程序理解并不主要依赖人类大脑的语言中枢

Shashank Srikant[1,2], Anna A. Ivanova[3], Yotaro Sueoka[3], Hope H. Kean[3], Riva Dhamala [4], Evelina Fedorenko[3], Marina U. Bers[4], Una-May O'Reilly[1,2]

{shash, annaiv}@mit.edu, evelina9@mit.edu, unamay@csail.mit.edu[1]计算机科学与人工智能实验室，麻省理工学院 [2]麻省理工学院-IBM 沃森人工智能实验室 [3]脑与认知科学系及麦戈文脑研究所，麻省理工学院，[4]儿童研究与人类发展系，塔夫茨大学埃利奥特·皮尔逊系

arXiv:2304.12373v1 [cs.SE] 11 Apr 2023

本稿改写自**Ivanova, A.A.,Srikant, S., Sueoka,Y., Kean, H.H.,Dhamala, R.,O'Reilly, U.M., Bers, M.U.and Fedorenko, E., 2020.** 计算机代码的理解主要依赖领域一般性执行脑区。《**eLife**》，**9**，**第 e58906** [37]页。本文面向计算机科学读者介绍《eLife》论文的研究结果；原《eLife》论文主要旨在向认知神经科学界介绍相关工作。

摘要

我们的目标是确定参与理解计算机程序的脑区。我们使用功能性磁共振成像（fMRI），研究两种可能支持这一过程的脑区系统：已知会对各种认知要求较高的任务作出反应的多重需求（MD）系统，以及已知主要会对语言刺激作出反应的语言系统。我们设计了实验条件，以分离代码理解这一行为，并采用最先进的方法定位感兴趣的脑系统。我们分别使用 Python（24 名参与者）和 ScratchJr（19 名参与者）开展实验；ScratchJr 提供了编程可视化界面，因此消除了文本对代码理解的影响。通过这一稳健的实验设计，我们发现，语言系统并未始终参与代码理解，而 MD 系统则始终参与其中。此外，我们发现，除 MD 系统中的脑区外，没有其他脑区会对代码作出反应。我们还发现，变量名、程序所采用的控制流以及所执行操作的类型均不会影响脑反应。我们讨论了这些发现对软件工程和计算机科学教育领域的启示。

关键字

神经影像学、功能性磁共振成像（fMRI）、人脑、代码理解、语言系统、多重需求系统、人因、Python、ScratchJr

1 引言

据估计，阅读和理解计算机程序（代码）几乎要占用软件专业人员60%的时间 [58]。然而，我们对人类如何通过认知过程完成这一任务知之甚少，因此这仍是科学领域中的一个开放性问题。在开创性先例的基础上 [27, 54], ，本研究尝试探究并确定参与理解计算机代码的脑区。

代码理解作为一种认知技能出现的时间较晚，这表明，专门支持其他认知活动的脑区可能也支持代码理解。鉴于代码理解与逻辑和问题解决相关，可以认为，它可能由负责工作记忆和认知控制的脑区，或参与数学和逻辑处理的脑区完成。同样，代码和自然语言有许多共同属性。二者具有相似的句法和语义结构，并通过层级组合来传达有意义的信息——无论是在代码还是文本中，词元都会组合成语句，语句再进一步组合成完整的代码或文档，从而使意义与制品相关联 [25]。可以认为，参与语言处理的脑区能够支持代码理解。

神经影像学研究非常适合用于确定哪些脑区参与代码理解。功能性磁共振成像（fMRI）等技术可以测量人们执行阅读或听音乐等认知任务时的大脑活动。当功能已得到充分确立的脑区（如语言或音乐中枢）对一项新任务（如代码理解）产生反应时，这可以说明可能与该任务相关的认知过程 [43]。

在本研究中，我们使用功能性磁共振成像来探究代码阅读相关任务如何调动两个已知的脑区系统，即多重需求（MD）系统和语言系统（详见第3节）。尽管以往的神经影像学研究也考察了参与代码理解的脑区，但其结果仍未达成一致。这些研究为大致对应于MD系统的脑区 [27, 34, 42, 54, 55], 以及类似于语言系统的脑区 [54, 55]中的活动提供了证据。重要的是，这些研究没有将代码理解行为与其他代码阅读相关活动（例如代码心理模拟）区分开来。此外，大多数研究都没有量化脑反应，也没有将其与工作记忆或语言相关的其他任务所引发的反应进行比较，从而无法对其观察结果作出有意义的解释。我们在第2节中回顾这些研究，并在第4节中将它们的设计选择与我们的设计进行对比。

我们在这项工作中的贡献主要有两点。首先，我们设计了新颖的实验，并引入了改进的方法，以识别参与代码理解的脑区。其次，我们提出了

一组新的结果，进一步加深了我们目前对代码理解认知基础的认识。下面我们总结了我们的设计和方法贡献。有关结果的详细信息，请参见第 6 节。

- 我们提出了对代码理解更清晰的定义，并设计实验条件以分离和测量代码理解。
- 我们采用最先进的程序，以确定哪些已知且特征明确的脑系统会对代码理解产生反应。
- 我们在两种编程语言中进行实验：Python 和 ScratchJr。ScratchJr 是一种具有完全可视化界面的编程系统，参与者分别为 24 人和 19 人。使用 ScratchJr 能够测量文本对代码理解的影响，还有助于验证我们结果的普适性。此前的研究仅使用过一种编程语言进行实验。
- 我们还确保所观察到的结果能够推广到不同的代码属性，例如控制流（顺序程序、循环、条件语句），或所执行操作的类型（字符串、数学运算）。
- 我们还研究了语言系统中与代码相对应的大脑活动是否源于代码中所使用的描述性变量名。
- 我们公开提供代码、刺激材料和脑数据，供社区重复使用和扩展。链接 - https://github.com/ALFA-group/neural-program-comprehension

2 相关工作

关于人类大脑中是否存在专门负责特定认知功能的区域这一问题，可以追溯到 Paul Broca 在 19 世纪 50 年代对语言理解的研究 [31]。过去三十年中，用于准确测量神经活动的技术不断进步，揭示了负责多种认知功能的专门区域的存在，例如语言处理、面孔识别、导航等。 [39]

近年来，利用神经影像学技术研究编程引发的认知反应逐渐受到重视。以往的研究考察了阅读程序时涉及的调试[14], 变量跟踪、语义线索或程序布局 [36, 46], 程序生成 [18, 51], 操纵数据结构 [41],代码审查过程中的偏差 [34], 以及编程专长 [33], 。

与我们的研究范围相关的是那些探究参与代码理解的脑区的研究（而非参与编写代码或任何其他与编码相关的活动的脑区）。

Siegmund 等人 [54],开展了一项具有影响力的研究，将学界的注意力引向这一主题，探究大脑中哪些脑区参与代码理解。他们在一项功能性磁共振成像研究中，向17名参与者呈现了两组刺激材料。第一组要求参与者阅读代码片段并确定其输出结果。第二组要求参与者阅读包含语法错误的代码片段，并提出修正方案。作者将这两组刺激材料所引发的激活与无活动的基线进行对比；这两组刺激材料均对应于大脑中的代码理解活动。他们表明，这一对比的一部分位于根据布罗德曼分区界定的布罗卡区（语言中枢） [12]。

Floyd 等人 [27]提出了一个不同的主要研究问题。他们在更大的样本中，即29名参与者中，探究是否可以利用脑活动测量来区分程序理解与英语句子理解。他们的解码实验表明，代码的神经表征具有独特性，并且不同于语言。作为次要结果，他们确实讨论了参与其中的脑区，并部分证实了 Siegmund 等人的发现。在研究设计中，他们采用英语理解任务与两项代码阅读任务的基线对比。

Liu 等人 [42] 最近表明，代码理解与大脑的语言中枢重叠程度非常低，这与我们在本文中提出的结果一致。他们让17名专业程序员完成两项与代码相关的任务——第一项任务与Siegmund 等人的研究相似，要求参与者确定代码输出。第二项任务要求参与者记忆他们所谓的“伪代码”——每一行都包含打乱的词元的代码片段——并确认特定子字符串是否存在。此外，他们还安排数学、逻辑和语言任务，以定位每位参与者执行这些功能时涉及的脑区。他们报告称，代码活动与属于MD系统的脑区存在重叠，但与语言中枢没有重叠。

我们提出了与Siegmund 等人 和Liu 等人 研究相同的问题。不过，我们在实验设计和工作流程上有所不同。我们在第4节中详细比较了我们的设计选择与这些研究的异同。我们将看到，这会得出一组不同于Siegmund 等人研究结论的结论。

3 背景

我们简要介绍 fMRI研究的背景、此类扫描仪所测量的内容，以及我们所研究的脑区。

3.1 fMRI研究

功能性磁共振成像（fMRI）通常用于识别对某项认知任务作出反应的脑区（在我们的研究中，该任务是理解代码）。MRI设备能够标记并显示多达约一百万个体素的脑反应，每隔几秒进行一次采样 [28]。体素大致相当于像素的三维等价物，覆盖我们大脑中数立方毫米的空间。

当某个脑区参与一项认知任务时，血流会流入该区域，以帮助其进行处理。磁共振成像仪会测量这种血流变化，并报告以仪器频率采样的血氧水平依赖（BOLD）值。按照惯例，将一般线性模型拟合到这些随时间变化的数值上，并将其参数用作脑活动的度量。详细信息见第5节。

3.2 感兴趣区域（感兴趣区域）

我们考察两个经过充分研究、且已知如何定位的脑区系统——MD系统和语言系统——在理解代码时是否也会被激活。区域（也称为分区）在此指参与某项认知任务的一块连续脑组织。脑区系统（也称为网络）可以由多个互不相连的（在皮层层面上）脑区组成，而这些脑区都参与同一项认知任务。

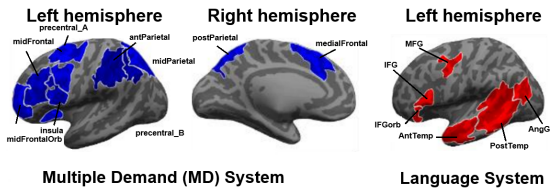

图1：神经典型成人大脑中突出显示的多重需求（MD）系统和语言系统。这两个系统横跨大脑中多个彼此相近的区域，并且已被证实具有截然不同的反应特征。通常所称的布罗卡区包含这两个系统的部分区域 [21]。

多重需求（MD）系统。由于编程可以被认为涉及算术和一般逻辑技能，我们考察多重需求系统 [16], 这一已知支持这些技能的最主要系统是否会被激活。该脑区系统通常位于大脑的前额叶和顶叶区域，已知不受特定领域限制，并且会在大量需要工作记忆和一般问题解决技能的任务中被激活，包括数学和逻辑 [4, 16]。

语言系统。处理代码的另一个可能候选是语言系统。这些区域已被确定能够对各种模态（书面语言、言语、手语）中的语言理解和语言产出作出反应，能够对类型学上多样的语言作出反应（来自10个语系的> 50 种语言），构成功能上整合的系统，能够可靠且稳健地追踪语言刺激，并且已被证明对语言具有因果上的重要性 [9, 10, 15, 24, 45, 52]。

图1展示了神经典型成人大脑中这些系统的大致位置。across individuals [23, 24]这些系统在不同个体中始终大致位于大脑的相同部位。虽然ROI提供了一组在不同个体中被观察到参与某项认知任务的广泛区域，但我们进一步定位了功能性 ROI （ fROI ） —— 这些广泛区域内分别对工作记忆和语言作出反应的特定体素在个体中。通过这种方式，我们将这些体素的确切解剖位置纳入考虑，而这些位置因个体而异。这是我们的实验方法相较于以往研究的一项改进之处。我们在第4.4节中提供了关于fROI的详细信息。

4 实验设计

我们首先概述整体工作流程。随后，我们将详细介绍实验条件设计的三个关键组成部分：**条件设计** ——创建向参与者展示的代码刺激材料时所考虑的各种设计选择；**功能性磁共振成像任务**——参与者在磁共振成像仪中作出反应的任务，这些任务使我们能够测量脑活动；以及**处理功能性磁共振成像数据** ——我们如何分析参与者的功能性磁共振成像数据，并量化代码理解的影响。在介绍这些组成部分时，我们还将对比它们与先前研究的不同之处。

4.1 实验流程——概览

我们工作流程的第一步是提出假设，并设计能够检验这些假设的条件。这些条件决定了我们在磁共振成像仪中向人类参与者呈现的刺激材料和任务。我们的目标是观察阅读代码对大脑中两个感兴趣区域的影响，即MD系统和语言系统。我们首先确定每位参与者的哪些体素（功能性兴趣区）属于MD系统和语言系统。为此，我们让参与者完成定位任务 ——这些任务已被证明能够持续激活这两个系统 [9, 24]。随后，我们向参与者展示与我们精心设计的代码条件相对应的刺激材料，并测量这些已识别功能性兴趣区内对这些条件产生的脑反应。分析针对代码条件的功能性磁共振成像反应，旨在评估这些条件激活功能性兴趣区的程度是否与定位任务相当。如果它们对相关区域的激活程度相当，我们便推断这些功能性兴趣区参与了代码处理。例如，如果理解代码对语言系统的激活程度与理解英语文本（语言系统的定位任务）相当，那么我们便得出结论：语言系统除了参与语言理解之外，也参与代码理解的处理。

4.2 条件设计

给定条件下的脑活动测量（例如 阅读代码时的反应）只有相对于另一种条件（例如 阅读纯文本时的反应）才具有明确的解释意义，即 通过对比 两个或更多条件。我们将介绍本研究中设计并进行对比的不同条件，并结合先前研究所采用的设计选择对这些条件进行讨论。

控制 非代码。观察阅读代码效果的最简单条件组合，是将代码与非代码进行对比（在认知神经科学文献中记作 code > non-code）。在这里，非代码对应的是参与者即使面对不像代码的刺激材料也能够理解的内容。在我们的研究中，这些内容对应于自然语言（英语）中的语句。

不过，我们的目标是进一步深入。我们设计了一些条件，以帮助分离其他因素的影响；当人们理解代码时，这些因素也可能解释我们在不同大脑区域观察到的激活。

控制 代码模拟。可以认为，阅读代码的任务不仅仅涉及代码理解这一行为。要理解其中的原因，可以考察阅读和理解代码所涉及的不同认知步骤。在呈现代码后：1）程序中字符的存在会激活视网膜细胞；2）我们大脑的视觉系统会处理这些字符；3）在识别字符后，我们的大脑会解释文本中存在的词元；4）我们的大脑将词元分组以识别程序语句，随后将这些语句分组，形成对整个代码的心理表征，并理解其目标；5）我们的大脑执行或模拟该程序，以推导出最终输出。

在我们的研究中，我们不探讨阅读代码对视觉系统的影响（步骤 1-2）。我们将步骤 3-4 视为代码理解，将步骤 5 视为执行代码模拟——这也被称为程序跟踪 [56],和处理代码内容 [37]。例如，理解语句 `x=10+20` 是指将该语句与“x 存储着 10 和 20 的和”这一概念联系起来，

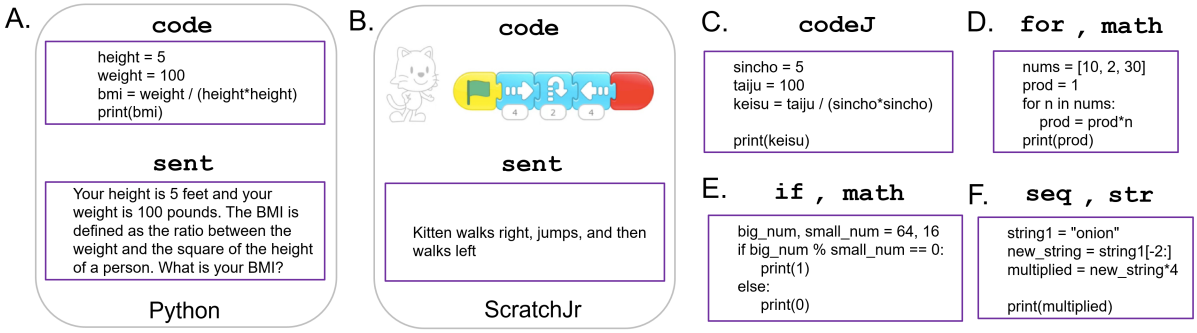

图 2：（A）Python 中的一个代码条件刺激及其等价的句子条件，后者用文字描述代码刺激。句子条件控制了大脑对代码模拟**的反应。这些条件之间的差异，即代码>句子，估计了**代码理解**。（B）ScratchJr 中的一个代码刺激和句子刺激示例。ScratchJr 是一种具有可视化界面的编程系统，可用于测量文本在代码中的影响。（C）使用日语变量名的 codeJ 条件，用于控制有意义变量名的影响。（D、E、F）用于测量控制流属性（for循环、if条件语句、顺序）以及操作类型（数学、字符串）影响的条件。**

数字 10 和 20”。在此情形下，代码模拟 是指在脑海中将数字 10 和 20 相加，并意识到 x 存储的是 30。我们将步骤 3-5 统称为代码阅读。

步骤 5 可能会主导阅读和理解代码时获得的脑测量结果。为了排除其影响，我们提出以下见解：在保留代码模拟 操作的同时，可以用不同方式描述代码。用句子或流程图描述代码，并不会改变其操作。基于这一见解，我们设计了内容与代码条件相匹配的句子。我们将该条件记作 sent，并将对比记作 code ＞sent。示例见图 2.A。如果代码条件主要测量代码理解 和代码模拟 这两个主要认知步骤，那么 sent 条件可以被认为测量自然语言（句子）加工和代码模拟。因此，这两个条件 code ＞sent 之间的差异，使我们能够分离并测量代码理解这一行为。**控制变量名的影响**。 如果代码理解 确实被当作语言理解来处理，并且发现语言系统会对其产生反应，那么有理由质疑：语言系统的反应是否仅仅由有意义变量名的存在引起，而不是由代码的其他方面引起。我们通过将变量名替换为在该语境中没有意义的名称，来控制这一可能性。对这类代码的反应，便可以归因于代码理解 本身，而不是代码中有意义的英语单词。我们的研究选择将变量重命名为对应的日语名称（使用英语字母拼写），并将其呈现给不懂日语的参与者。我们将该条件称为 codeJ。图 2.B 展示了图 2.A 中的代码，但使用了日语变量名。我们还考虑了有意义的字符串字面量（例如x="hello"）或有意义的关键字（for、if）可能产生的影响：我们设计了数量相等、但不包含这些制品的刺激材料（将在下一点中讨论）。

控制流和操作的影响。 我们还进一步考察了代码引发的脑激活是否在不同代码属性之间保持一致。这有助于证明我们的观察结果对于代码中常见的可能变体具有稳健性

。我们测试了其中两种属性——控制流和操作类型。在控制流方面，我们分别测试了循环（for循环）、条件语句（if条件语句）和顺序语句。有关这些条件的示例，请参见图 2.D、E、F。我们测试了两种操作类型——数学和字符串。图 2.E、F 分别展示了数学操作和字符串操作的示例。这些条件下的每个刺激恰好包含三个控制结构中的各一个，以及两种数据操作中的一种。该设计还通过允许我们观察不包含有意义的字符串字面量和关键字的条件（分别为数学和顺序执行）所对应的脑活动，考虑了这些制品的存在。

代码中文本的影响。我们在两种编程语言——Python 和 ScratchJr——中对上文描述的条件进行实验。ScratchJr 是一种具有完全可视化界面的编程系统 [7]。通常，ScratchJr 被介绍给儿童，作为一种创造性地表达自我的方式；其可视化界面以及代表不同编程构造的直观拖放功能，使儿童无需依赖英语之类的语言即可进行编程 [8]。这种可视化界面的本质使我们能够排除文本对代码理解的影响。图 2.B 展示了一个示例。此外，使用 ScratchJr 作为第二种编程语言，有助于验证我们研究结果的普适性。此前所有研究都只在一种编程语言中评估其研究结果。

既有研究的设计选择。 Floyd 等人 也采用基本的代码 ＞非代码对比，但没有进一步的设计来单独考察代码理解。Siegmund 等人 则将代码 ＞代码与语法错误进行对比（第2节）。包含语法错误的代码仍然是代码，因此无法帮助区分那些已知会处理非代码（自然语言）的脑区中的活动。此外，包含语法错误的代码条件很可能测量代码理解、代码模拟以及其他可能特有于调试和查找此类错误的技能。因此，他们的对比并不能完全单独考察代码理解。虽然 Liu 等人 确保其代码刺激材料可推广到循环和条件，但他们主要采用的代码 ＞伪代码对比同样无法区分代码

理解 和代码模拟。其设置引入了记忆伪代码这一额外影响，而这涉及多个认知过程

摘要。 总而言之，在我们的 Python 实验中，整体实验设计是一项 $3 \times 3 \times 2$ 研究，包含 3 种条件：代码、句子、codeJ（日语变量名）；在这三种条件中的每一种下，我们还进一步设置了 3 类控制流条件和 2 类操作相关条件。由于其中许多条件不适用于 ScratchJr（变量名、操作类型），我们在 ScratchJr 中仅评估关键的代码 > 句子条件。

4.3 功能磁共振成像任务

对于每位参与者，除了在磁共振成像仪中呈现与代码相关条件对应的刺激材料外，我们还安排了两项独立任务，以定位 他们的两个感兴趣区域。定位任务的核心在于能够在每位个体中强烈激活某个感兴趣区域。已有实证表明，阅读语义结构正确的任何自然语言句子都会强烈激活语言系统，而执行空间记忆任务则会强烈且有区别地激活 MD系统 [23, 24]。我们在研究中沿用了这些经过验证的定位任务。详情见第 5 节。

现在，我们将介绍在分析代码理解过程中的脑活动时，如何使用这些定位信息。

4.4 定位功能性兴趣区和数据分析

我们通过以下五个关键步骤分析脑数据。我们采用群体约束的个体特异性（GSS）方法，定位在不同个体中持续被激活的功能性兴趣区（fROIs） [47]。

1. 映射到典型脑结构。 为了对脑部解剖结构的差异进行标准化处理，我们将每位参与者的大脑在空间上转换到类似蒙特利尔神经病学研究所（MNI）模板的典型脑结构 [57]。这些经过空间转换的坐标将用于后续分析。

2. 选择感兴趣区域。 感兴趣区域（ROIs）标示出一组被观察到在不同个体执行某项认知任务时参与其中的广泛区域。对于每位参与者，我们以这些区域为起点，在其中寻找对某项认知任务产生反应的体素。这有助于避免在与任务无关的区域中进行搜索。例如，阅读代码显然也会激活视觉皮层，而视觉皮层并不是本研究关注的区域。

在我们的工作中，我们重复使用了先前研究定义的一组 20 个 MD 分区（每个半球 10 个）和 6 个语言分区 [23, 24]。这些分区是通过分别汇总 ~200 参与者对空间工作记忆和语言任务的脑反应整理而成的。另一种做法是，从布罗德曼分区 [12], 图谱所定义的分区中选择感兴趣区域；该图谱将典型个体大脑的区域映射到认知功能。

3. 识别功能性兴趣区。对于每位个体，功能性兴趣区（fROI）是指兴趣区内一组会对该兴趣区所涉及的认知任务产生反应的体素。由于解剖结构存在差异，会对某项认知任务（如空间推理或语言）产生反应的具体体素集合因个体而异，而这些差异遍布个体之间。由多个个体汇总得到的兴趣区通过指出一片已知会对该任务产生反应的区域，帮助缩小搜索空间，从而在每位个体中定位这些特定体素。功能性兴趣区则进一步识别出在认知任务中发挥功能作用的特定体素。每个系统的定位任务（第4.3节）有助于识别这些体素。完成此步骤后，我们会在每位参与者中为MD系统和语言系统建立功能性兴趣区。详细信息见附录1。

4. 在参与者内部汇总激活数据。在后续所有实验条件中，我们都使用上一步为这两个系统定义的功能性兴趣区。具体而言，我们测量所选功能性兴趣区中代码条件的激活情况。在此阶段，对于功能性兴趣区中的每个体素，我们至少有两组激活测量值：一组对应定位任务，其余各组对应不同的代码相关条件。对于每个功能性兴趣区，我们通过对其中所有体素的反应取平均值，获得每种条件下的单一反应值。

5. 汇总参与者之间的激活数据。随后，对于每个系统，我们评估参与者层面的响应在代码条件下的分布是否与定位任务的分布具有可比性。如果是，则我们得出结论：该系统参与了代码条件的加工。

不使用功能定位任务的多参与者分析。在以往的研究中，只有 Liu 等人使用定位任务来找出单个参与者中具有任务选择性的体素。然而，他们没有使用兴趣区（上述第2步），而是进行全脑分析，并报告重叠情况，而不是测量功能性兴趣区中的确切激活。他们的实验设置，加上其含义模糊的条件设计（将在第4.2节讨论），使得准确推断脑区变得困难。

在不使用定位任务信息的fMRI研究中，例如 Sieg mund 等人、Floyd 等人以及其他研究编程不同方面的工作，主要区别在于，兴趣区是依据解剖结构而非其功能来定义的（即它们对定位任务的响应方式）。具体而言，这一区别体现在第3步和第4步：研究者不是汇总功能性兴趣区内的激活，而是在整个大脑的每个体素中估计激活，并在参与者之间进行汇总（也称为群体分析过程）。随后，使用解剖学标签（例如布罗德曼区 [12]）描述这些汇总后活跃体素的位置。在神经影像学研究中，这种方法通常被称为逆向推理 [49]。

反向推断方法假设功能性兴趣区在不同个体之间具有固定的空间位置，并且可以在标准大脑结构中被唯一定位。虽然逆向推理并不总是一个问题，尤其是在各区域在解剖结构上彼此分离且界限清晰时（例如，视觉系统与MD系统），但已有研究表明，在测量位置相近的MD系统和语言系统时，这种方法会产生不准确的估计 [5, 11, 21, 22]。在某个个体中，布罗德曼分区（44区和45区）所指的语言区域，在另一个个体中可能实际上对应于属于MD系统的功能区域，这是由于个体解剖结构存在差异 [20, 22]。基于功能的兴趣区识别GSS方法有助于规避这一潜在的不准确来源。

5 实验程序

我们简要介绍实验程序。详细信息请参见附录 A。

我们招募了24名参与者参加实验1（Python），以及19名参与者参加实验2（ScratchJr），两组之间没有人员重叠。在扫描当天，参与者提供知情同意后，在扫描仪中度过了 1.5 – 2 小时。在实验1中，在预定进行功能性磁共振成像扫描的那一周，所有参与者还完成了一项Python评估，以评估其Python熟练度（附录A.1）。

进入扫描仪后，向参与者呈现了两个定位任务，这些任务改编自先前的研究 [23, 24]，分别用于定位其大脑中的MD系统和语言系统（附录 A.2）。MD系统定位任务是一项工作记忆任务，包含两个难度等级——简单和困难。语言系统任务包含两个条件——句子阅读（SR）和非词阅读（NR）。SR要求阅读结构和语义均有意义的句子。NR要求阅读由可发音但无意义的非词组成的句子（例如BIZBY ACWORILLY BUSHU SNOOKI BILIBOP KUKEE）。这两个条件作为参照，用于衡量其他实验条件——已有研究表明，语言系统对SR的反应强烈，而对NR的反应很弱。

参与者还完成了编码任务（附录 A.3）。所呈现的任务在三种条件之间保持平衡：codeE（使用英语中语义上有意义的变量名的代码）、sent（描述程序的句子，并针对代码模拟进行控制）和 codeJ（使用日语变量名的代码）。每位参与者完成了 72 道题目，每种条件各 24 道。这 24 道题目中的控制流条件和操作相关条件数量也相等。每位参与者针对同一道题目只会看到三个版本中的一个（附录 A.4）。

定位扫描所得的数据用于在每位参与者的大脑中定位 MD 和语言系统中的功能性兴趣区（附录 A.5）。我们对不同任务响应所产生的时间序列脑激活数据拟合了一般线性模型。该模型的参数（β）被用作我们所有分析中脑活动（血氧水平依赖）的度量指标（附录 A.6）。

6 结果

我们在此提出研究问题及其相应的结果。在结果部分，我们讨论大脑不同脑区的神经激活（图 3.A、3.B）。这些图中的 x 轴对应参与者作出反应的不同条件，y 轴表示激活强度（β 值，详情见附录 A.6）。每个柱形中的每个点代表一名参与者在其功能性兴趣区中的汇总活动。在报告任意两个条件之间的对比结果时，我们测量平均 β 值（$\Delta\beta$）之差，并计算其对应的 p 值。

研究问题1。 代码阅读会激活多重需求（MD）系统吗？

对比的条件。代码、句子阅读、非词阅读

我们首先探究代码阅读是否会激活MD系统。代码阅读同时涉及代码理解和代码模拟。为此，我们将主要代码相关条件——代码问题——的激活情况，与语言系统的定位任务——句子阅读和非词阅读——进行比较。在图3.A、B中，我们分别将这些条件记为CP、SR和NR。我们评估了MD系统中的两组功能性兴趣区（fROIs）——大脑每个半球各一组（图3.A、B中的左图和中图）。从图中可以看出，句子阅读和非词阅读这两种语言定位条件在两个大脑半球的MD系统中都只有极少的激活。这是意料之中的，因为MD系统对语言任务并不敏感 [9]。我们发现，在两项实验中，同时涵盖代码理解和代码模拟的代码问题，在MD系统两个大脑半球的功能性兴趣区中引发了稳定且显著高于基线的激活（Python： $\Delta\beta = 2.17$，$p < 0.001$；ScratchJr： $\Delta\beta = 1.23$，$p < 0.001$）。这表明MD系统参与代码阅读。

我们检验这些反应在不同代码属性下是否一致，以此确定其对代码中可能出现的变化是否具有稳健性。我们在 Python 中测试了两种属性——控制流（顺序执行、for循环、if条件语句）以及其中所操作的数据类型（字符串、数学运算）。无论运算类型和控制流如何，我们都观察到强烈的反应（图3.C；纵轴表示对 CP 的反应）。因此，我们得出结论：MD系统对代码问题的反应并非源于某一种特定类型的问题，也不是与某种特定控制流相关的心理运算所致。

这清楚地揭示并确立了 MD系统在代码阅读中的作用。以往研究既未识别，也未研究这一组脑区。

研究问题1的结果。 是的，代码阅读 会激活 MD系统。其反应不受代码中控制流运算和数据运算类型的影响。

研究问题2。 代码理解 会激活多重需求（MD）系统吗？ **对比条件**。 代码、句子、句子阅读、非词阅读

既然我们发现代码阅读会激活MD系统，我们便进一步探究这些反应究竟是对代码理解还是代码模拟的反应。为此，我们研究了两种与代码相关的条件的影响：代码问题（CP），以及内容与代码问题相匹配的句子问题（SP）。我们发现，对于Python，测量的仅是代码模拟而非代码理解的句子问题，在左右两个大脑半球中激活MD系统的程度均显著高于语言定位基线（左： $\Delta\beta = 1.51$，$p < 0.001$；右： $\Delta\beta = 0.78$，$p < 0.001$）。对于ScratchJr，这种激活并不显著（左： $\Delta\beta = 0.09$，$p = 0.93$；右： $\Delta\beta = -0.40$，$p = 0.004$），这表明代码模拟

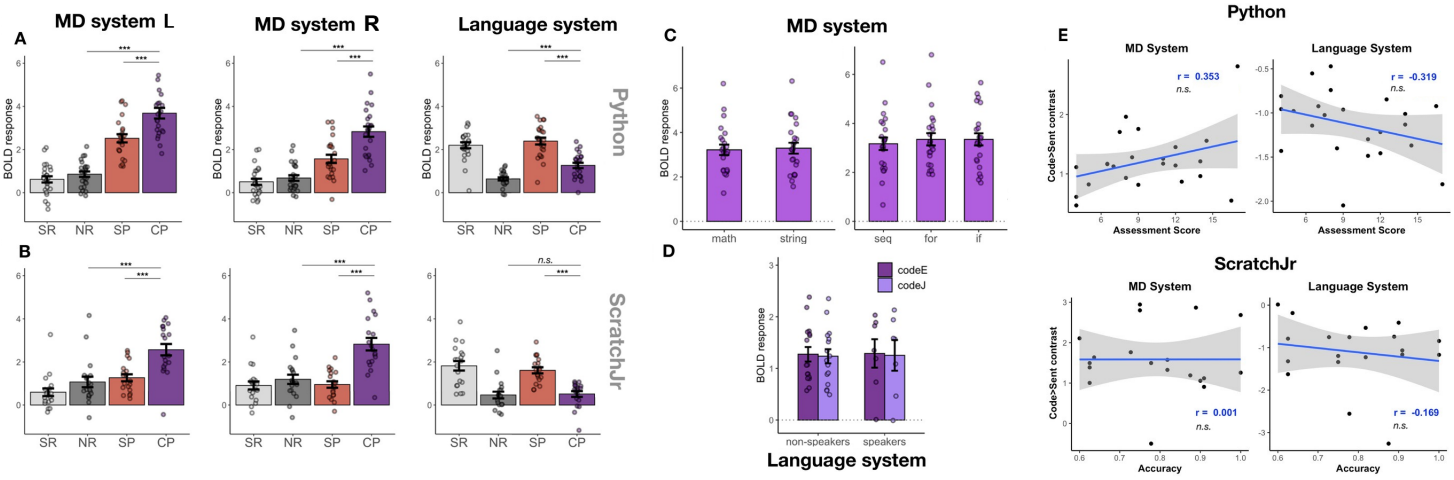

图3：（A、B）MD系统左半球（MD system L）、MD系统右半球（MD system R）和语言系统中的脑激活。我们测量了四种条件下的反应——代码（CP）、与代码操作相匹配的句子（SP）、句子阅读（SR）和非词阅读（NR）。实验使用 Python（N=24）和 ScratchJr（N=19）。每个柱状图中的点对应一位参与者的汇总数据。* 表示 $p < 0.001$，n.s. 表示不显著。（C）MD系统对两种代码属性的反应——操作类型（数学、字符串操作）和控制流（顺序执行、循环（for）、条件语句（if））。（D）语言系统对英语（codeE）和日语（codeJ）变量名的反应。（E）MD系统和语言系统的反应与 Python（上）和 ScratchJr（下）熟练度的相关性。**

并未始终得到MD系统的支持。不过，我们发现，同时测量代码理解和代码模拟的代码问题会强烈激活两个大脑半球中的功能性兴趣区。尽管句子问题的平均作答时间稍长，情况依然如此（附录B.2）。因此，我们发现CP会强烈激活MD系统，而SP不会。这意味着，衡量代码理解的差值CP ＞SP会强烈激活该系统。这有力地证明，MD系统会持续、稳健地响应代码理解，并表明它的反应并非仅仅源于底层的代码模拟操作。

我们进一步考察了代码理解是否存在大脑半球偏侧化。既有研究表明，数学和逻辑问题通常会激活大脑左半球的MD系统 [3, 4]。我们在Python中未发现这种偏侧化现象（MD-左侧图，图3.A）。在ScratchJr中，我们观察到右半球的反应更强（$\Delta\beta$= 0.57，p ＜ 0.001；MD-右侧图，3.B），这可能反映了右半球在视觉空间加工方面已知的偏侧化倾向 [53]。

对MD系统中个体脑区的活动进行后续分析显示，在Python实验中，20个功能性兴趣区中有17个，在ScratchJr实验中，20个功能性兴趣区中有14个，对代码问题的反应显著强于对句子问题的反应（详见附录B.1）。这表明，代码加工广泛分布于MD系统中，并未局限于其中某个特定的脑区子集。在这一激活的脑区子集中，我们评估了是否有任何功能性兴趣区相较于其他会激活MD系统的认知要求较高的任务，对代码问题表现出选择性 。ScratchJr实验中没有发现这样的脑区；而在额叶中，有三个脑区（中央前回-A、中央前回-B、中额叶）对Python代码问题的反应强于对困难的MD系统工作记忆定位任务。然而，这些脑区中的代码 > 句的效应量（$\Delta\beta$= 1.03，0。95、 0。97）与所有MD系统功能性兴趣区的平均效应量（平均 $\Delta\beta$= 1.03）相当，这表明较高的反应是由潜在的代码模拟 而非代码理解所引起的。我们得出结论：代码理解 得到MD系统的广泛支持，MD系统中没有任何特定脑区在功能上专门负责代码理解。

这些新结果进一步确立了MD系统在代码理解加工中的作用，而我们在本研究中对代码理解进行了严格且清晰的定义。

研究问题2结果。 是的，代码理解 会持续激活MD系统。与数学和逻辑不同，它会激活左、右两个大脑半球中的功能性兴趣区。事实上，MD系统内没有任何特定的功能性兴趣区专门负责代码理解；相反，代码理解广泛依赖于整个系统。

研究问题 3。代码阅读是否激活语言系统？**对比条件**。代码、句子、句子阅读、非词阅读

我们以类似的方式考察了语言系统对代码条件的反应。图3.A、B（最右侧的图）展示了语言系统对上述两个代码条件和两个语言定位条件的总体反应。正如对定位任务的预期，我们发现句子阅读的激活显著高于非词阅读 [9, 24]。在

代码条件中，我们发现，在 Python 和 ScratchJr 中，句子问题对语言系统的激活程度与句子定位任务相当。这同样是意料之中的，因为句子问题包含描述程序功能的英语句子（图2. A）。然而，在两项实验中，代码问题引发的反应都弱于句子问题引发的反应（Python：$\Delta\beta$= 0.98，$p < 0.001$；ScratchJr：$\Delta\beta$= 0.99，$p < 0.001$）。仅凭这一观察无法判断代码是否会激活语言系统，因此我们进一步将这些激活与定位基线中的非词阅读进行比较。非词阅读是语言系统活动的下限；这是语言系统未主动参与语言解释时所表现出的活动。在 Python 实验中，代码条件引发的反应仅强于非词阅读（$\Delta\beta$= 0.78，$p < 0.001$），而在 ScratchJr 实验中则并非如此（$\Delta\beta$= 0.15，p= 0.29），这表明代码并不能持续激活语言系统。

这项对语言系统进行的严谨研究所得结果，与 Siegmund 等人的结论相反；他们报告称，除其他脑区外，语言系统也参与其中。我们将在第 7 节进一步讨论这一点。

研究问题 3 的结果。不会，代码阅读并不会始终激活语言系统。

研究问题 4。有意义的变量名会影响语言系统对代码的反应吗？**对比条件**。codeE、codeJ

鉴于我们发现 Python 代码会激活语言系统，而 ScratchJr 不会，我们进一步研究这是否是代码中有意义的变量名所致。为考察这一影响，我们让参与者阅读一半变量名在英语中具有语义意义的 Python 代码问题（codeE），以及另一半变量名为日语、因而没有语义意义的 Python 代码问题（codeJ）；24 名参与者中有 18 名报告称不懂日语。在语言系统中，我们没有发现有意义的变量名的影响（$\Delta\beta$= 0.03，$p = 0.84$）（图3.D，非日语使用者）、日语知识的影响（$\Delta\beta$= 0.03，$p = 0.93$）（图3.D，日语使用者），也没有发现两者之间的交互作用（$\Delta\beta$= 0.09，$p = 0.71$），这表明语义上有意义的变量名并不会影响语言系统的反应。这一结果令人意外，因为已有研究表明，语言系统对词义非常敏感 [6]。一种可能的解释是，参与者并未充分利用词语的含义来解决问题。

RQ 4 结果。 有意义的变量名不会影响语言系统对代码的响应。

研究问题 5。除了MD系统和语言系统之外，是否还有对代码理解作出响应的脑区？

为了寻找MD系统和语言系统之外对代码理解 作出响应的脑区，我们进行了全脑组约束受试者特异性分析。对于Python和ScratchJr，我们都寻找在代码 >句子条件下出现激活的脑区。随后，我们考察这些脑区对代码问题和句子问题的响应（通过留出数据进行交叉验证），以及它们对两项定位实验中各条件的响应。在两项实验中，发现的脑区在空间分布上都与MD系统相似。对于Python，任何对代码作出响应的脑区也会对空间工作记忆任务（MD系统定位任务）作出响应。对于ScratchJr，一些功能性兴趣区对代码问题的响应强于对空间工作记忆任务的响应；这些功能性兴趣区位于早期视觉区／腹侧视觉通路，很可能是对ScratchJr代码的低级视觉特征（其中包含色彩鲜艳的图标、对象等。）作出响应。这些全脑分析表明，MD系统会稳定且一致地对代码理解作出响应，证实了研究问题1和3中讨论的基于功能性区域的分析结果。此外，这也表明基于功能性区域的分析并未遗漏MD系统和语言系统边界之外任何因代码理解而激活的非视觉脑区。

RQ 5 结果。 我们未在MD和语言系统之外发现代码选择性区域。

RQ 6。 专业水平是否会影响MD和语言系统对代码理解的响应?

我们通过将每个系统内的响应与独立获得的专业水平分数进行相关分析，研究专业水平的作用。专业水平分数来自实验1参与者（1小时的Python评估模块；附录A.1），以及实验2参与者在扫描仪内的准确率分数。图3.E绘制了百分比专业水平分数（横轴）与代码理解 （代码 > 句）之间的关系。没有任何相关性达到显著水平。然而，由于参与者数量相对较少（分别为N = 24和N = 19），应谨慎解释这些结果。

RQ 6 结果。 我们没有足够的数据来观察专业水平对MD和语言系统响应代码理解的影响。

7 讨论

我们提出了一组新的结果，加深了我们对程序理解认知基础的认识。我们发现，代码阅读 （我们将其定义为同时包括代码理解 和代码模拟）只会显著激活MD系统，而不会激活语言系统。尽管这两个系统在我们大脑的左半球中解剖位置相近，但通过功能定位任务，我们的研究清楚地区分了这两个系统各自的作用。

MD系统结果。 我们发现，MD系统始终会处理代码阅读。我们通过展示这些激活在两种代码属性——控制流和数据操作——之间具有普遍性来支持我们的观察，这表明该系统的反应

对代码的变化具有稳健性。我们进一步发现，MD系统对代码理解始终会产生反应。它还会对代码模拟 产生强烈反应（在Python中尤为明显），但我们仅在ScratchJr中发现了微弱证据，这一点有待未来研究进一步探讨。鉴于代码理解 和模拟 都需要注意力、工作记忆、抑制控制、规划以及一般性的灵活关系推理——这些认知过程长期以来都与MD系统相关联——因此，认为MD系统会处理代码阅读是合理的 [16, 17] [42] 。

这一发现也支持了Liu等人的最新结果。由于MD系统之外没有其他区域对代码产生反应，我们推测，除处理代码之外，该系统还存储代码特定知识。这些知识可能包括特定于某种编程语言的概念（例如Java与Python中标记数组的语法），以及不同语言之间共享的概念（循环、条件）。关于数学和物理学加工的研究证据 [4,26] 表明，MD系统能够在长期记忆中存储某些领域特定表征，这或许适用于那些在进化上出现较晚、在个体发展中习得较晚的知识领域。总而言之，我们发现，一个已知的大脑系统此前尚未被研究其在代码阅读 任务中的作用，但它确实参与了代码理解。**语言系统结果**。 我们在本研究中明确确立了一个重要结论：代码阅读 并非 始终由语言系统加工。这是一项新发现，补充了 Siegmund 等人 和 Floyd 等人的结果，同时也证实了 Liu 等人的结果。Siegmund 等人 通过展示左侧化脑活动的证据，报告了语言中枢参与代码理解 的情况。尽管尚不清楚他们的观察在技术上究竟来自语言系统还是 MD系统，但我们推测，他们观察到的是代码模拟 ，而非代码理解。此外，令人意外的是，我们发现语言系统对是否存在有意义的变量名并不敏感。还需要开展更多研究，以确定语言系统为何会对 Python 代码表现出一定活动。

尽管代码与自然语言之间存在诸多相似之处，语言系统对代码理解 的反应并不一致。然而，语言系统在代码理解 过程中缺乏一致性参与，并不意味着语言和代码加工背后的机制完全不同。MD系统和语言系统可能都具有组织方式相似的神经回路，使它们能够将符号映射到概念。然而，我们观察到与代码相关的活动主要出现在 MD系统中，这一事实表明代码理解 并不会激活与语言相同的神经回路，而需要使用领域通用的 MD系统神经回路。

在确定阅读程序时被激活的区域后，我们讨论我们的结果将如何影响编程语言（PL）、软件工程（SE）和计算机科学教育（CS-Ed）社群。**对PL、SE和CS-Ed社群的影响**。为了理解我们的结果如何具体应用于改进我们理解程序的方式，我们首先确定涉及同一大脑系统的两种认知活动之间的关系（在我们的案例中，即工作记忆任务和代码理解涉及MD系统）。一些研究声称，对于任何共享相同脑资源的两种认知活动，训练其中一种活动都会促进另一种活动的提升 [38, 44]。例如，如果语言和音乐共享并激活同一个大脑系统，那么用于参与和训练其中一种活动的工具与方法应当能够迁移到另一种活动，并促使另一种活动得到提升。由于训练和改善一个人的MD系统所产生的影响尚未得到充分理解，因此尚不清楚，针对认知要求较高的非编码任务进行训练，是否能够提升我们阅读和理解程序的能力。

鉴于 Siegmund 等人、Portnoff 等人 [50] 以及其他类似研究得出了相反的结论，一些人建议在教授编程时采用“语言优先”的方法。我们的研究证据并不支持这一主张，我们也提醒不要仅依据近期神经影像学研究的结论，就将用于教授自然语言的实践应用于编程教学。

语言系统未参与代码理解，不应削弱语言在理解程序方面的作用。研究表明，使用命名不佳的变量会增加认知负荷 [19], , 而且非英语母语者在学习基于英语的编程语言时往往会遇到困难 [30]。未来的工作应调和这一差异，并努力说明认知研究的结果如何帮助我们理解程序。

代码机器学习模型。近期，使用大规模程序语料库训练的机器学习（ML）模型取得了显著进展，展现出执行函数重命名、错误检测等任务 [2]的能力。深度网络学习程序的“语言模型”，并且很可能学会以一种通用方式表示这些程序。这些由模型学习到的表示，是否对应于我们研究中不同功能性兴趣区中的表示（激活数据）？ 这种对应关系可能带来深远影响。例如，如果我们能够探查并分离出在 seq2seq、code2seq 或 GPT-3 等循环模型中编码循环和递归的特定权重或层，那么表示之间的对应关系或许能帮助我们定位大脑中循环和递归的编码位置。近期的研究已经在用于图像处理和识别的视觉皮层表示与深度卷积网络学习到的表示之间建立了这种对应关系 [13, 40, 59]。这为未来的研究提供了一个极具吸引力的方向。

8 效度威胁

我们在本文中报告的结果存在一些局限性。一个可能的效度威胁来自我们设计的任务——我们的程序任务是否测量了代码理解和领会能力？本研究所展示的程序都是较短的过程式代码片段，所涉及的程序属性有限。它们不具备生产级编程项目中通常会出现的复杂控制依赖和数据依赖。程序中的函数调用、对象、类型、复杂数据和状态变化等属性也未纳入研究，未来应在本研究对较简单代码片段的理解基础上进一步探究。此外，我们研究的是一种与编程相关的特定活动——阅读程序，并未考察其他同样重要的方面，例如设计解决方案、选择合适的数据结构以及编写程序。

设计代码刺激材料时，使用过于直白且信息量丰富的变量名，可能会导致参与者不逐行阅读呈现给他们的

代码，而是仅凭变量名猜测代码的作用。为避免这种情况，我们对变量名作出限制：它们应当信息丰富且自然（如同真实代码库中会出现的变量名），但又不能完全揭示整个代码片段的意图。然而，实际编程场景中并不存在这样的限制。刺激材料之间的差异可能造成混杂因素。如果某些条件中的代码明显比其他条件更长，那么我们观察到的脑激活趋势究竟由条件所致，还是由代码长度等因素所致，就难以判断。为避免这种情况，我们确保 $3 \times 3 \times 2$ 条件下的刺激材料具有相近的代码长度和总体反应时间。不过，它们并非完全相同。

专业水平也是一个潜在的混杂因素，可能会影响我们所见结果的可推广性。我们的大多数参与者来自大学环境，具有大约3至6年的编程经验。尽管在本研究中，参与者的经验水平总体上较为同质，但专业水平可能会与程序理解相关的脑功能发生交互作用，正如它会与其他认知功能发生交互作用一样 [1, 29, 32]。要考察专业水平的作用，就需要在具有不同熟练程度的人群中开展这些实验。

神经影像学实验通常会因样本量较小而面临结果缺乏普适性的风险。在我们的工作中，相对较小的样本量（实验1为N=24，实验2为N=19）仅影响我们的组分析（比较参与者之间的汇总信息）。尽管这些样本量在神经影像学界属于常见规模，但参与者数量较少限制了我们结果的稳健性。我们认为，类似神经影像学研究的作者可以借此机会开展合作，分析这些研究中的数据，这也将有助于分摊开展此类实验的高昂成本。

9 结论

我们的研究提出了一套新的方法、实验和结果，进一步加深了我们对大脑如何理解计算机代码的认识。我们的研究独特之处在于解决了两个导致先前研究结果难以解释的核心问题。首先，我们将对代码理解 这一行为的反应与对代码模拟的反应区分开来。其次，我们分析了在各个参与者中具有功能定位的脑区中的反应，从而能够准确解释所观察到的反应。我们发现，大脑中的MD系统会持续处理代码理解，而语言系统则不会。我们公开了代码、刺激材料和脑数据，供研究群体重复使用并进一步拓展。

10 致谢

本研究部分由美国国家科学基金会（NSF）资助项目1744809支持。SS和UMO'R获得了麻省理工学院-IBM沃森人工智能实验室的资助。

参考文献

[1] Aakash Agrawal、KVS Hari和SP Arun。2018年。阅读专业水平如何影响大脑中的字母表征？一项功能性磁共振成像研究。《Journal of Vision》18，10（2018），1161–1161。[2] Miltiadis Allamanis、Earl T Barr、Premkumar Devanbu和Charles Sutton。2018年。关于大规模代码和自然性的机器学习综述。《ACM Computing Surveys》（CSUR）51，4（2018），1–37。[3] Marie Amalric和Stanislas Dehaene。2016年。专家数学家高等数学大脑网络的起源。《美国国家科学院院刊》113，18（2016），4909–4917。[4] Marie Amalric和Stanislas Dehaene。2019年。人脑中负责数学知识的独特皮层网络。《NeuroImage》189（2019），19–31。[5] Katrin Amunts和Karl Zilles。2012年。布罗卡区的结构与组织原则。《Trends in cognitive sciences》16，8（2012），418–426。[6] Andrew James Anderson、Edmund C Lalor、Feng Lin、Jeffrey R Binder、Leonardo Fernandino、Colin J Humphries、Lisa L Conant、Rajeev DS Raizada、Scott Grimm和Xixi Wang。2019年。皮层网络中的多个脑区通常会编码所读句子中多个语法位置的词义。《Cerebral cortex》29，6（2019），2396–2411。[7] Marina Umaschi Bers。2018年。幼儿教育中的编程、游戏场和读写能力：KIBO机器人和ScratchJr的开发。载于《2018 IEEE全球工程教育会议》（EDUCON）。IEEE，2094–2102。[8] Marina U Bers、Carina González-González和Ma Belén Armas-Torres。2019年。将编程作为游戏场：促进儿童课堂中的积极学习体验。《Computers & Education》138（2019），130–145。[9] Idan Blank、Nancy Kanwisher和Evelina Fedorenko。2014年。BOLD血氧水平依赖信号波动模式揭示了语言系统与多重需求系统之间的功能性分离。《Journal of neurophysiology》112，5（2014），1105–1118。[10] Idan A Blank和Evelina Fedorenko。2017年。领域一般性脑区对语言输入的跟踪并不如语言选择性脑区紧密。《Journal of Neuroscience》37，41（2017），9999–10011。[11] Matthew Brett、Ingrid S Johnsrude和Adrian M Owen。2002年。人脑功能定位问题。《Nature reviews neuroscience》3，3（2002），243–249。[12] Korbinian Brodmann。1909年。《大脑皮层比较定位学：基于细胞结构阐述其原则》。Barth。[13] Santiago A Cadena、George H Denfield、Edgar Y Walker、Leon A Gatys、Andreas S Tolias、Matthias Bethge和Alexander S Ecker。2019年。深度卷积模型改进了对猕猴初级视觉皮层对自然图像反应的预测。《〈PLoS计算生物学〉》15，4（2019），e1006897。[14] Joao Castelhano、Isabel C Duarte、Carlos Ferreira、Joao Duraes、Henrique Madeira和Miguel Castelo-Branco。2019年。岛叶在计算机代码直觉式专家级错误检测中的作用：一项功能性磁共振成像研究。《Brain imaging and behavior》13，3（2019），623–637。[15] David Glenn Clark和Jeffrey L Cummings。2003年。失语症。载于《Neurological Disorders》。Elsevier，265–275。[16] John Duncan。2010年。灵长类动物大脑的多重需求（MD）系统：用于智能行为的心理程序。《Trends in cognitive sciences》14，4（2010），172–179。[17] John Duncan和Adrian M Owen。2000年。多种认知需求共同招募的人类额叶脑区。《Trends in neurosciences》23，10（2000），475–483。[18] Sarah Fakhoury、Yuzhan Ma、Venera Arnaoudova和Olusola Adesope。2018年。糟糕的源代码词汇和可读性对开发者认知负荷的影响。载于《2018 IEEE/ACM第26届程序理解国际会议》（ICPC）。IEEE，286–28610。[19] Sarah Fakhoury、Devjeet Roy、Yuzhan Ma、Venera Arnaoudova和Olusola Adesope。2019年。衡量词汇和结构不一致性在错误定位期间对开发者认知负荷的影响。《Empirical Software Engineering》（2019），1–39。[20] Evelina Fedorenko、Michael K Behr和Nancy Kanwisher。2011年。人脑高级语言处理的功能特异性。《美国国家科学院院刊》108，39（2011），16428–16433。[21] Evelina Fedorenko和Idan A Blank。2020年。布罗卡区不是一个自然类。《Trends in cognitive sciences》24，4（2020），270–284。[22] Evelina Fedorenko、John Duncan和Nancy Kanwisher。2012年。语言选择性脑区和领域一般性脑区在布罗卡区内并列分布。《Current Biology》22，21（2012），2059–2062。[23] Evelina Fedorenko、John Duncan和Nancy Kanwisher。2013年。额叶和顶叶皮层局部脑区具有广泛的领域一般性。《美国国家科学院院刊》110，41（2013），16616–16621。[24] Evelina Fedorenko、Po-Jang Hsieh、Alfonso Nieto-Castañón、Susan Whitfield-Gabrieli和Nancy Kanwisher。2010年。语言功能性磁共振成像研究的新方法：在个体受试者中对兴趣区进行功能定义。《Journal of neurophysiology》104，2（2010），1177–1194。[25] Evelina Fedorenko、Anna Ivanova、Riva Dhamala和Marina Umaschi Bers。2019年。编程的语言：认知视角。《Trends in cognitive

528–[26] Jason Fischer、–525。John G Mikhael、Joshua B Tenenbaum 和 Nancy Kanwisher。2016。直觉物理推理的功能神经解剖学。《美国国家科学院院刊》113，34（2016），E5072–E5081。[27] Benjamin Floyd、Tyler Santander 和 Westley Weimer。2017。解码大脑中的代码表征：关于代码审查和专业水平的功能性磁共振成像研究。载于2017年IEEE/ACM第39届国际软件工程大会（ICSE）。IEEE，175–186。[28] Gary H Glover。2011。功能性磁共振成像概述。《神经外科诊所》22，2（2011），133–139。[29] Jesse Gomez、Michael Barnett 和 Kalanit Grill-Spector。2019。儿童时期对 Pokémon 的长期接触表明，偏心率驱动着视觉皮层的组织。《自然：人类行为》3，6（2019），611–624。[30] Philip J Guo。2018。非英语母语者学习计算机编程：障碍、愿望和设计机会。载于2018年CHI会议计算系统中的人因论文集。1–14。[31] Victor W Henderson。1986。Paul Broca 对失语症研究较少为人所知的贡献：历史视角与当代意义。《神经病学档案》43，6（1986），609–612。[32] Klaus Hoenig、Cornelia Müller、Bärbel Herrnberger、Eun-Jin Sim、Manfred Spitzer、Günter Ehret 和 Markus Kiefer。2011。专业音乐家对乐器语义表征的神经可塑性。《NeuroImage》56，3（2011），1714–1725。[33] Yu Huang、Kevin Leach、Zohreh Sharafi、Nicholas McKay、Tyler Santander 和 Westley Weimer。2020。使用医学成像和眼动追踪进行代码审查时的偏差与差异：性别、人类与机器。载于第28届ACM欧洲软件工程大会与软件工程基础研讨会联合会议论文集。456–468。[34] Yu Huang、Xinyu Liu、Ryan Krueger、Tyler Santander、Xiaosu Hu、Kevin Leach 和 Westley Weimer。2019。使用功能性磁共振成像和功能性近红外光谱提炼数据结构操作的神经表征。载于2019年IEEE/ACM第41届国际软件工程大会（ICSE）。IEEE，396–407。[35] Yoshiharu Ikutani、Takatomi Kubo、Satoshi Nishida、Hideaki Hata、Kenichi Matsumoto、Kazushi Ikeda 和 Shinji Nishimoto。2020。专业程序员拥有经过精细调节的源代码皮层表征。《eNeuro》（2020）。[36] Yoshiharu Ikutani 和 Hidetake Uwano。2014。使用近红外光谱测量程序理解过程中的脑活动。载于第15届IEEE/ACIS软件工程、人工智能、网络与并行/分布式计算国际会议（SNPD）。IEEE，1–6。[37] Anna A Ivanova、Shashank Srikant、Yotaro Sueoka、Hope H Kean、Riva Dhamala、Una-May O'Reilly、Marina U Bers 和 Evelina Fedorenko。2020。计算机代码的理解主要依赖领域一般性执行脑区。《eLife》9（2020），e58906。[38] Susanne M Jaeggi、Martin Buschkuehl、John Jonides 和 Walter J Perrig。2008。通过工作记忆训练提高流体智力。《美国国家科学院院刊》105，19（2008），6829–6833。[39] Nancy Kanwisher。2010。人脑中的功能特异性：窥见心智的功能架构。《美国国家科学院院刊》107，25（2010），11163–11170。[40] Seyed-Mahdi Khaligh-Razavi 和 Nikolaus Kriegeskorte。2014。深度监督模型而非无监督模型或可解释IT皮层表征。《PLoS计算生物学》10，11（2014），e1003915。[41] Ryan Krueger、Yu Huang、Xinyu Liu、Tyler Santander、Westley Weimer 和 Kevin Leach。2020。神经学鸿沟：散文写作与代码写作的功能性磁共振成像研究。载于2020年IEEE/ACM第42届国际软件工程大会（ICSE）。IEEE，678–690。[42] Yun-Fei Liu、Judy Kim、Colin Wilson 和 Marina Bedny。2020。计算机代码理解在额顶网络中与形式逻辑推理共享神经资源。《eLife》9（2020），e59340。[43] Mara Mather、John T Cacioppo 和 Nancy Kanwisher。2013。功能性磁共振成像如何为认知理论提供信息。《心理科学展望》8，1（2013），108–113。[44] Monica Melby-Lervåg 和 Charles Hulme。2013。工作记忆训练有效吗？一项荟萃分析综述。《发展心理学》49，2（2013），270。[45] Zachary Mineroff、Idan Asher Blank、Kyle Mahowald 和 Evelina Fedorenko。2018。语言网络、多重需求网络和默认模式网络之间稳健的分离：来自效应量区域间相关性的证据。《Neuropsychologia》119（2018），501–511。[46] Takao Nakagawa、Yasutaka Kamei、Hidetake Uwano、Akito Monden、Kenichi Matsumoto 和 Daniel M German。2014。基于脑血流测量量化程序员在程序理解过程中的心理负荷：一项对照实验。载于第36届国际软件工程大会配套论文集。448–451。[47] Alfonso Nieto-Castañón 和 Evelina Fedorenko。2012。被试特异性功能定位器提高多被试分析的敏感性和功能分辨率。《Neuroimage》63，3（2012），1646–1669。[48] Chris Parnin、Janet Siegmund 和 Norman Peitek。2017。论程序员专业水平的本质。载于Ppig。16。

[49] Russell A Poldrack。2011。从神经影像数据推断心理状态：从逆向推理到大规模解码。*Neuron* 72，5（2011），692–697。[50] Scott R Portnoff。2018。入门计算机编程课程首先且最重要的是一门语言课程。*ACM Inroads* 9，2（2018），34–52。[51] Ivonne Schröter、Jacob Krüger、Janet Siegmund 和 Thomas Leich。2017。理解有关程序理解的研究。载于*2017*年*IEEE/ACM*第*25*届国际程序理解大会（*ICPC*）。IEEE，308–311。[52] Cory Shain、Idan Blank、Marten van Schijndel、Evelina Fedorenko 和 William Schuler。2019。功能性磁共振成像揭示自然语境句子理解过程中的语言特异性预测编码。*Neuropsychologia*（2019）。[53] Summer L Sheremata、Katherine C Bettencourt 和 David C Somers。2010。随着视觉短时记忆负荷增加，视网膜拓扑后顶叶皮层呈现半球不对称性。*Journalof Neuroscience* 30，38（2010），12581–12588。[54] Janet Siegmund、Christian Kästner、Sven Apel、Chris Parnin、Anja Bethmann、Thomas Leich、Gunter Saake 和 André Brechmann。2014。使用功能性磁共振成像理解源代码理解。载于第*36*届国际软件工程大会论文集。378–389。

[55] Janet Siegmund、Norman Peitek、Chris Parnin、Sven Apel、Johannes Hofmeister、Christian Kästner、Andrew Begel、Anja Bethmann 和 André Brechmann。2017年。测量程序理解的神经效率。载于*2017*年第*11*届软件工程基础联合会议论文集。140–150。[56] Elliot Soloway。1986年。学习编程= 学习构建机制和解释。*Commun. ACM* 29，9（1986），850–858。[57] Nathalie Tzourio-Mazoyer、Brigitte Landeau、Dimitri Papathanassiou、Fabrice Crivello、Olivier Etard、Nicolas Delcroix、Bernard Mazoyer 和 Marc Joliot。2002年。使用MNI磁共振成像单被试大脑的宏观解剖分区，对SPM中的激活进行自动解剖标注。*Neuroimage* 15，1（2002），273–289。[58] Xin Xia、Lingfeng Bao、David Lo、Zhenchang Xing、Ahmed E Hassan 和 Shan- ping Li。2017年。测量程序理解：一项针对专业人士的大规模实地研究。*IEEE*软件工程汇刊 44，10（2017），951–976。[59] Daniel LK Yamins、Ha Hong、Charles F Cadieu、Ethan A Solomon、Darren Seibert 和 James J DiCarlo。2014年。性能优化的层次模型预测高级视觉皮层中的神经反应。美国国家科学院院刊 111，23（2014），8619–8624。

A 方法

本节将详细介绍我们用于功能性脑区定位任务的刺激材料、我们如何定义功能性兴趣区、为编码任务设计的刺激材料、向研究参与者呈现这些任务时所遵循的流程、参与者的招募方式，以及最后对记录的MRI数据进行的分析。

A.1 参与者

在实验1中，我们招募了25名参与者（13名女性，平均年龄为 = 22 岁，标准差为 = 3.0）。参与者开始编程时的平均年龄为16岁（标准差为 = 2.6）；平均编程年限为6.3年（标准差为 = 3.8）。除Python外，20人还表示了解一些Java，18人表示了解C/C++，4人表示了解函数式语言，20人表示了解Matlab和R等数值语言。23名参与者为右利手，1名为左右利手，1名为左利手；这名左利手参与者的语言系统位于右半球，因此未纳入分析，最终共纳入24名参与者（其余参与者的语言区域均位于左半球）。实验1的所有参与者还在预定扫描日期所在的一周内完成了为时1小时的纸笔式Python测验。测验包括23道题，其中22道要求简短作答。另有一道题要求设计一种算法并用Python实现。测验题目与实验中呈现的刺激材料没有重叠。我们用该测验衡量参与者的Python和编程总体熟练度。

对于实验2，我们招募了21名参与者（13名女性，平均年龄为 = 22 岁，标准差为 = 2.8）。两项实验的参与者没有重叠。除ScratchJr外，8人还报告具备一些Python知识，6人报告具备Java知识，9人报告具备C/C++知识，1人报告具备函数式语言知识，14人报告了解Matlab和R等数值语言（1名参与者未完成编程问卷）。20人惯用右手，1人为双利手；通过语言定位任务评估发现，所有参与者的语言区域均位于右侧。实验2中的2名参与者因磁共振扫描期间运动程度过高而被排除，最终共计19名参与者。

所有参与者均从当地大学及其周边社区招募，并因参与研究获得补偿。所有参与者均为英语母语者，视力正常或经矫正后正常，并分别报告具备Python或ScratchJr的工作知识。该研究方案经机构审查委员会批准。所有参与者均按照方案要求签署了书面知情同意。

A.2 定位任务

所有参与者均完成了语言定位任务，旨在识别对语言有反应的大脑脑区 [24], 空间工作记忆定位任务，旨在识别多重需求脑区 [23], 以及一组编码任务。

语言定位任务识别出每位参与者大脑中会选择性响应语言刺激的脑区。在该任务中，参与者阅读句子（例如 NOBODY COULD HAVE PRE- DICTEDTHEEARTHQUAKEINTHISPART OFTHECOUNTRY）以及由彼此不相关但可发音的非词组成的列表（例如 U BIZBY ACWORRILY MIDARALBUSHU SNOOKI BILIBOP KUKEE WEPS WIBRONPUZ）。每个刺激材料总共包含十二个词/非词。关于语言材料构建方式的详细信息，请参见 Fedorenko 等 [24]。我们将句子阅读任务称为 SR，将非词阅读任务称为 NR。此前研究表明，SR > NR 对比能够可靠地激活左侧化的额颞叶语言处理脑区，并且对材料、任务和呈现模态的变化具有稳健性 [24, 45]。刺激材料呈现在屏幕中央，每次呈现一个词 / 非词，呈现速率为每个词/非词 450 毫秒。每个刺激材料之前会显示 100 毫秒的空白屏幕，之后显示一张手指按下按钮的图片，持续 400 毫秒，随后再显示 100 毫秒的空白屏幕，因此每个试次时长总计为 6 秒。参与者被要求在看到手指按下按钮的图片时按下按钮。纳入该任务是为了帮助参与者保持警觉和清醒。

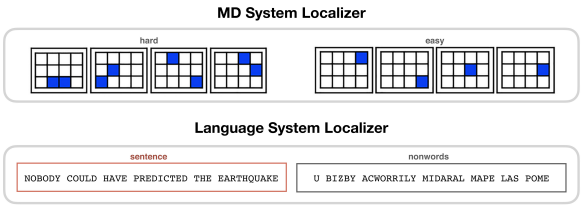

图4：我们采用了先前研究中的两个定位任务 [23, 24]。MD系统定位任务测试空间推理能力。参与者会快速连续地看到4个屏幕，每个屏幕上网格中的不同方格被高亮显示。随后，他们会看到两个网格，并需要正确识别出能够叠加呈现此前所有高亮方格的网格。语言系统定位任务要求参与者阅读两组句子：一组连贯且有意义（左侧），另一组虽可发音但毫无意义（右侧）。

先前的研究建立了一项空间工作记忆任务，用于识别个体中的MD系统 [23]。参与者需要记住在 3×4 网格中按顺序呈现的4个位置（简单条件）或8个位置（困难条件）（见图2）。在两种条件下，他们都要在每次试次结束时完成二择一迫选任务，以指出刚刚看到的位置集合。此前研究表明，这种困难 > 简单对比能够稳定激活MD脑区 [23]。研究还表明，在许多不同类型的任务中，这些脑区都会对难度操控产生反应[16, 23]。刺激材料在屏幕中央呈现，共分为四个步骤。每个步骤持续1000毫秒；在简单条件下，每一步在网格上呈现一个位置，在困难条件下则呈现两个位置。每个刺激材料之后都会进入选择步骤，屏幕上并排显示两个网格。其中一个网格包含前四个步骤中呈现的位置，另一个网格则包含错误的位置集合。参与者需要按下两个按钮中的一个，以选择显示正确位置的网格。

A.3 编码任务

实验1（Python）中的编码任务包括三个条件：使用英文变量的Python问题（codeE）、使用

日语变量的Python问题（codeJ），以及用英语句子描述的问题（sent）。完整的问题列表可在项目代码仓库中找到。每位参与者都会看到72个问题，三个条件各24个。每位参与者针对同一个问题只会看到一种版本。每个条件中，一半问题要求执行数学运算，另一半要求进行字符串操作。此外，数学和字符串操作问题在程序结构上也有所不同：每种问题中，13 的问题仅包含顺序执行语句，13 包含一个for循环，13 包含一个if/else语句。参与者被要求阅读问题描述，并在准备好作答时按下按钮（最短阅读时间限制为5秒，最长为50秒；平均阅读时间为19秒）。按下按钮后，他们会看到四个回答选项，并须按下相应按钮指示自己的回答。回答界面显示5秒。

实验2（ScratchJr）包含两个条件：ScratchJr中的简短程序，以及以书面句子呈现的问题文字描述。在每个试次中，参与者首先会看到注视十字4秒，随后呈现程序8秒（以代码或文字形式呈现），接着播放一个视频，该视频的内容可能与程序的输出相符，也可能不符。参与者需要在视频播放期间按下两个按钮中的一个，以指示视频是否与描述相符。视频平均时长为4.13秒（标准差为1.70秒）。

A.4 实验流程

每项任务的组织方式如下：区块包含来自某一条件的多个刺激材料（在神经影像学文献中称为试次）。不同条件的区块排列在一起，构成一个运行。语言系统和MD系统定位任务包含两个区块：一个注视区块，以及包含定位条件的实验区块。实验区块持续18秒（每个区块包含3个试次），注视区块持续14秒。每次运行（由5个注视区块和16个实验区块组成）持续358秒。每位参与者完成2次运行。

对于 MD 定位任务，实验区块持续 32 秒（每个区块包含 4 个试次），注视区块持续 16 秒。每次运行（由 4 个注视区块和 12 个实验区块组成）持续 448 秒。每位参与者完成 2 次运行。

各运行中的条件顺序采用随机平衡。在实验1（Python）的每次运行中，参与者完成了三种条件（codeE、codeJ、句子）各 2 个试次。每次运行包括 3 个注视区块，每个区块持续 10 秒。每次运行平均持续 176 秒（标准差为 $=34$ 秒）。每位参与者完成 12 次运行。

对于实验2（ScratchJr），每次运行包括 6 个试次（每种条件 3 个），并在运行开始和结束时各注视 10 秒。每次运行平均持续 184.75 秒（标准差为 3.86 秒）。每位参与者完成 4 次运行。

A.5 定义功能性兴趣区

功能感兴趣区（fROIs）采用群组约束的个体特异性（GSS）方法定义： [24] 将一组空间分区与每位受试者的局部定位任务激活图结合起来，以约束个体功能性兴趣区的定义。这些分区标示了根据先前研究，特定对比下激活的大致位置；其范围足够大，能够涵盖个体激活位置的变异范围。

我们重新使用了先前研究中确定的分区 [23, 24]。对于语言定位器，我们使用了由220名参与者的句子阅读（SR） > 非词阅读（NR）对比的群体水平概率激活重叠图所衍生的六个分区。这些分区包括左下额回中的两个区域（LIFG、LIFGorb）、左中额回中的一个区域（LMFG）、左颞叶中的两个区域（LAntTemp和LPostTemp），以及一个延伸至角回的区域（LAngG）。

对于MD定位器，我们使用了由197名参与者的困难 > 简单空间任务对比的群体水平概率激活重叠图所衍生的20个分区（每个大脑半球10个）。这些分区包括额叶和顶叶中的区域，以及一个位于前扣带的区域。

在每个分区内，我们根据SR > NR对比的t值，选取响应最强的前10%体素。以这种方式定义的个体水平功能性兴趣区随后用于后续分析，以考察MD和语言系统在编码任务期间的行为。

A.6 数据处理与分析

使用 SPMS 参与者的数据进行分析。使用 MRI 的数据进行了运动校正，随后将其标准化到共同的大脑空间（蒙特利尔神经学研究所（MNI）模板），并重采样为 2mm 各向同性体素。随后，使用 4mm FWHM高斯滤波器对数据进行平滑处理，并进行高通滤波（截止时间为 200s ）。使用一般线性模型（ GLM ）估计效应，其中每个实验条件均采用与规范血流动力学反应函数（HRF）卷积的箱形函数进行建模（对整个区块进行建模）。

B 补充结果

我们提供了对 MD 和语言系统进行数据分析所得的补充结果。

B.1 功能性兴趣区响应——MD系统

对MD系统内个体脑区活动的分析显示，在Python实验中，20个功能性兴趣区中有17个对代码问题的反应显著强于对句子问题的反应；在ScratchJr实验中，20个功能性兴趣区中有14个呈现了同样的反应（图5）。这表明，代码加工广泛分布于MD系统之中，而不是局限于其中某个特定的脑区子集。

我们通过测量代码问题对MD系统的高难度工作记忆定位任务的反应，评估其对选择性 代码理解 的程度。图6绘制了MD系统各个区域的激活情况。我们在ScratchJr中未发现任何区域，但在额叶的三个区域（中央前回-A、中央前回-B、中额叶）发现这些区域对代码问题表现出更强的反应。然而，这些区域中代码 > 句子的幅度（$\Delta\beta = 1.03$，0.95，0.97）与所有MD系统功能性兴趣区的平均幅度（平均值 $\Delta\beta = 1.03$）相当，这表明较高的反应是由底层的代码模拟 而非代码理解引起的。

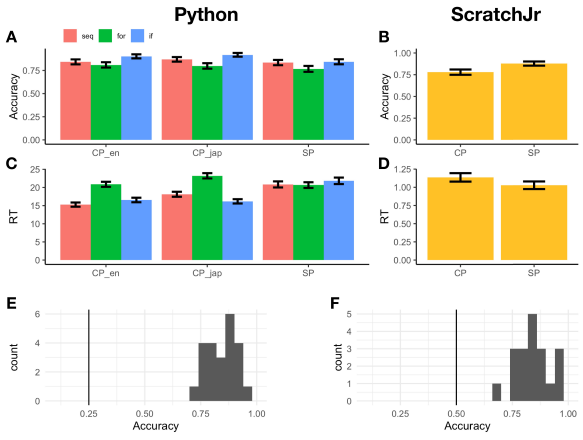

图7：对代码相关刺激材料的反应准确率和反应时

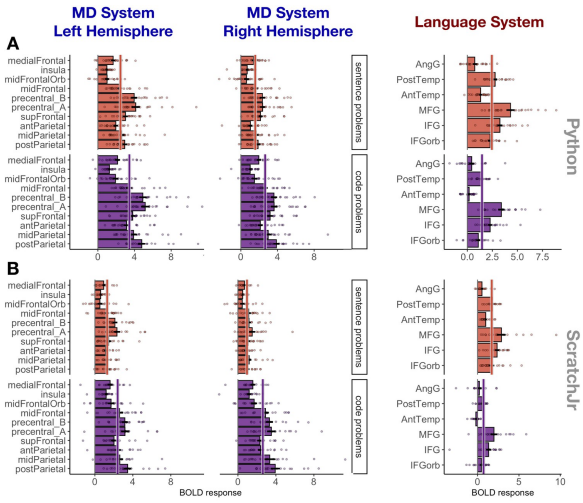

图5：MD系统左半球、MD系统右半球和语言系统中，功能性兴趣区对代码和句子条件的激活。

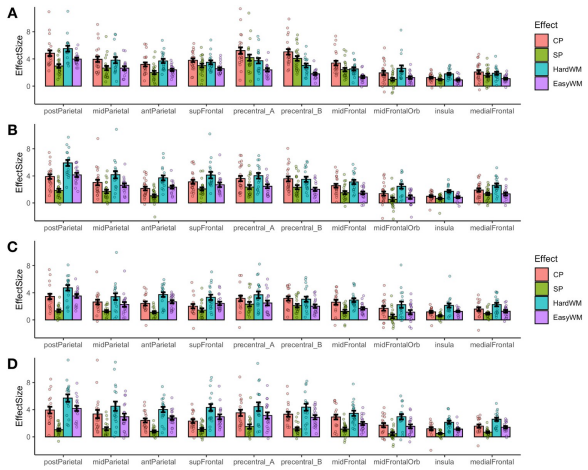

图6：MD系统功能性兴趣区对代码和句子条件的激活，并与简单和困难的MD系统定位任务进行对比。（A）实验1，Python；左半球功能性兴趣区；（B）实验1，Python；右半球功能性兴趣区；（C）实验2，ScratchJr；左半球功能性兴趣区；（D）实验2，ScratchJr；右半球功能性兴趣区。

B.2 行为结果

图7展示了在成像扫描仪中向我们的参与者呈现的与代码相关的刺激材料的反应准确率和反应时。实验1（Python）中的参与者反应率为99.6%，代码问题的平均准确率为85%（图7.A）。图7.E显示了Python参与者反应准确率的直方图。

实验2（ScratchJr）的参与者响应率为98.6%，代码问题的平均准确率为79%（图7.B）。这些结果表明，参与者熟练掌握了相关编程语言，并积极参与了任务。图7.F显示了ScratchJr参与者反应准确率的直方图。