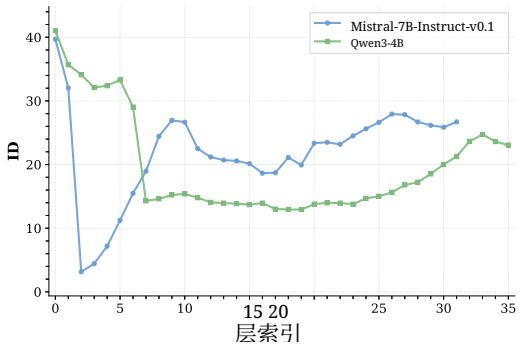

图 3：Mistral-7B-Instruct-v0.1 和 Qwen3-4B 各层的内在维度（ID）。不同架构呈现出不同的语义压缩模式。

均匀性 均匀性用于评估嵌入在单位超球面上的分布是否均匀。均匀性高（得分更负）意味着嵌入能够捕获最多的信息，并避免各向异性问题，即向量坍缩到一个狭窄的锥体中：

$$\mathcal{L}_{\text{uniform}}(f;t) \triangleq \log \mathbb{E}_{x,y \overset{i.i.d.}{\sim} p_{\text{data}}}\left[e^{-t\|f(x)-f(y)\|_2^2}\right] \quad (8)$$

其中，$t = 2$ 用于评估。这些指标共同全面展现了表示的质量，兼顾局部聚类与全局离散。

I 内在维度轨迹

图 3 展示了 Mistral-7B 和 Qwen3-4B 所有层的内在维度（ID）轨迹。我们使用 TwoNN 估计器（Facco 等人，2017年）对 F2LLM 数据集中的 1,000 个样本进行估计。

可视化结果揭示了这两种架构截然不同的几何模式。Mistral-7B 的内在维度先急剧下降，随后出现波动，并在第 13–19 层附近形成稳定的低内在维度区域。Qwen3-4B 的下降则更为平缓，全局最小值出现在第 17–19 层附近。相比之下，Qwen3-4B 的下降同样更为平缓，其全局内在维度最小值出现在中间层。两种模型都呈现出相同趋势：随着表示接近最终层，内在维度再次上升。

这些轨迹的差异表明，不同模型的最大压缩层并不相同；在这些层中，表示流形所占据的内在维度数量最少。对于 Mistral-

模型	方法	2wiki mqa	法院 HTML	法院 Text	gov rep	法律 case	多 qa	pass ret	qasper abs	qasper 标题	QMSum	堆栈 over	summ scrn	Avg.
Mistral-7B	PromptEOL	0.0757	0.0020	0.0010	0.0059	0.0060	0.1515	0.0757	0.0047	0.0109	0.1492	0.0009	0.0134	0.0414
	Echo	0.1699	0.0034	0.0055	0.0047	0.0089	0.1515	0.1461	0.0703	0.0649	0.1033	0.0085	0.0151	0.0627
	令牌前置	0.0757	0.0023	0.0023	0.0047	0.0059	0.1515	0.0757	0.0109	0.0109	0.1492	0.0028	0.0077	0.0416
	KV 嵌入	0.1983	0.0077	0.0090	0.2023	0.0163	0.6080	0.1656	0.1654	0.0338	0.3936	0.6191	0.1791	0.2165
Qwen3-4B	PromptEOL	0.0757	0.0045	0.0025	0.0000	0.0206	0.1515	0.0757	0.0109	0.0109	0.1289	0.0000	0.0144	0.0413
	Echo	0.3677	0.0106	0.0078	0.0047	0.0097	0.1515	0.0678	0.1034	0.2250	0.1673	0.0248	0.0134	0.0961
	令牌前置	0.0674	0.0023	0.0023	0.0047	0.0059	0.1515	0.0757	0.0109	0.0109	0.1492	0.0029	0.0134	0.0414
	KV 嵌入	0.1155	0.0052	0.0087	0.0946	0.0125	0.3301	0.1221	0.0509	0.0187	0.2055	0.5391	0.0434	0.1289
Llama-3.1-8B	PromptEOL	0.0757	0.0045	0.0045	0.0044	0.0000	0.1515	0.0757	0.0099	0.0218	0.2012	0.0052	0.0009	0.0463
	Echo	0.1035	0.0186	0.0245	0.2445	0.0239	0.1515	0.0757	0.2633	0.3693	0.0436	0.0044	0.0304	0.1128
	令牌前置	0.0757	0.0023	0.0023	0.0047	0.0059	0.1515	0.0757	0.0109	0.0109	0.1492	0.0029	0.0134	0.0421
	KV 嵌入	0.2357	0.0100	0.0184	0.3367	0.0206	0.4842	0.1985	0.1755	0.0436	0.2658	0.6868	0.1535	0.2191

表15：LoCoV1实验结果：1024 个令牌长度

模型	方法	2wiki mqa	法院 HTML	法院 Text	gov rep	法律 case	多 qa	pass ret	qasper abs	qasper 标题	QMSum	堆栈 over	summ scrn	Avg.
Mistral-7B	PromptEOL	0.1308	0.0023	0.0023	0.0047	0.0106	0.1515	0.0599	0.0148	0.0000	0.0955	0.0032	0.0134	0.0407
	Echo	0.1175	0.0051	0.0023	0.0047	0.0069	0.1973	0.0757	0.0109	0.0109	0.1181	0.0170	0.0160	0.0549
	令牌前置	0.0757	0.0028	0.0023	0.0047	0.0059	0.1515	0.0757	0.0109	0.0109	0.1492	0.0029	0.0134	0.0422
	KV 嵌入	0.2246	0.0181	0.0197	0.1203	0.0266	0.4857	0.1307	0.0433	0.0415	0.3183	0.6072	0.1698	0.1838
Qwen3-4B	PromptEOL	0.0757	0.0015	0.0020	0.0073	0.0183	0.1515	0.0757	0.0070	0.0176	0.1492	0.0058	0.0134	0.0438
	Echo	0.3004	0.0055	0.0038	0.0047	0.0059	0.1515	0.0572	0.0109	0.0617	0.1033	0.0289	0.0556	0.0658
	令牌前置	0.0757	0.0028	0.0023	0.0047	0.0059	0.1515	0.0757	0.0109	0.0109	0.1492	0.0029	0.0134	0.0422
	KV 嵌入	0.1387	0.0123	0.0166	0.0299	0.0319	0.2505	0.1203	0.0505	0.0419	0.1741	0.5090	0.0542	0.1192
Llama-3.1-8B	PromptEOL	0.0551	0.0006	0.0023	0.0047	0.0059	0.1515	0.0757	0.0109	0.0218	0.1492	0.0005	0.0143	0.0410
	Echo	0.0451	0.0012	0.0014	0.0058	0.0062	0.1515	0.0789	0.0171	0.0134	0.1035	0.0238	0.0141	0.0385
	令牌前置	0.0757	0.0028	0.0023	0.0047	0.0059	0.1515	0.0757	0.0109	0.0109	0.1492	0.0029	0.0134	0.0422
	KV 嵌入	0.1591	0.0248	0.0355	0.1384	0.0552	0.4116	0.1237	0.0533	0.0433	0.3135	0.6733	0.1482	0.1817

表 16：LoCoV1实验结果：2048 令牌长度

模型	方法	2wiki mqa	法院 HTML	法院 Text	gov rep	法律 case	多 qa	pass ret	qasper abs	qasper 标题	QMSum	堆栈 over	summ scrn	Avg.
Mistral-7B	PromptEOL	0.0279	0.0023	0.0023	0.0051	0.0059	0.1816	0.1036	0.0133	0.0109	0.1649	0.0005	0.0123	0.0442
	Echo	0.1432	0.0023	0.0147	0.0063	0.0149	0.2289	0.0920	0.0055	0.0109	0.1534	0.0228	0.0147	0.0591
	令牌前置	0.1097	0.0092	0.0133	0.0462	0.0709	0.2463	0.0771	0.1725	0.1611	0.1342	0.1327	0.0171	0.0992
	KV 嵌入	0.1626	0.0310	0.0416	0.0789	0.0794	0.3861	0.1058	0.3443	0.3061	0.2569	0.5945	0.0947	0.2068
通义千问 3-4B	PromptEOL	0.0868	0.0137	0.0201	0.0689	0.0390	0.2347	0.0816	0.2283	0.2197	0.1202	0.3995	0.0307	0.1286
	Echo	0.1547	0.0096	0.0106	0.1195	0.0135	0.4423	0.1792	0.1291	0.0618	0.2233	0.1812	0.0564	0.1318
	令牌前置	0.0790	0.0094	0.0212	0.0666	0.0323	0.2608	0.0968	0.2468	0.1953	0.2239	0.2436	0.0297	0.1255
	KV 嵌入	0.1118	0.0259	0.0369	0.0983	0.0964	0.3070	0.0952	0.4180	0.2973	0.1519	0.5003	0.0476	0.1822
Llama-3.1-8B	PromptEOL	0.1032	0.0121	0.0082	0.0852	0.0838	0.2628	0.0913	0.1678	0.1277	0.1383	0.4313	0.0306	0.1285
	Echo	0.1713	0.0036	0.0023	0.0047	0.0051	0.1515	0.0872	0.0081	0.0078	0.1242	0.4377	0.4263	0.1191
	令牌前置	0.1233	0.0090	0.0084	0.0958	0.0579	0.2952	0.0989	0.2379	0.2285	0.1713	0.3247	0.0251	0.1397
	KV 嵌入	0.1685	0.0503	0.0552	0.1138	0.1360	0.3872	0.1369	0.4459	0.3751	0.2862	0.6730	0.0567	0.2404

表17：LoCoV1实验结果：4096令牌长度

注意力偏置	STS	检索	分类	配对	聚类	重排序	摘要	Avg.
0.00	0.7690	0.2942	0.6948	0.7310	0.3855	0.5128	0.3096	0.5281
1.00	0.7720	0.3014	0.6951	0.7564	0.3902	0.5145	0.3088	0.5341
2.00	0.7736	0.3064	0.6949	0.7342	0.3926	0.5159	0.3107	0.5326
3.00	0.7723	0.3089	0.6938	0.7329	0.3928	0.5135	0.3100	0.5320
4.00	0.7649	0.3066	0.6907	0.7259	0.3928	0.5093	0.3027	0.5276
5.00	0.7483	0.2952	0.6819	0.7078	0.3875	0.5010	0.3110	0.5190

表18：Mistral-7B-Instruct 在不同任务中使用不同注意力偏置值时的性能。

对于 7B，最小 ID 出现在非常早期的位置；而对于 Qwen3-4B，它则位于主干网络的中部。这些实证观察表明，相较于固定的层启发式方法，采用自动选择策略更为合理，因为基于内在维度的方法能够针对给定架构，自适应地识别信息最为浓缩的特定层。

J 层选择分析

为验证内在维度（ID）在层选择中的有效性，我们将所用策略与均匀划分策略进行比较：早期层、中间层和后期层，每种策略均覆盖约三分之一的模型总深度。正如表 19 所总结的，基于 ID 分析得出的重路由层 $\mathcal{L}$ 始终取得最高的平均分数，同时所选层的数量显著更少。

对于通义千问3-4B，通过早期层（0–11）进行重路由的平均分数最低（0.409）。这印证了这样一种观点：浅层侧重于表层词汇和句法特征，而这些特征缺乏实现有效全局信息再分配所需的语义密度。相较之下，基于 ID 的层选择确定了一个特定范围的层（12–21），其平均分数达到 0.494，优于覆盖范围更广的中间三分之一策略（12–23）。

Mistral-7B-Instruct-v0.1 上也呈现出类似的模式。在 Mistral-7B 上，基于内在维度的选择仅使用了 7 层（13–19），却超过了使用 10 层的中间层策略（10–19）。这一效率表明，最小内在维度能够准确定位表示流形实现最大压缩的层范围。通过在这些特定位置注入全局上下文，我们的方法避开了早期层中的噪声低层特征，以及最终层隐藏状态中固有的任务特定预测偏差。这些发现表明，与均匀或启发式选择相比，在抽象程度达到峰值的阶段进行语义重分配最为有效。

K 注意力约束的影响

本节对朴素的掩码移除为何会失败进行了进一步分析，并补充了表4中的结果。

仅解码器大语言模型是在严格的因果约束下进行预训练的，每个词元只能关注前面的位置。当强行施加双向注意力时，冻结模型中的每个词元都会突然接收到来自未来位置的键值状态，而这些状态在预训练期间从未出现过。这些分布外状态会扰乱模型学到的注意力模式，导致模型产生质量下降的表示。

Qwen3-4B 和 Mistral-7B 上一致的性能崩溃（表 4）证实，这并非模型特有的问题，而是双向注意力与因果预训练权重之间存在根本性不兼容。值得注意的是，在检索任务上性能下降最为严重，其中 Bi-Attn 得分降至接近零（0.02–0.03），这表明文档级语义受到了尤其严重的破坏。

KV 嵌入通过在因果框架内运作来避免这一问题。重新路由后的 KV 状态源自最终 token；在前向传播过程中，该 token 已合法地关注了所有前置位置。通过在这些状态前添加它们，而非改变注意力掩码，我们既保留了模型的内部动态，又实现了全局上下文访问。

L 提示设计

表20 评估了五种不同提示变体对 Mistral-7B-Instruct-v0.1 性能的影响。结果表明，模型在不同指令措辞下的表现一致，平均分数仅在 0.529 至 0.540 之间小幅波动。这种稳定性表明，内部重新路由机制不受自然语言指令具体选择的影响。

尽管总体差异很小，但某些模板在特定方面表现更佳。例如，提示 0（“用一个词压缩上下文”）作为我们的默认配置，在语义文本相似度（0.772）、分类（0.695）和摘要（0.309）任务中得分最高。这表明它捕捉全局语义信息的能力更强。另一方面，提示 2（“提取关键概念”）在检索（0.336）和聚类（0.419）任务中略占优势，这说明强调提取的措辞可能更契合稠密向量空间中的特定任务表示。

M 池化策略

表 21 比较了键值重新路由后的池化策略。均值池化的表现始终最差

在所有骨干网络上。这很可能是因为早期词元保留了均值运算无法过滤的词汇噪声，即使上下文经过重新路由，也会导致语义稀释。

混合池化（对最后词元表示和均值表示取平均）取得了最高的平均分数。这种策略结合了序列中的分布式证据，以及由最终状态捕获的全局信息摘要。

模型	层选择	STS	检索	分类	配对	聚类	重排序	摘要	Avg.
Qwen3-4B	早期层（0–11）	0.527	0.253	0.578	0.470	0.362	0.441	0.234	0.409
	中间层（12–23）	0.713	**0.284**	**0.639**	0.677	0.391	0.490	0.255	0.493
	后期层（24–35）	**0.726**	0.272	0.634	0.666	**0.399**	0.477	0.247	0.489
	ID 选择（12–21）	0.714	0.277	0.638	**0.680**	0.390	**0.501**	**0.257**	**0.494**
Mistral-7B-Instruct-v0.1	早期层（0–9）	0.759	0.277	0.691	0.732	0.381	0.505	0.301	0.521
	中间层（10–19）	0.771	0.301	0.695	0.731	0.389	0.514	0.304	0.529
	后期层（20–31）	0.771	0.297	**0.696**	0.735	0.382	0.510	0.300	0.527
	ID 选择（13–19）	**0.772**	**0.301**	0.695	**0.756**	**0.390**	**0.515**	**0.309**	**0.534**

表 19：层选择的消融研究。我们比较了均匀策略（Early、Middle、Late）与基于 ID 的选择（括号中为层范围）。ID 选择在使用更少层的同时，取得了最佳平均性能。

ID	提示模板	STS	检索	分类	配对	聚类	重排序	摘要	Avg.
0	用一个词压缩上下文	**0.772**	0.301	**0.695**	0.756	0.390	0.515	**0.309**	0.534
1	用一个词概括	0.768	0.292	0.683	0.756	0.391	0.518	0.303	0.530
2	提取关键概念	0.754	**0.336**	0.666	**0.775**	**0.419**	**0.533**	0.299	**0.540**
3	用一个词表示	0.766	0.294	0.683	0.755	0.392	0.517	0.299	0.529
4	压缩成一个词	0.767	0.294	0.684	0.766	0.393	0.518	0.296	0.531

表 20：提示模板消融实验（Mistral-7B）。不同提示词下的性能稳定，其中“提取关键概念”在检索和聚类任务上表现突出。

模型	池化策略	STS	检索	分类	配对	聚类	重排序	摘要	Avg.
Qwen3-4B	最后一个词元	**0.718**	0.266	**0.652**	0.638	**0.397**	**0.511**	**0.271**	0.493
	均值池化	0.504	0.177	0.516	0.535	0.351	0.411	0.219	0.388
	混合池化	0.714	**0.277**	0.638	**0.680**	0.390	0.501	0.257	**0.494**
Mistral-7B-指令版-v0.1	最后一个词元	0.767	0.253	0.694	0.683	0.338	0.503	0.292	0.504
	均值池化	0.588	0.226	0.566	0.644	0.379	0.446	0.260	0.444
	混合池化	**0.772**	**0.301**	**0.695**	**0.756**	**0.390**	**0.515**	**0.309**	**0.534**

表 21：消融实验：池化策略。混合池化（最后词元 +均值）取得了整体最佳性能 ce.
粗体：最佳，下划线：次佳。