拼写并不简单：
LLMs 将标记转换为字符的能力

Tatsuya Hiraoka **Kentaro Inui**
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)
RIKEN
{ tatsuya.hiraoka, kentaro.inui } @mbzuai.ac.ae

Abstract

大型语言模型（LLMs）能够以较高的准确率逐字拼写出字符，但在处理更复杂的字符级任务方面存在困难，例如识别标记中的组合子组件。在这项工作中，我们研究了 LLMs 在拼写过程中如何在内部表示和利用字符级信息。我们的分析表明，尽管拼写对于人类来说是一个简单的任务，但 LLMs 并没有以一种简单明了的方式处理。具体来说，我们发现嵌入层并没有完全编码字符级信息，尤其是在第一个字符之后。因此，LLMs 依赖于中间和更高的变压器层来重构字符级知识，在此过程中我们观察到了一个明显的拼写行为“突破”。我们通过三种互补分析验证了这一机制：探测分类器、知识神经元的识别以及注意力权重的检查。

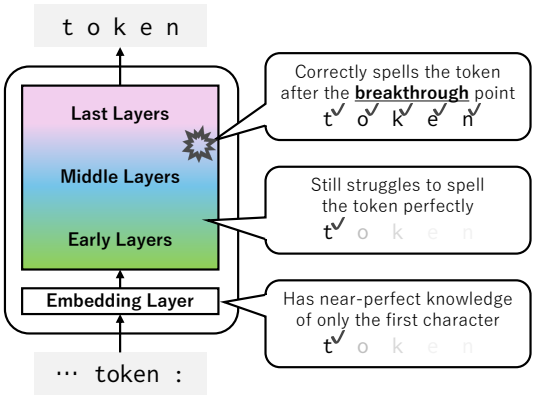

Figure 1: 我们研究结果的总结。大型语言模型（LLMs）通过直接依赖嵌入级信息来拼出第一个字符的符号，但逐渐转向使用分布式的高层表示来拼写后续字符。我们发现了一个“突破”层，在这一层中字符的知识变得可检测。

近年来，大型语言模型（LLMs）增长显著，但有几项研究报告称，它们在细粒度的字符级操作上仍然存在困难，比如插入、删除或提取标记内的单个字符。尽管大多数 LLM 是在子词标记上运行的，但对子标记信息的真正掌握对于一系列应用至关重要，比如形态变化、字母计数、颠倒字母的阅读障碍处理法以及处理拼写错误。为了提高它们在这些场景中的可靠性，我们必须理解 LLM 如何在内部表示和处理字符。

之前的研究中出现了一个悖论：LLMs 可以将整个标记准确地拼写为字符序列（Edman et al., 2024; Xiong et al., 2025），但它们在简单任务上常常失败，例如识别固定位置的单个字符（Hiraoka and Okazaki, 2024; Chai et al., 2024）。例如，对于一个标记“`language`”，模型可以按需生成“`l a n g u a g e`”，但不能可靠地在第五个位置提取“`u`”。这种差异表明，尽管缺乏对组合字符知识的显性访问，LLMs 仍有某种逐个字符拼写的机制。

在本文中，我们研究了在拼写过程中 LLM 如何以及在哪里捕获和使用字符级知识。我们首先从四个代表性 LLM 的词汇中构建一个词元-字符数据集（§2），并确认它们能够通过少量示例提示来拼写词元（§3）。对嵌入层的探查显示，它并没有直接编码子词元字符（§4），但下游的 Transformer 层逐步恢复了这些信息。我们识别出一个明确的“突破”层，在该层字符身份可以通过我们的探查分类器可靠地检测到（§**??**），并观察到第一个字符与后续字符的处理有所不同。最后，通过神经元级别分析（§5）和注意力权重检查（§6），我们追踪了字符知识的存储和分流方式，证明其位置与突破点精确对齐。

综上所述，我们的研究为 LLMs 的内部字符级机制带来了新的见解。我们的主要贡献如下：

- 我们证明了标记嵌入并未完全编码字符级信息，而随后的 Transformer 层在拼写处理过程中发挥了重建该信息的主要作用。
- 我们找到了组合特征信息“突破”的确切层次，并通过探测分类器进行了验证。
- 我们研究字符知识到特定神经元和注意力模式，显示与突破层的一致性。
- 我们提供了一个诊断框架（数据集、探针和分析）供将来对子标记和字符级建模的

	# Vocab	# Dataset	%
LLaMA3-8B	128,256	19,724	15.38
Gemma-7B	256,000	47,833	18.68
Qwen2.5-7B	152,064	18,973	12.48
Amber-6.7B	32,000	6,130	19.16

Table 1: 每个大型语言模型（# Vocab）的词汇表中的词元数量和我们数据集中的词元数量（# Dataset，§2.2 ）。% 显示了词汇表中数据集词元的比例。

研究使用。

1 相关工作

我们的研究与关注大型语言模型（LLM）字符级操控能力的工作一致。Edman et al. (2024) 介绍了一个数据集，以从字符级操控的一些视角评估这种能力，包括拼写出单词或符号。类似地，Wang et al. (2024) 提供了一个复杂字符级操控任务的数据集。这两项工作总结出，LLM 在处理复杂字符方面的能力有限。此外，这些工作报告称，LLM 可以按原始顺序拼写出单词，如 Xiong et al. (2025) 所述，但它们对单词内部的单个字符缺乏足够的知识（Shin and Kaneko, 2024; Hiraoka and Okazaki, 2024; Chai et al., 2024），除非对模型进行微调以直接学习符号的内部信息（Xu et al., 2024）。

这一系列文献促使我们研究 LLMs 在缺乏字符级知识情况下拼写行为的内部运作。最近，我们可以看到一种趋势，即了解 LLMs 识别字符级信息的能力，例如数字符的能力（Fu et al., 2024）。此外，超越从词到字符的拼写，Wu et al. (2025) 研究了它们识别汉字内部部首的能力。

我们的工作也与 LLMs 在其后层中“去标记化”能力的发现有关（Kaplan et al., 2025; Kamoda et al., 2025），即将标记内部合并成单词或短语的能力。相比之下，我们关注的是逆问题：LLMs 如何在内部将标记“标记化”为字符。

2 实验设置

2.1 目标模型

我们研究了四个中等大小的 LLMs（≈ 7 B 参数）：LLaMA3-8B（Dubey et al., 2024）、Gemma-7B（Team et al., 2024）、Qwen2.5-7B（Yang et al., 2024）和 Amber（Liu et al.）。我们选择它们是因为我们的初步试验表明较小的模型（例如，3B）不能以足够的准确性拼出标记。所有模型都采用了基于 Transformer 的架构（Vaswani et al., 2017）。表格 1 显示了每个模型的词汇大小，范围从 32K 到 256K 标记。

Few-shot Example	`hello : h e l l o,`
	`world : w o r l d,`
	`orange : o r a n g e,`
Single Token Input	`libert :`
Expected Output	`l i b e r t`

Table 2: 一个三次示例设置的输入示例。

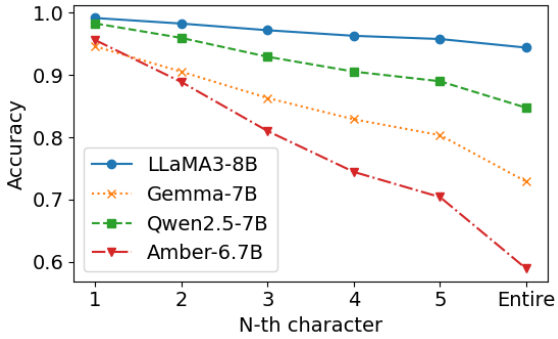

Figure 2: 带有小样本设置的实验结果。$N = 1...5$ 代表第 N 个字符预测的准确率。“Entire”显示了对标记所有字符预测的准确率。

2.2 评估数据集

我们的评估集中于将单个标记拼写成其组成字符。换句话说，我们排除多标记词，以确保良好控制的实验设置。例如，在“`token/s`”的情况下，模型可以轻松揭示最终的“`s`”，而不需要字符级别的理解。

我们通过从每个模型的词汇表中选择所有以下单个标记来构建数据集：1）仅由小写字母组成（`a-z`），2）以指示词首的特殊前缀开头（例如，在“`_token`”中的“`_`”），以及 3）至少有五个字符。这产生了一组大约占每个模型完整词汇表的 15 %（表 1 ）。请注意，某些标记（例如，“somew”、“immedi”）是词的片段而非完整单词。附录中的图 8 和 9 分别展示了标记长度和字母表频率的分布。

鉴于我们数据集中存在的目标标记，预期大型语言模型会输出每个标记的词汇构成字符序列。在我们的实验中，我们通过空格分隔来表示拼写过程。例如，输入“`token`”的模型需要生成“`t o k e n 。`”

3 LLMs 可以拼写出标记

我们首先评估每个大型语言模型在给定少量样本的情况下拼写出单个标记的能力，这遵循以前关于词级拼写的研究（Edman et al., 2024; Xiong et al., 2025）。表 2 显示了这个实验的一个三次样本提示。我们在 §2.2 中创建的完整评估数据集上测量每个大型语言模型的性能。只有当模型的输出与真实的字符序列完全匹配且没有遗漏或多余字符时，我们才认为预测是正

确的。

在图 2 中标记为 “Entire” 的数据点报告了每个 LLM 在整体标记级别的拼写准确率。虽然无法直接在不同模型之间进行比较，因为每个模型使用各自的词汇子集，但我们观察到显著差异：LLaMA3-8B 达到 94.41 % 的准确率，而 Amber-6.7B 仅达到 58.86 %。这种变化显示了模型架构和预训练数据对字符级能力的影响。

在图 2 中，我们也按每个标记中的字符索引报告准确率。所有模型在第一个字符上均达到超过 94 % 的准确率，但在后续位置的准确率稳步下降。这个结果表明，正确生成标记中更远位置的字符变得越来越困难。然而值得注意的是，每个模型在第五个字符时仍能保持超过 70 % 的准确率，这展示了大型语言模型在中间位置标记拼写方面的强大能力。

4 嵌入不识别字符

尽管第 3 节展示了大型语言模型（LLMs）可以拼写出单个 token，但仍有两个关键问题：字符级知识存储在模型的什么位置，以及这种知识在拼写过程中如何被利用？一个自然的假设是，token 嵌入自身编码了这些信息，因为一个 token 的拼写（即其字符序列）本质上是与上下文无关的，而嵌入直接来自于 token 的身份。本节探讨 token 嵌入是否储存了关于拼写 token 的知识。

为了研究 LLMs 在何处存储字符级别的信息，我们训练了探测分类器，该分类器可以根据其嵌入 $\boldsymbol{v}_t \in \mathbb{R}^d$ 来预测标记 t 的第 N 个字符。我们为每个字符位置 N （$1 \leq N \leq 5$）训练了一个单独的 MLP 分类器 [1] ，并在各个位置使用相同的架构。

MLP 分类器将 d 维的标记嵌入 $\boldsymbol{v}_t$ 映射到一个 26 维的 logits 向量，对应于 26 个小写英文字母。N 处的 t 的字符是 c 的概率计算如下：

$$p(t_N = c|t) = \text{softmax}(f(\boldsymbol{v}_t; \theta_N))_c, \quad (1)$$

，其中，$f : \mathbb{R}^d \to \mathbb{R}^{26}$ 是一个具有三层线性层和 tanh 激活的 MLP 分类器，θ_N 是用于预测 N 处字符的可训练参数集。$\text{softmax}(\cdot)_c$ 操作提取分配给字符 c 的概率。

我们在数据集的随机抽样 90 % 上训练每个探测分类器（ğ 2.2 ），并在其余的 10 % 上进行评估。这个过程重复十次，作为 k 折交叉验证。如果分类器能够准确预测给定位置的字符，我们将其解释为该位置的标记嵌入编码了字符级信息的证据。训练使用交叉熵损失和 Adam (Kingma, 2014) 优化器进行，训练的最多轮数为 300。我们基于训练损失使用了早停技术。

[1] 我们选择了非线性探测，因为线性分类器无法从嵌入中提取字符级信息。

4.1 实验结果

图 **??** 显示了探测分类器在各字符位置的性能。对于词汇量较大的大型语言模型（如 LLaMA3-8B，Gemma-7B 和 Qwen2.5-7B），分类器在预测第一个字符时的准确率超过 80 %。然而，随着字符位置的增加，性能持续下降。同时，探测准确率（深色线条）和少样本准确率（浅色线条）之间的差距在后续位置越来越大。这些结果表明，LLM 依赖于标记嵌入来检索第一个字符，但在拼写后续字符时，依靠上层访问字符级信息。

词汇量最小的模型 Amber-6.7B 显示出明显的趋势。即使是对于第一个字符，其探测准确性也显著更低。该结果表明，其标记嵌入几乎或根本不包含字符级信息 [2] 。

总之，这些结果表明，LLM 中的标记嵌入并未完全编码字符组成。虽然有时可以恢复第一个字符，但超出这一的字符级知识主要存储在 LLM 的上层中。

这一部分将探测分析扩展到内部 Transformer 层，以研究字符级知识在模型中的哪个位置出现。

为了分析模型中字符知识的发展，我们应用了 ğ 4 中相同的探测分类器，但用各个 Transformer 层的隐藏状态替代了标记嵌入 $\boldsymbol{v}_t$ 。具体来说，为了预测第 N 个字符，我们从第 l 层提取对应于被预测字符之前的标记的隐藏状态 $\boldsymbol{h}^l_{N-1}$ 。例如，为了预测标记 “`token`” 的第三个字符 ($N = 3$)，我们从输入序列 “`[few-shot examples] token : t o`” 的最后一个标记中提取隐藏状态，期望模型预测 “`k`”。

探测设置的所有其他方面，包括模型架构、训练过程和评估，均与 §4 中描述的一致。

4.2 实验结果

图 3 展示了用于所有四个大型语言模型的探测分类器在 Transformer 层间的字符预测准确性。出现了一种一致的双峰趋势：使用早期层隐藏状态的分类器可以在某种程度上预测字符，但不完美，而使用高层表示的分类器几乎可以在所有位置完美预测字符，中间层经过性能下降。最终层的准确性与图 2 中的结果非常接近，支持了我们评估设置的有效性。

有趣的是，LLaMA3-8B 和 Gemma-7B 在第一个 Transformer 层（即从左边数的第二个数据点）表现出显著的准确率下降。这表明字符级信息在嵌入层之后并未立即可用，甚至在这一阶段可能被扰乱。

[2] 考虑到在 Amber 的中间层上探测性能是合理的（ğ **??** ），较低的嵌入级别准确性不太可能是由数据集规模较小导致的（表 1 ）。

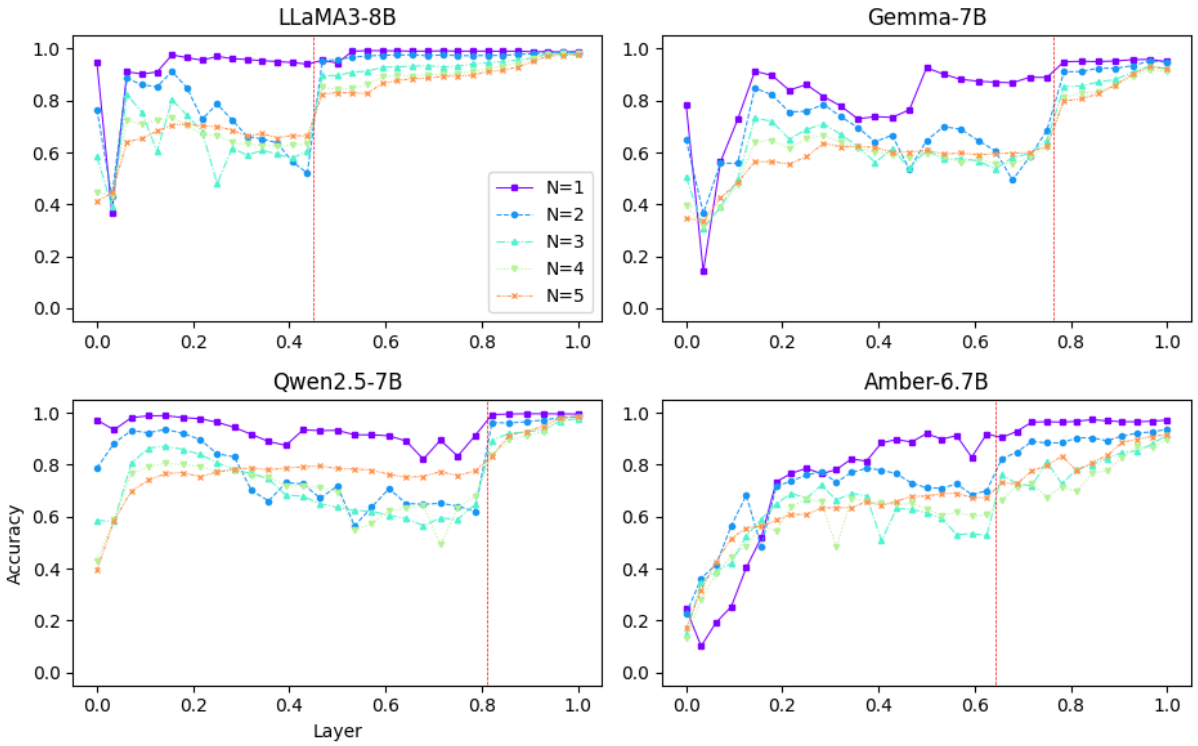

Figure 3: 通过在每个 Transformer 层上探测分类器来预测第 N 个字符的准确度。X 轴表示相对层深度（0.0 = 嵌入层，1.0 = 最终层）。红色垂直线表示突破层，这些层是根据 $2 \leq N$ 相邻层之间的平均性能提升计算得出的。

对于除了 Amber-6.7B 以外的模型，字符预测在早期层中达到更高的准确性，但随后暂时下降，然后在后续层中恢复。这表明，LLM 并不是简单地通过网络传递字符级的信息，而是参与中间处理，将其重新组织为构建拼写任务解决知识的一部分。

所有模型都显示出一个显著的“突破”点，突出显示为红色垂直线，在该点处字符预测的准确性在后面的层中急剧提高。例如，LLaMA3-8B 在层深度约 0.45 处表现出准确性的大幅跃升。Amber-6.7B 在后面的层也显示出上升趋势，尽管与其他模型相比，其改善较为温和。当我们使用明确的分隔符“/”进行拼写时，这一趋势可以更加清晰地看到（图 11）。

这些结果表明，大型语言模型（LLM）开始在网络的后期阶段整合字符级的拼写知识。换句话说，模型首先在其中间层解释拼写的任务，然后在其最终层可能更像字符级语言模型。这种解释与先前的研究结果一致，显示 LLM 的隐藏层状态逐渐转向类似于下一个预测词元的表征（Voita et al., 2024; Chang and Bergen, 2025）。

最后，我们观察到即使在突破点之后，探测准确率仍在继续上升，特别是在最后两个层。这进一步证明了字符级信息不是静态存储在嵌入层中，而是在整个模型中动态构建和优化的，尤其是在前向传递的结束阶段。

5 拼写知识神经元

在 §?? 中观察到的突破点表明，大型语言模型（LLM）具有在此点之前的层中拼写出标记的能力。最近关于可解释性的研究表明，Transformer 模型中的个别神经元可能会编码事实知识或技能，例如知识神经元（Dai et al., 2022）和技能神经元（Wang et al., 2022）。受这类研究启发，我们研究了拼写出标记的知识在每个 LLM 的神经元中哪里存在。

与之前关于 Transformer 架构神经元级分析的研究一致，我们将知识神经元定义为 Transformer 块中前馈网络（FFN）子层中激活函数的输出。为了量化每个神经元对特定输出字符的贡献，我们计算一个归因分数。

例如，给定一个输入 $x =$ “token : t o”，我们的目标是衡量一个神经元 w 在预测下一个正确字符 $c =$ “k”中有多少贡献。我们使用以下基于集成梯度的近似来定义神经元 w 的归因得分 $\mathrm{Attr}(w|c,x)$：

$$\mathrm{Attr}(w|c,x) = \frac{\bar{w}}{m} \sum_{k=1}^{m} \frac{\partial P_{c,x}(\frac{k}{m}\bar{w})}{\partial w}, \qquad (2)$$

，其中 $\bar{w}$ 是神经元 w 激活值，当 LLM 处理输

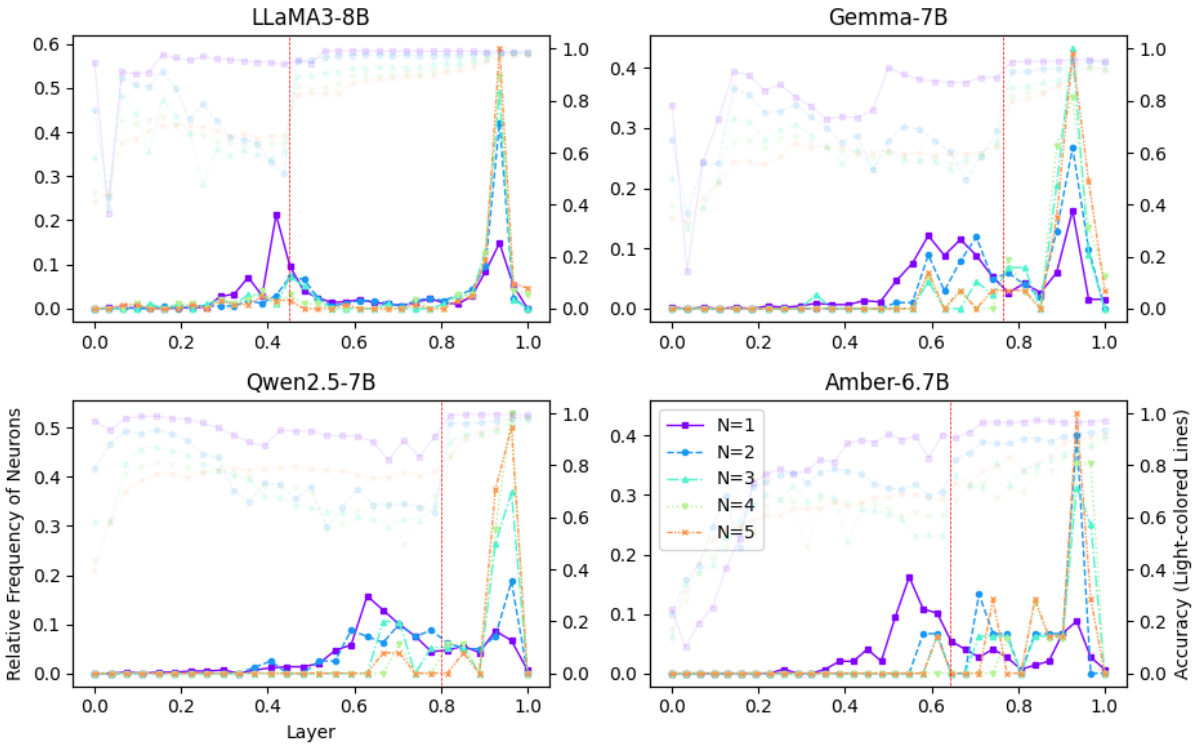

Figure 4: 每一层的知识神经元分布（深色线条）。探测分类器的准确性（浅色线条）从图 3 中复制。

入 x 时。这里，$P_{c,x}(\bar{w})$ 表示给定输入 x 时模型预测的字符 c 的概率，条件是 w 是 $\bar{w}$，而 $m = 20$ 是用于 Riemann 和近似的插值步骤数，如同 Dai et al. (2022)。

利用这一归因分数，我们识别出负责在拼写任务中生成第 N 个字符的知识神经元。换句话说，我们研究神经元激活如何根据预测字符的位置而变化。

为了识别每个字符位置 N 的知识神经元，我们进行如下操作。对于从数据集中抽样的 1000 个标记中的每一个，我们计算所有神经元的 $\mathrm{Attr}(w|c,x)$ 并对它们进行排序。然后，针对每个标记，我们选择在位置 N 具有最高归因分数的前 1 % 个神经元。最后，如果一个神经元在至少 75 % 个 1000 个标记的前 1 % 集中出现，我们将其定义为位置 N 的知识神经元。

5.1 N 的神经元分布

图 4 显示了知识神经元在各层中的分布，用于预测 N -个字符。分布通常表现出两个峰值：一个在中间层，另一个在接近最后的层。然而，第一个字符的峰值往往出现在中间层，而后续字符（例如，$N = 2$ 及以后）的峰值在接近最后的层更为突出。这种趋势在所有四个模型中是一致的，表明了各种大型语言模型共享的一种普遍现象。

我们还观察到，第一个峰的位置与之前识别的突破点大致对齐。例如，在 LLaMA3-8B 中，突破发生在中深度（层 ~ 0.5）附近，知识神经元的第一个峰也出现在相似深度。相比之下，在 Gemma-7B 中，突破出现得较晚，神经元的第一个峰也相应地移到了更晚的层。这些发现表明中深层的知识神经元在拼写过程中的基础性作用。

5.2 神经元的交集

图 5 展示了不同字符位置 N 上知识神经元的重叠。在所有模型中，最大数量的神经元是唯一地用于第一个字符的预测，而第二和第三个字符则共享更多的神经元。

此外，知识神经元的总数往往从 $N = 1$ 减少到 $N = 3$。这表明，当大型语言模型启动拼写过程时，会使用更广泛和冗余的神经元集合，但随着字符位置的推进，则依赖于更少且更专业化的神经元。

对于后面的字符位置（$N = 3$，4，5），知识神经元的数量及其重叠程度在所有模型中都保持相对较小且稳定。这表明了一种内部策略的转变：虽然大型语言模型（LLMs）在 $N = 1$ 处依赖于一种广泛且共享的神经元集合来启动拼写，但它们逐渐转向使用更多位置特定和案例相关的神经元来处理后面的字符。换句话说，模型似乎通过通用机制来推广拼写的开始，但随着字符位置的推进，它适应每个标记的独特结构。

四个大语言模型（LLM）中的这些一致趋势

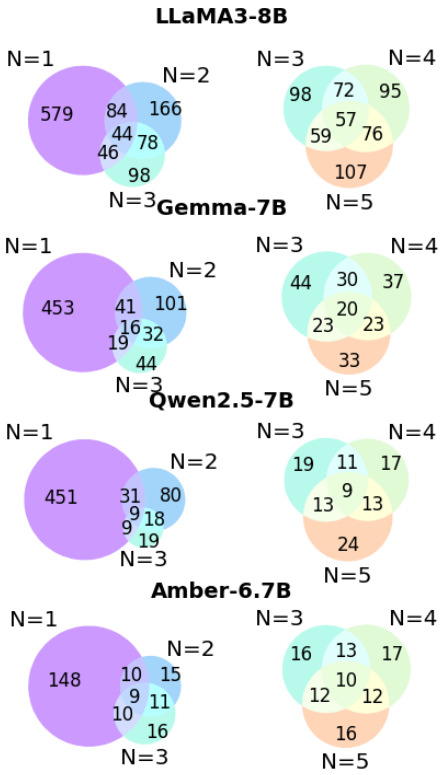

Figure 5: 维恩图显示四种大型语言模型在不同字符位置 N 上重叠的知识神经元数量。

支持了一个普遍的结论：大语言模型倾向于使用广泛且共享的一组神经元来预测第一个字符，在一定程度上也包括第二个字符，这是使用一般机制来启动拼写过程。相比之下，对于第三个位置之后的字符，它们使用更多特定情况的神经元，这表明处理过程转向更适应于个别标记结构的专业化处理。

为了更深入地理解个体神经元的作用，我们对每个字符位置 N 中的前 100 个最具影响力的神经元进行了消融，并测量了相对于图 2 中显示的初始准确性的性能下降结果。我们使用 LLaMA3-8B 进行此实验，因为它提供了足够数量的神经元以进行有意义的消融分析（图 5）。

图 6 中的实验结果显示，消融 $N = 1, 2$ 的神经元仅导致轻微的准确性下降，而消融 $N \geq 3$ 的神经元则导致性能的显著下降。这表明，后期位置的神经元在拼写任务中发挥更为重要的作用，通过利用上下文信息来补偿早期位置神经元的功能。这些结果支持了我们在第 5.2 节中的假设：早期位置的神经元主要从初始字符嵌入中提取特征，而后期位置的神经元负责基于上下文的字符预测。

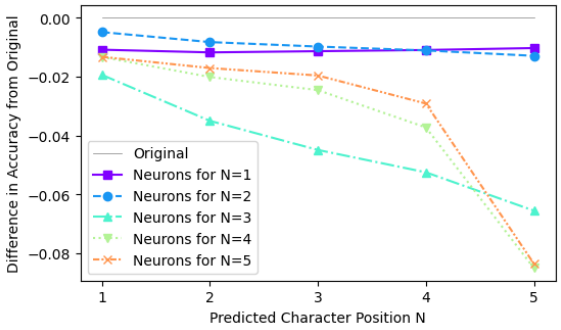

Figure 6: 消融每个字符位置的 100 个神经元后准确性的差异 N (LLaMA3-8B)。

6 拼写输出的注意力权重

鉴于拼写任务的性质，LLMs 在生成字符时必须关注目标符号。专注于正确符号的能力对于准确拼写至关重要，我们假设这种能力对应于 §?? 中识别的突破。本节进一步研究拼写能力与目标符号的注意权重之间的关系。

给定一个由目标 token 及其拼写组成的序列，我们计算所有相关元素对目标 token 的平均注意力权重。例如，当输入为 `[few-shot examples] token : t o k e n` 时，其中目标 token 是“`token`”，并且后随少量示例，我们计算每个元素对“`token`”的平均注意力权重：`token`，`:`，`t`，`o`，`k`，`e` 和 `n`。我们在每一层[3] 中的所有注意力头上对 1,000 个样本的这些注意力权重进行平均。为了分离对目标 token 的注意力，我们移除对“`<BOS>`”的注意力权重，并在平均之前重新归一化分布，灵感来自于 Kobayashi et al. (2020)。

图 7 展示了 1,000 个样本的平均注意得分。有趣的是，在四个模型中的三个中，对目标标记（红条）注意最高的层与突破点（红线）相一致。这个发现表明，中间层观察到的性能提升是由于模型越来越能正确地关注目标标记。

在 Amber-6.7B 中观察到的显著结果表明，性能突破不一定与注意力权重的行为一致。鉴于该模型分配给目标符号的注意力权重整体上低于其他模型，我们认为该模型需要关注更广泛的上下文信息，而不是专注于单个目标符号，这可能是由于其嵌入中缺乏字符级别的知识。

本文研究了一个悖论，即尽管大型语言模型（LLMs）难以识别单词中的单个字符，它们却能准确地逐字拼出单词。

我们的探测分析揭示了词元嵌入层并未编码完整的字符级信息，尤其是在第一个字符之后的部分。相反，字符级特征是在中间层和后续层中动态重建的，我们在此观察到一个明显的

[3]这些标记是从数据集中选择出来的，每个大型语言模型在少样本设置（§3）中可以正确拼写的标记。

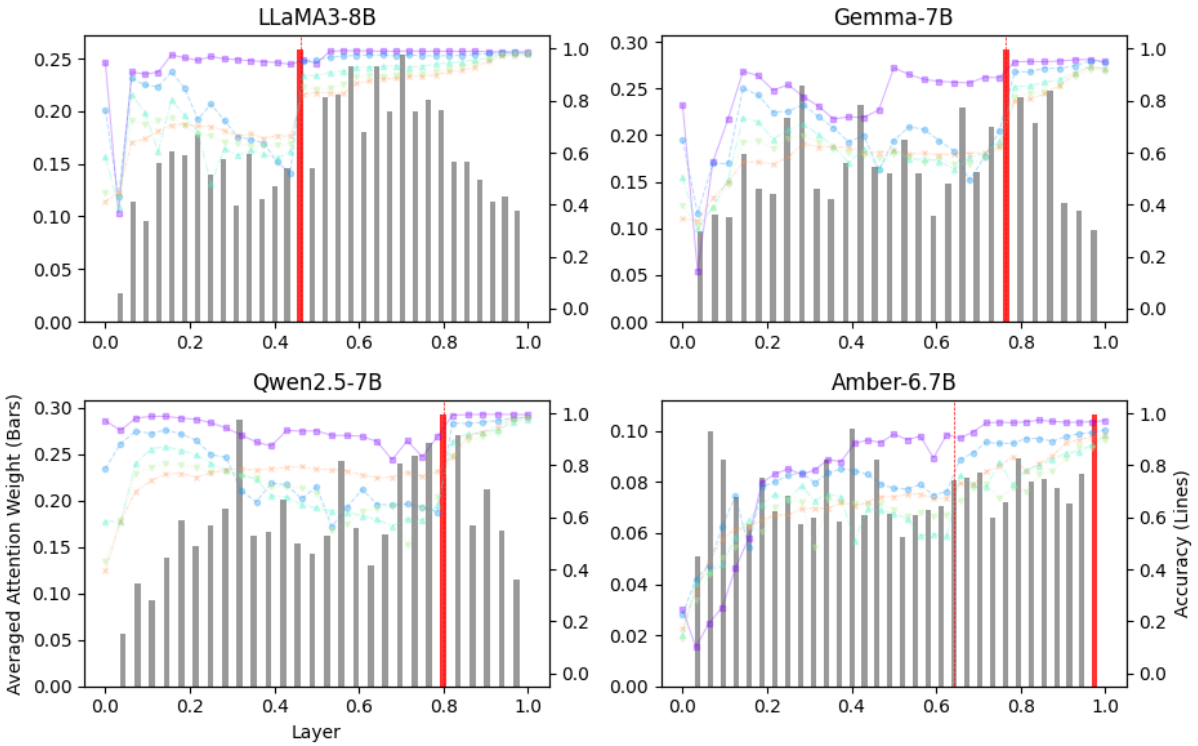

Figure 7: 跨层平均到目标标记的注意力权重（柱状图）。红色条表示具有最高注意力权重的层。折线图显示使用探测分类器预测第 N 个字符的性能，复制自图 3 。

“突破”点。在这个阶段，模型开始可靠地访问和利用字符级知识。

我们通过知识神经元和注意力模式进一步考察了拼写行为，这两者都与突破层相符。至关重要的是，这种行为表明拼写是一个学习任务（即，依赖于识别目标标记并检索字符级信息），而不是简单地提取嵌入层中存储的字符级信息。

因此，我们的研究结果表明，LLM 在拼写方面的表面成功并没有延伸至更复杂或不熟悉的任务，例如反向拼写、字母插入或在新环境中的字符层次推理。我们的研究结果显示，目前的 LLM 并不是从本质上具备字符意识；相反，它们依赖于在训练或提示过程中获得的特定任务启发。这表明模型必须明确学习如何将字符知识应用于更复杂的操作。

虽然我们认为我们的实验是在良好控制的条件下进行的，并为我们的假设提供了充分的支持，但我们承认以下限制：

- 为了保证良好的实验条件控制，我们将目标词汇限制为由小写字母字符组成的单词。当使用其他语言时，可能会出现不同的趋势。然而，考虑到之前的工作表明，LLM 可以预测汉字的初始部首（Wu et al., 2025），我们预计类似的趋势也会在各语言间出现。

- 本研究使用探测分类器、知识神经元和注意力头来调查 LLM 行为。然而需要注意的是，这些方法并不能完全捕捉或解释模型知识和能力的底层机制（Jain and Wallace, 2019; Belinkov, 2022; Kumar et al., 2022; Niu et al.）。

- 我们的实验集中在使用空格作为分隔符的拼写行为。虽然使用其它分隔符（例如“/”，参见图 11 ）时观察到了类似的趋势，但结果可能因输入格式而异。

- 我们的发现不一定可以推广到所有大型语言模型。事实上，Amber-6.7B 在几个实验中表现出不同的行为，与其他模型观察到的模式有所偏离。然而，我们认为展示这些反例对于深入理解模型行为是至关重要的。在这项工作中，我们将 Amber 的偏差归因于其嵌入层中明显缺乏字符级信息，这是一种其他模型不具备的独特特性。鉴于剩下的三个模型始终遵循相同的趋势，我们认为 Amber-6.7B 是一个特殊案例，而不是一个具有代表性的反例。

References

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. Computational Linguistics, 48(1):207–219.

Yekun Chai, Yewei Fang, Qiwei Peng, and Xuhong Li. 2024. Tokenization falling short: On subword robustness in large language models. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 1582–1599, Miami, Florida, USA. Association for Computational Linguistics.

Tyler A Chang and Benjamin K Bergen. 2025. Bigram subnetworks: Mapping to next tokens in transformer language models. arXiv preprint arXiv:2504.15471.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8493–8502.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Lukas Edman, Helmut Schmid, and Alexander Fraser. 2024. CUTE: Measuring LLMs' understanding of their tokens. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 3017–3026, Miami, Florida, USA. Association for Computational Linguistics.

Tairan Fu, Raquel Ferrando, Javier Conde, Carlos Arriaga, and Pedro Reviriego. 2024. Why do large language models (llms) struggle to count letters? arXiv preprint arXiv:2412.18626.

Tatsuya Hiraoka and Naoaki Okazaki. 2024. Knowledge of pretrained language models on surface information of tokens. arXiv preprint arXiv:2402.09808.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Go Kamoda, Benjamin Heinzerling, Tatsuro Inaba, Keito Kudo, Keisuke Sakaguchi, and Kentaro Inui. 2025. Weight-based analysis of detokenization in language models: Understanding the first stage of inference without inference. In Findings of the Association for Computational Linguistics: NAACL 2025, pages 6324–6343, Albuquerque, New Mexico. Association for Computational Linguistics.

Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. 2025. From tokens to words: On the inner lexicon of LLMs. In The Thirteenth International Conference on Learning Representations.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 7057–7075, Online. Association for Computational Linguistics.

Abhinav Kumar, Chenhao Tan, and Amit Sharma. 2022. Probing classifiers are unreliable for concept removal and detection. Advances in Neural Information Processing Systems, 35:17994–18008.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. Llm360: Towards fully transparent open-source llms. In First Conference on Language Modeling.

Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. What does the knowledge neuron thesis have to do with knowledge? In The Twelfth International Conference on Learning Representations.

Andrew Shin and Kunitake Kaneko. 2024. Large language models lack understanding of character composition of words. In ICML 2024 Workshop on LLMs and Cognition.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2024. Neurons in large language models: Dead, n-gram, positional. In Findings of the Association for Computational Linguistics: ACL 2024, pages 1288–1301, Bangkok, Thailand. Association for Computational Linguistics.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022. Finding skill neurons in pre-trained transformer-based language models. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11132–11152.

Xilong Wang, Hao Fu, Jindong Wang, and Neil Zhenqiang Gong. 2024. Stringllm: Understanding the string processing capability of large language models. arXiv preprint arXiv:2410.01208.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Xiaofeng Wu, Karl Stratos, and Wei Xu. 2025. The impact of visual information in Chinese characters: Evaluating large models' ability to recognize and utilize radicals. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 331–350, Albuquerque, New Mexico. Association for Computational Linguistics.

Zhen Xiong, Yujun Cai, Bryan Hooi, Nanyun Peng, Zhecheng Li, and Yiwei Wang. 2025. Enhancing llm character-level manipulation via divide and conquer. arXiv preprint arXiv:2502.08180.

Zhu Xu, Zhiqiang Zhao, Zihan Zhang, Yuchi Liu, Quanwei Shen, Fei Liu, Yu Kuang, Jian He, and Conglin Liu. 2024. Enhancing character-level understanding in llms through token internal structure learning. arXiv preprint arXiv:2411.17679.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115.

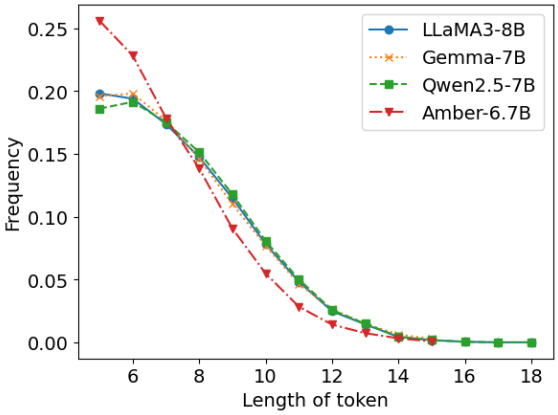

Figure 8: 数据集中的标记长度频率分布。

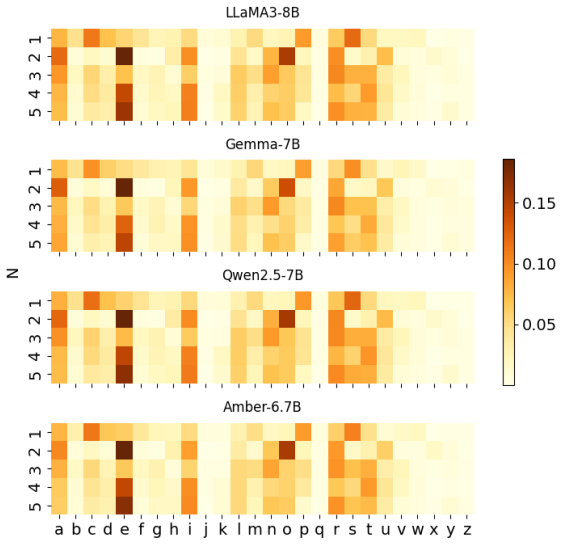

Figure 9: 数据集中每个字符位置的字母频率分布。

本论文中的所有实验均使用 HuggingFace Transformers 库进行 (Wolf et al., 2020)。除非另有说明，否则我们使用默认的超参数设置。大多数实验是在 NVIDIA A40 GPU 上进行的，涉及 Gemma-7B 的实验除外，这些实验是在 NVIDIA H100 GPU 上进行的。

我们工作中计算量最大的一部分是识别 §5 中描述的神经元，这部分处理 1000 个样本耗时超过 24 小时。所有其他实验在单个 GPU 上都在 24 小时内完成。

A 额外的数据集统计

图 8 显示了我们数据集中标记长度的分布。如图所示，具有较大词汇量的三个 LLM 表现出了相似的分布。图 9 展示了标记中每个位置的字符（字母）的分布，显示了各模型之间高度一致的模式。

图 10 展示了来自 §3 的少样本拼写准确率，作为标记长度（5-14 个字符）的函数。与直觉

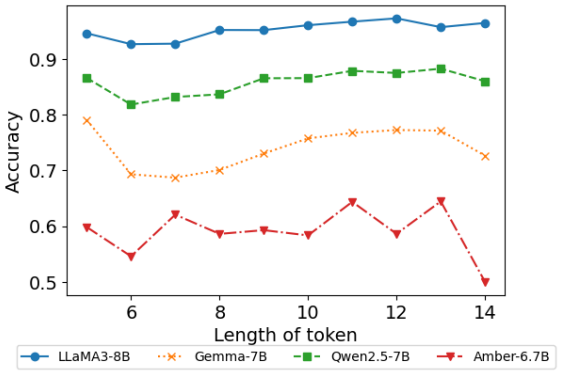

Figure 10: 按输入标记长度的少样本拼写准确性。

认为较长的标记更难正确拼写相反，较为平缓的趋势表明，标记长度对精确的标记级准确率影响有限。

B 使用“/”分隔的层探测

图 11 展示了当用于拼写的分隔符从空格更改为斜杠（“ / ”）时，对分类器进行探测的结果。实验设置与 §?? 中相同，除了分隔符的更改。在少量样本例子、目标输入和期望输出中一致使用了此修改后的分隔符（见图 2）。例如，输入“ `hello :/h/e/l/l/o/` ”在其中一个例子中使用。

为了确保字符位置之间的表示一致性，我们在拼写之前和之后添加了一个斜杠，这样模型就不需要使用带有特殊前缀来表示单词头的标记。相比之下，主要实验使用诸如“ `_h_e_l_l_o` ”这样的输入，其中下划线（“_”）表示前缀。

如图 11 所示，我们观察到与图 3 类似的趋势，包括在大约同一层出现“突破”的现象。这种一致性表明，我们关于突破行为的发现可能会推广到拼写任务的不同输入格式。

C 用于字母的神经元

图 12 显示了与每个字母字符的输出相关的知识神经元的分布。我们使用了与 §5 中描述的相同识别方法，但专注于单个字符而非位置索引（N）。结果表明，负责字母输出的神经元主要位于模型的近末层。

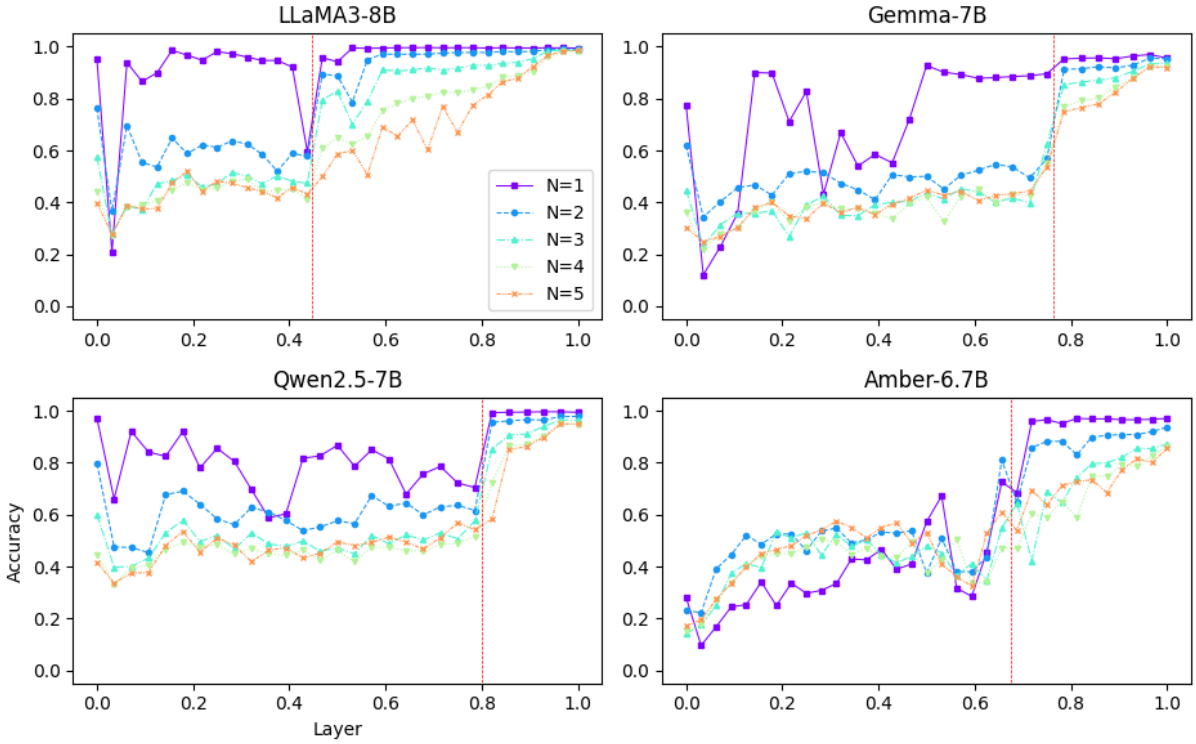

Figure 11: 使用 “ / ’’ 分隔符的 N -字符预测准确性，通过探测分类器跨 Transformer 层进行测量。

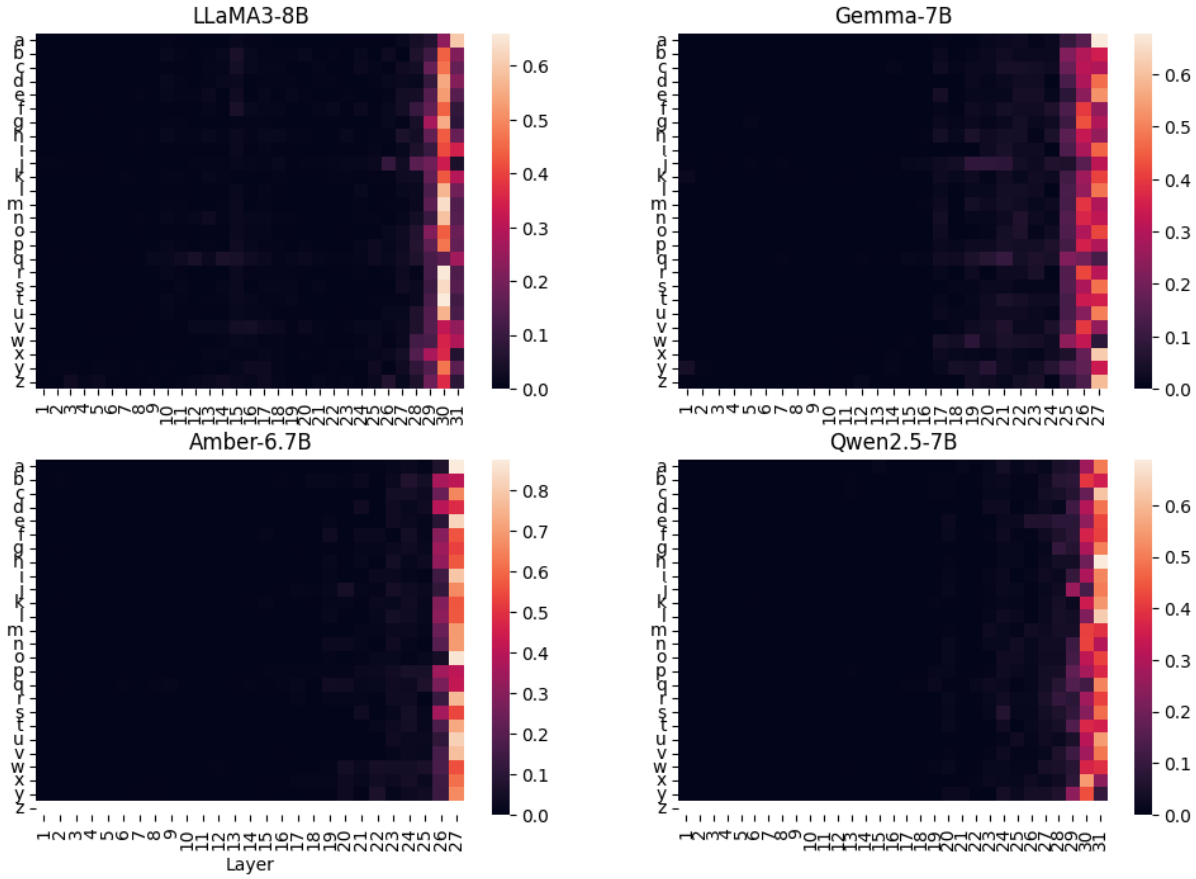

Figure 12: 负责输出每个字母字符的知识神经元的分布。