arXiv:2302.00178v1 [cs.CV] 1 Feb 2023

从多样化视频演示生成程序

Anthony Manchin[1 2 3]
a
nthony.
manchin@adelaide.edu.auJamie
Sherrah[1 3]
jamie.sherrah@adelaide.edu.auQi Wu[1 3]
qi.wu01@adelaide.edu.auAnton van den
Hengel[1 3]
anton.vandenhengel@adelaide.edu.au

[1]阿德莱德大学，澳大利亚[2]Apollo Labs，阿德莱德，澳大利亚[3]澳大利亚机器学习研究所，阿德莱德

摘要

运用归纳推理从多次观测中提取一般规则的能力，是衡量智力的重要指标。人类运用这种能力，不仅能理解周围的世界，还能预测所经历的各种交互会产生怎样的结果。长期以来，机器一直难以从多次观测中归纳出一般规则，尤其是在需要计算机视觉的情况下。本文提出了一种模型，能够通过同时执行摘要和翻译，从视频演示中提取一般规则。与以往的研究不同，我们将问题定义为多序列到序列任务，并由模型学习摘要。这样一来，我们的模型便能利用边缘情况，而这些情况通常会被传统摘要技术弱化或舍弃。此外，我们还表明，该方法无需额外的过滤方法即可处理含噪规格。我们在 Vizdoom 环境中通过根据视频演示合成程序来评估模型，并取得了当前最先进水平的结果：与以往研究相比，程序准确率相对提高了 11.75%。

1 引言

归纳推理是指运用逻辑从多种观测中提取一般规则，这项技能被广泛视为智力的体现。众所周知，人类（以及许多动物）能够观察演示，随后运用归纳推理获取新的知识和技能，而无需明确指示。这种行为的一个例子是儿童学习玩电子游戏：他们能够观察其他玩家，并学会游戏规则，而不必亲自玩游戏。

此外，儿童还能抽象并概括从一款电子游戏中归纳出的信息，并将同一套逻辑规则集应用于完全不同的

© 2022年。本文档的版权归其作者所有。可免费以印刷或电子形式原样传播。

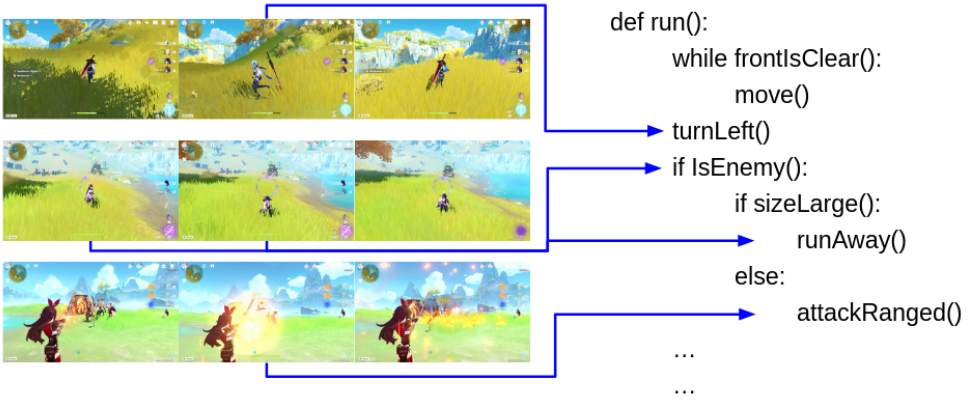

图1：以游戏《原神》为例，展示视觉程序合成任务。人类仅通过观察游戏玩法示例，就能推断出一套通用规则。视觉程序合成任务旨在训练一台机器，使其仅通过观看智能体的视觉演示，就能合成一个准确捕捉规则集的程序。

虚拟领域。例如，有人观察到，在某款游戏中，红十字符号代表健康相关奖励，随后便利用这一信息推断，另一款游戏中类似的符号也与玩家生命值有关。然而，要从复杂的观测中归纳出一般规则，对机器而言一直是一项艰巨的任务。事实上，直到最近，关于如何从多种视觉观测中提取一般规则的研究仍然寥寥无几。不过，计算机视觉和机器学习技术的进步正在改变这一局面。

我们旨在创建一种模型，使其能够通过从多个视觉演示中推断规格来生成可执行代码。创建能够生成可执行代码的模型，长期以来一直是人工智能研究人员的梦想 [12, 27]。然而，直到最近，研究才主要聚焦于在给定所需规格 [2, 13]的情况下生成代码。然而，当模型还需要自行推断规格时，问题就变得困难得多。这项任务最初由 [24], 提出，带来了独特的挑战：模型需要准确检测演示中的相关语义，理解各演示之间的关系，定义一套能够捕捉这些信息的规格，最后生成满足这些规格的程序。

以往的方法 [10, 24] 将其视为序列到序列任务，并采用曾一度流行的长短期记忆单元循环神经网络架构作为模型骨干。然而，这些方法受限于循环模型处理长序列的能力。这一限制导致两者都 [10, 24] 使用了摘要后的潜在空间表示，这无疑限制了模型完整捕捉所有规格约束的能力。与此同时， [5] 试图使用基于规则的求解器生成所需的代码，其规格由神经网络模型提供。由于基于规则的求解器极易受到噪声影响， [5]提出使用动态过滤方法对规格进行去噪。

为解决这些局限，我们提出“视频到文本迁移Transformer”（VT4）。我们利用基于注意力的Transformer网络处理长而不连续序列的能力，无需再对特征进行摘要。具体而言，我们构建了一种“视觉语言”，用于封装每个演示中的相关语义信息。随后，我们将所有演示（以视觉语言表示）同时输入编码器-解码器 Transformer 网络，该网络学习对演示进行摘要，并将其转换为可执行程序。此外，我们还表明，我们的方法无需额外的过滤方法即可处理含噪规范。我们在一个部分可观测虚拟环境（Vizdoom）中评估了模型 [16]并证明，我们的方法优于以往的摘要和基于规则的方法，取得了当前最先进水平的结果。与摘要方法和基于规则的方法相比，我们分别观察到 15.45% 和 11.75% 的相对提升。

贡献

- 我们提出了视频到文本迁移Transformer，用于从视频演示中生成可执行程序。针对以往摘要方法和基于规则的方法的局限性，VT4 无需对平均演示进行摘要表示，即可从一组时间跨度长且彼此不连续的演示中生成程序。

- 我们评估了含噪规格对程序准确率的影响，并表明我们的模型能够适应大量错误检测。在感知原语的错误率为 10% 时，我们取得了当前最先进水平的结果。与严重依赖动态滤波器进行降噪的以往基于规则的模型相比，我们的表现显著更优。

- 我们评估了模型从部分可观测、视觉复杂的演示中生成程序的能力。与以往当前最先进水平相比，我们的精确程序准确率和别名程序准确率分别提升了 9% 和 11.75%。这分别意味着绝对提升 5.3% 和 7.7%，也是迄今为止该任务单次性能提升幅度最大的一次。

2 文献综述

视频理解任务 [18, 29] 可视为从视频演示生成程序（PGfVD）这一任务的子任务。与 PGfVD 一样，理解视频需要具备提取并理解事件与特征之间关联的能力。这通常通过能够执行动作识别和感知识别等任务的模型来实现 [15]。然而，与 PGfVD 不同，视频理解旨在描述观测到的内容，即单次演示中观测到了什么，而不是原因，即某件事为何发生。大多数情况下，这是一项直接的翻译任务，能够容许较大程度的歧义 [8]。这主要是因为对于视频翻译任务，多个字幕都可能是恰当的，并且在语义上被视为正确。然而，对于 PGfVD 任务，视频演示可能只展示了人们试图学习的整个规则集中的一个组成部分。此外，PGfVD 任务的歧义程度要低得多，因为程序的细微变化就可能导致截然不同的结果。

2.1 程序归纳

从观测中提取算法表示的任务称为程序归纳。许多研究采用了多种不同方法来解决这一问题，为该领域作出了贡献。 [11, 14, 17]采用图灵机等基于记忆的方法，而 [19]则采用端到端网络来解决代数文字题并加以解释。然而，与这些方法不同，我们旨在用领域特定语言生成一个定义完整的可执行程序。

2.2 程序合成

程序合成的目标是生成一个能够捕捉给定示例底层逻辑的程序。通常，这项任务将程序限制在简单的领域特定语言中，并涉及生成抽象语法树。这方面的工作包括 [20], 提出使用递归-逆向递归神经网络（R3NN）进行字符串转换。其他工作则将神经网络模型与搜索算法和基于规则的求解器结合起来 [3, 5]。此外，强化学习也被研究为解决程序合成任务的一种可能方法 [4, 22]。

然而，该领域的大多数工作并未考虑从视觉观测中合成程序的任务。 [24] 发现了这一点，并提出了通过观察多样化的视觉演示来生成程序的任务。为实现这一目标， [24]提出使用序列到序列 LSTM 模型。他们的模型由卷积神经网络组成，该网络将视频帧输入 LSTM 网络进行编码。 [24] 引入了平均池化与关系网络的组合，以总结编码后的演示，随后将其传递给 LSTM 解码器。 [10] 旨在通过引入偏差池化摘要器来替代关系网络，从而提高 [24] 方法的计算效率。此外， [10] 提出使用多个解码层来提升生成程序的准确性，最终使性能得到小幅改善。

[5] 针对这一问题采用了不同的方法，提出了一种混合模型，将用于提取规格的神经网络与用于生成程序的基于规则的求解器相结合。具体而言， [5] 提出使用卷积神经网络对视频帧进行编码，并使用两个不同的解码层来预测视频中观察到的感知和动作。[5] 正确指出，基于规则的求解器对输入规格噪声十分敏感 [6],，并提出了一种动态过滤方法，根据神经网络预测的置信度，忽略某些演示。引入动态滤波器后，[5] 的表现超过了先前基于 LSTM 的程序生成器。

3 方法

本节首先正式定义从视频演示生成程序这一任务，该任务最初由 [24], 提出；随后将介绍本文提出的 VT4 模型，并详细阐述我们的贡献。

3.1 程序生成

给定k 个视频演示来生成程序的目标，可以视为一个多序列到序列任务。程序由领域特定语言（DSL）定义，

程序由感知原语、动作原语和控制流语句组成。动作和感知原语分别定义了智能体与环境交互和感知环境的方式。DSL 语言的控制流语句包括 while 循环、repeat、if/else 语句和简单的逻辑运算。

3.1.1 程序

程序 $\pi_\theta(s_t) = a_t$ 定义为一个确定性函数，它以时刻 t 的状态$s \in \mathcal{S}$为输入，返回一个动作 $a \in \mathcal{A}$。对于此任务，我们将程序的参数限制为一种向量化领域专用语言，记作 $\theta \in \Theta$。这些参数通常被称为程序的“代码”。

3.1.2 演示

演示定义为从 $\pi(\theta)$ 中采样得到的状态-动作对序列

$$\tau = ((s_1, a_1), (s_2, a_2), ..., (s_T, a_T)) \tag{1}$$

然而，无法保证智能体按照程序 $\pi(\theta)$ 执行时生成的任何特定演示，都会提供程序 $\pi(\theta)$ 中所有控制流语句的示例。因此，有必要观察一组演示 $\mathcal{D} = (\tau_1, \tau_2, ..., \tau_k)$，其中$\mathcal{D} \subset D$ 包含程序 $\pi(\theta)$ 中所有控制流语句的转移示例。

3.1.3 动作与感知

鉴于用于定义程序 θ 的参数 π 的领域特定语言具有条件结构，我们采用感知原语来降低状态$s_{1 \to T}$的高维性。每个感知原语 μ 都以布尔值表示，共有 q 种可能的感知原语。这样，便可以通过向量$s = (\mu_1, \mu_2, ..., \mu_q)$来表示状态的高层表示。

对于在离散时间步中与确定性环境交互的智能体，假设该环境的状态$s \in \mathcal{S}$已预先定义，我们可以定义一个可能动作的有限集合 $\mathcal{A}$。据此，我们可以将状态之间的确定性转移建模为 $\mathcal{P}: \mathcal{S} \times \mathcal{A} \to \mathcal{S}$。在我们的情形中，每次转移只有一个可能的动作，因此可以将动作原语表示为长度为 m 的独热张量，其中m 等于可能动作的总数。

3.1.4 视觉语言

在定义了感知和动作原语之后，我们可以用它们来定义由语义令牌 $\psi \in \Psi$ 构成的视觉语言，并通过确定性函数 $F(x, y)$ 来表示，使得：

$$\psi_t = F(s_t, a_t) \tag{2}$$

这种将高层语义信息分词，以便于与 Transformer 模型配合使用的方法，已被证明非常有效 [23]。借助这一方法，我们还可以将转移标记代入方程1，得到：

$$\tau = (\psi_1, \psi_2, ... \psi_T) \tag{3}$$

3.2 VT4 模型

我们的 VT4 模型可分为两个主要部分：i）语义编码器和 ii）程序生成器网络。我们将根据视频演示生成可执行程序这一问题视为翻译与摘要相结合的任务。与以往将问题简化为纯粹的序列到序列任务的方法不同（这类方法会为多个演示创建摘要表达），我们将问题视为其本质上的多序列到序列任务。这消除了对多个多样化演示进行摘要时固有的信息丢失概率。

3.2.1 语义编码器

语义编码器本身可以分为两部分：神经模块和分词器。神经模块是一个多层卷积神经网络（CNN），用于学习检测每一帧中出现的感知原语，以及帧与帧之间执行的动作。动作预测网络由一个包含两层全连接层的五层卷积神经网络组成，并从头开始训练。感知预测网络使用一个预训练的 Efficientnet 模型 [25]。

给定一组视频演示 $\mathcal{V} = \{v_i\}_{i=1}^{k}$，我们希望获得每个演示对应的感知 p 和动作 a 序列。这是一个可直接建模的简单问题，表示如下：

$$p_{i,j} = MLP(CNN(v_{i,j})) \tag{4}$$

$$a_{i,j} = MLP(CNN(v_{i,j}), CNN(v_{i,j+1})) \tag{5}$$

其中i 和j 分别指第 i 个th 演示中的第 j 个th 帧。虽然可以仅凭单帧$v_{i,\ j}$ 利用纯粹的空间信息预测所有可观测的感知，但预测动作则并非如此。要预测任意时刻t 采取的动作，需要至少两帧所提供的时间信息。因此，我们将连续帧拼接在一起，如公式5所示，以进行动作预测。

现在，我们已经获得了每个视频中每一帧的预测动作$a_{i,\ j}$和感知$p_{i,\ j}$，可以利用这些结果创建语义令牌，完整封装每一帧所需的全部信息。具体来说，我们将预测结果输入语义编码器的第二部分，也就是分词器。分词器本身是一个确定性函数，逐帧接收预测的动作和感知作为输入。

$$\psi_{i,j} = F(p_{i,j}, a_{i,j}) \tag{6}$$

如第3.1节所述，我们的感知预测是一个多类别预测问题，表示为$p_{i,\ j} = (\mu_1,\ \mu_2,\ ...,\ \mu_q)$；而动作预测则返回一个单类别预测。我们的分词器首先将预测结果拼接成一个张量pa，然后计算基本幂级数的有限和。

$$\psi_{i,j} = \sum_{n=0}^{n} pa_n \times 2^n \tag{7}$$

现在，我们将演示编码为一系列语义令牌，其中封装了从原始视频帧中提取的所有语义信息。

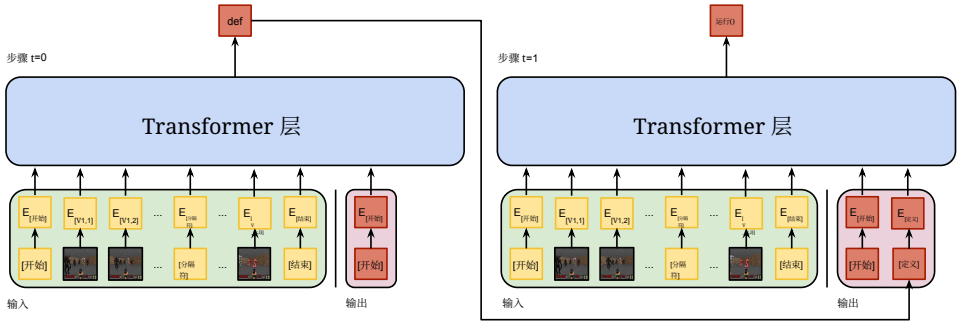

图 2：程序生成器模块。程序生成器由一个编码器-解码器 Transformer 网络构成，该网络接收语义编码器输出的视觉语言标记作为输入。该网络以序列方式迭代解码程序。

我们将每个经过令牌化处理的演示作为一组互不相交的序列输入程序生成器网络。这种创建“视觉词”[23] 的方法[23] 使我们能够借鉴自然语言处理领域的思路，将演示串接起来，并用特殊标记分隔不同的演示。V =(<开始 >$\psi_{1,\ 1}$，$\psi_{1,\ 2}$.....，$\psi_{1,\ l}$ <分隔 >，.....，$\psi_{k,\ l}$，<结束 >)

3.2.2 程序生成器网络

鉴于基于 Transformer 的架构在机器翻译等多种场景中取得了广泛成功 [26]，例如机器翻译 [1, 28], 自然语言处理 [7, 21],，甚至最近还应用于图像分类 [9],，我们假设它们在程序生成任务中也会非常有效。因此，我们的程序生成器采用编码器-解码器 Transformer 网络。借鉴 [23] 的研究成果，并将输入转换为视觉语言，我们便可以利用预训练语言模型。我们选择采用 [21]提出的预训练「文本到文本迁移 Transformer」（T5）网络。

T5 模型特别适合我们的任务，因为它将所有任务都统一表述为文本到文本任务。得益于此，我们可以利用上游自然语言任务（如摘要和翻译）中预训练的特征。T5 的实现严格遵循 [26]最初提出的架构。编码器由一系列「模块」堆叠而成，每个模块包含一个自注意力层、一个层归一化层和一个前馈网络。解码器的结构与之相似，但额外包含一个标准注意力机制。该标准注意力机制位于自注意力层之后，并关注编码器的输出。图 2 概述了 Transformer 网络的生成序列；完整细节请分别参阅 [21]和[26] 。

4 实验

本节将介绍对我们模型的实验评估，包括本次评估所用数据集和指标的概述。随后，我们将讨论实验结果，并

提供一项噪声消融研究，以评估模型处理噪声输入的能力。

4.1 数据集和指标

4.1.1 数据集

我们使用 Vizdoom 程序数据集评估我们的模型[24]，该数据集具有以下结构。对于每个程序标签，都存在多个视频演示，展示智能体在称为 Vizdoom 的确定性虚拟环境中执行该程序[16]。对于每个演示（长度为 T），都存在长度分别为 $T-1$和 T 的动作和感知标签。我们遵循 [24] 此前确立的约定，并在测试和训练期间对每个程序使用总计二十五个演示。我们还遵循以下假设：动作和感知标签仅在训练期间可用，而在测试时我们只能访问智能体的视频演示。该数据集包含 80,000 个用于训练的程序和 8,000 个用于测试的程序。

4.1.2 别名和精确准确率

由于验证两个程序是否确实相等是一个难以处理的问题，ˆ 我们通过比较合成参数 θ 与真实程序 θ^*的实例化参数来评估模型的准确率。当且仅当合成参数 θ 与真实程序的实例化参数 θ^*完全匹配时，我们才认为程序精确匹配。这可正式表示为，

$$Acc_{exact} = \frac{1}{N}\sum_{n=1}^{N} 1_{exact}(\hat{\theta}, \theta^*) \tag{8}$$

虽然精确准确率是衡量模型性能的一种简单方法，但它没有考虑程序空间的歧义性。正如 [24], 所指出的，可以利用领域特定语言的简单语法，根据一组既定规则枚举代码的多种变体。例如，可以将 if/else 语句等控制流语句拆解，也可以展开 repeat 语句。据此，我们可以将别名准确率正式表示为

$$Acc_{alias} = \frac{1}{N}\sum_{n=1}^{N} 1_{alias}(\hat{\theta}, \theta^*) \tag{9}$$

4.2 整体性能

我们在表 1中报告了在 Vizdoom 基准测试上的评估结果。为确保比较公平，我们严格遵循 [24]最初提出的实验设置。我们的 VT4 模型显著提升了相较于此前当前最先进水平的精确程序准确率和别名程序准确率，相对增幅分别为 9% 和 11.75%，绝对增幅分别为 5.3% 和 7.7%。这些结果清楚地证明，我们的模型能够同时对不连续序列进行摘要和翻译。这些结果也提供了实证证据，支持在需要翻译的任务中使用视觉语言来描述复杂场景的语义。此外，我们观察到

模型	精确	使用别名
demo2program[24]	53.2	62.5
watch-reason-code[10]	55.8	63.4
PLANS（动态）[5]	58.8 ±0.6	65.5 ±0.6
VT4（我们的方法）	**64.1**	**73.2**

表 1：我们结果与先前发表的研究结果的精确比较。

感知噪声	精确	别名
0%	66.0	75.1
10%	64.1	73.2
20%	62.7	71.6

表2：不同感知噪声水平下精确和别名程序准确率。

正如此前所有研究中始终观察到的那样，别名程序的预期准确率高于精确程序。

4.3 噪声消融研究

虽然我们的模型显然能够基于多个演示进行归纳推理，但我们也考察噪声对其准确生成程序能力的影响。以往采用基于规则的求解器的方法 [5] 对噪声非常敏感。事实上，即使感知准确率很高，PLANS 模型仍需要动态过滤。这表明，输入中哪怕只有极少量噪声，也会导致 PLANS 模型表现不佳。

为了测试模型处理输入噪声的能力，我们设计了一项噪声消融研究。通过使用两个不同的网络分别预测动作和感知，我们可以独立调整感知预测的准确率，而不影响动作预测。在这一设置中，我们的语义编码器包含独立的动作编码器和感知编码器。采用简单的五层卷积神经网络（如3.2.1节所述），动作编码器便可轻松达到 98% 的准确率。我们使用相同的架构训练感知编码器，其准确率达到 79.9%。随后，我们利用一个预训练的目标检测网络（effi- cientnet [25]）进一步提升这一结果。这一次，预测编码器的准确率达到 90.2%。我们利用预测得到的感知原语和动作来生成视觉语言，因此可以评估一种“理想情况”。由于我们掌握这些感知原语和动作的真实标签，因此可以直接使用它们生成视觉语言标记。这样做人为地构造出一种场景，在其中模型实际上没有任何输入噪声。

本研究的结果见表 2。结果清楚地表明，我们的模型不仅能够对多个演示进行推理，还能处理相互矛盾或含噪的信号。与基于规则的求解器不同，正如 [5] 所示，基于规则的求解器严重依赖预先定义的过滤启发式规则，而我们的 VT4 模型能够学习自己的启发式规则。这些结果还表明，即使感知预测的错误率达到 20%，我们的模型仍超越了此前的当前最先进水平。

5 结论

从多个视觉演示中合成程序这一任务涉及解决许多不同的问题，其中包括从视觉上复杂的输入中学习时空关系，并成功将其转换为不同领域中的逻辑序列。以往解决这一问题的尝试依赖于对时空关系的摘要表示，并且对输入噪声高度敏感。本文提出了一种视频到文本迁移 Transformer 网络，无需对时空嵌入进行摘要，即可执行多序列到序列翻译。我们的方法对输入噪声也具有很强的鲁棒性，而输入噪声正是以往方法面临重大挑战的原因。此外，我们在这项任务上的性能提升幅度创下迄今最高纪录。

参考文献

[1] Karim Ahmed、Nitish Shirish Keskar 和 Richard Socher。用于机器翻译的加权 Transformer 网络。*CoRR*，abs/1711.02132，2017 年。网址：http://arxiv.org/abs/1711.02132。[2]R. Alur、R. Bodik、G. Juniwal、M. M. K. Martin、M. Raghothaman、S. A. Se- shia、R. Singh、A. Solar-Lezama、E. Torlak 和 A. Udupa。语法引导的合成。载于*2013*年计算机辅助设计中的形式化方法，第 1–8 页，2013 年。doi：10.1109/FMCAD.2013.6679385。[3] Matej Balog、Alexander Gaunt、Marc Brockschmidt、Sebastian Nowozin 和 Daniel Tarlow。DeepCoder：学习编写程序。2016 年 11 月。[4] Rudy Bunel、Matthew J. Hausknecht、Jacob Devlin、Rishabh Singh 和 Pushmeet Kohli。利用语法和强化学习进行神经程序合成。*CoRR*，abs/1805.04276，2018 年。网址：http://arxiv.org/abs/1805.04276。[5] Raphaël Dang-Nhu。PLANS：从神经推断的规范中进行稳健的程序学习。*ArXiv*，abs/2006.03312，2020 年。[6] Jacob Devlin、Jonathan Uesato、Surya Bhupatiraju、Rishabh Singh、Abdel-rahman Mo- hamed 和 Pushmeet Kohli。RobustFill：在含噪输入输出条件下的神经程序学习。载于第*34*届国际机器学习会议论文集——第*70*卷，ICML'17，第 990–998 页。JMLR.org，2017 年。[7]Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。BERT：用于语言理解的深度双向 Transformer 预训练。*CoRR*，abs/1810.04805，2018 年。网址：http://arxiv.org/abs/1810.04805。[8] Jiarong Dong、Ke Gao、Xiaokai Chen、Junbo Guo、Juan Cao 和 Yongdong Zhang。并非所有词语都同等重要：视频字幕生成中的视频特定信息损失。*CoRR*，abs/1901.00097，2019。网址：http://arxiv.org/abs/1901.00097。[9] Alexey Dosovitskiy、Lucas Beyer、Alexander Kolesnikov、Dirk Weissenborn、Xiaohua Zhai、Thomas Unterthiner、Mostafa Dehghani、Matthias Minderer、Georg Heigold、Syl- vain Gelly、Jakob Uszkoreit 和 Neil Houlsby。一张图像胜过 16×16 个词：用于大规模图像识别的 Transformer，2020 年。

[10] Xuguang Duan、Qi Wu、Chuang Gan、Yiwei Zhang、Wenbing Huang、Anton van den Hengel 和 Wenwu Zhu。观看、推理与编码：学习使用程序来表示视频。载于第*27*届*ACM*国际多媒体会议论文集，MM '19，第 1543–1551 页，美国纽约州纽约，2019 年。计算机协会。ISBN 9781450368896。doi：10.1145/3343031.3351094。网址：https://doi.org/10.1145/3343031.3351094。

[11] Alex Graves、Greg Wayne 和 Ivo Danihelka。神经图灵机。*CoRR*，abs/1410.5401，2014 年。网址：http://arxiv.org/abs/1410.5401。

[12] Sumit Gulwani。使用输入输出示例自动处理电子表格中的字符串。*SIGPLANNot.*，46(1):317–330，2011 年 1 月。ISSN 0362-1340。doi: 10.1145/ 1925844.1926423。网址：https://doi.org/10.1145/1925844.1926423。

[13] Susmit Jha、Sumit Gulwani、Sanjit A. Seshia 和 Ashish Tiwari。预言机引导的基于组件的程序合成。ICSE '10，第 215–224 页，纽约，美国纽约州，2010 年。计算机协会。ISBN 9781605587196。doi: 10.1145/ 1806799.1806833。网址：https://doi.org/10.1145/1806799.1806833。

[14] Lukasz Kaiser 和 Ilya Sutskever。神经GPU学习算法。载于 Yoshua Bengio 和 Yann LeCun（编），第*4*届国际学习表征会议，*ICLR 2016*，圣胡安，波多黎各，*2016*年*5*月*2*日至*4*日，会议论文集，2016年。网址：http://arxiv.org/abs/1511.08228。

[15] Will Kay、João Carreira、Karen Simonyan、Brian Zhang、Chloe Hillier、Sudheendra Vijayanarasimhan、Fabio Viola、Tim Green、Trevor Back、Paul Natsev、Mustafa Su- leyman 和 Andrew Zisserman。Kinetics人体动作视频数据集。*CoRR*，abs/1705.06950，2017年。网址：http://arxiv.org/abs/1705.06950。

[16] Michal Kempka、Marek Wydmuch、Grzegorz Runc、Jakub Toczek 和 Wojciech Jaskowski。Vizdoom：一个基于Doom的视觉强化学习人工智能研究平台。*CoRR*，abs/1605.02097，2016年。网址：http://arxiv.org/abs/1605.02097。

[17] Karol Kurach、Marcin Andrychowicz 和 Ilya Sutskever。神经随机存取机器。载于 Yoshua Bengio 和 Yann LeCun 编，第*4*届国际学习表征会议（*ICLR 2016*），*2016*年*5*月*2*日至*4*日，波多黎各圣胡安，会议论文集，2016年。网址：http://arxiv.org/abs/1511.06392。

[18] Ji Lin、Chuang Gan 和 Song Han。TSM：用于高效视频理解的时间移位模块。载于*IEEE/CVF*国际计算机视觉会议（*ICCV*）论文集，2019年10月。

[19] Wang Ling、Dani Yogatama、Chris Dyer 和 Phil Blunsom。通过理由生成进行程序归纳：学习求解并解释代数文字题。*CoRR*，abs/1705.04146，2017年。网址：http://arxiv.org/abs/1705.04146。

[20]Emilio Parisotto、Abdel-rahman Mohamed、Rishabh Singh、Lihong Li、Dengy- ong Zhou 和 Pushmeet Kohli。神经符号程序合成。*CoRR*，abs/1611.01855，2016年。网址：http://arxiv.org/abs/1611.01855。

[21] Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、WeiLi 和 Peter J.Liu。通过统一的文本到文本转换器探索迁移学习的极限。*CoRR*，abs/1910.10683，2019。网址：http://arxiv.org/abs/1910.10683。

[22] Riley Simmons-Edler、Anders Miltner 和 H. Sebastian Seung。通过强化学习引导的树搜索进行程序合成。*CoRR*，abs/1806.02932，2018。网址：http://arxiv.org/abs/1806.02932。

[23] C. Sun、A. Myers、C. Vondrick、K. Murphy 和 C. Schmid。VideoBERT：一种用于视频与语言表征学习的联合模型。载于*2019*年*IEEE/CVF*国际计算机视觉会议（*ICCV*），第7463–7472页，2019。doi：10.1109/ICCV.2019.00756。

[24] Shao-Hua Sun、Hyeonwoo Noh、Sriram Somasundaram 和 Joseph Lim。基于多样化演示视频的神经程序合成。载于 Jennifer Dy 和 Andreas Krause（编），第 *35* 届国际机器学习会议论文集，第 80 卷，收录于机器学习研究论文集，第 4790–4799 页。PMLR，2018 年 7 月 10–15 日。网址：http://proceedings.mlr.press/v80/sun18a.html。

[25] Mingxing Tan 和 Quoc Le。EfficientNet：重新思考卷积神经网络的模型缩放。载于 Kamalika Chaudhuri 和 Ruslan Salakhutdinov（编），第 *36* 届国际机器学习会议论文集，第 97 卷，收录于机器学习研究论文集，第 6105–6114 页。PMLR，2019 年 6 月 9–15 日。网址：http://proceedings.mlr.press/v97/tan19a.html。

[26] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、undefinedukasz Kaiser 和 Illia Polosukhin。Attention Is All You Need。载于第*31*届神经信息处理系统国际会议论文集，NIPS'17，第6000–6010页，美国纽约州雷德胡克，2017年。Curran Associates Inc. ISBN9781510860964。

[27] Richard J. Waldinger 和 Richard C. T. Lee。Prow：迈向自动程序编写的一步。载于第一届人工智能国际联合会议论文集，IJCAI'69，第241–252页，美国加利福尼亚州旧金山，1969年。Morgan Kaufmann Publishers Inc.

[28] Qiang Wang、Bei Li、Tong Xiao、Jingbo Zhu、Changliang Li、Derek F. Wong 和 Lidia S. Chao。面向机器翻译的深度Transformer模型学习。*CoRR*，abs/1906.01787，2019。网址：http://arxiv.org/abs/1906.01787。

[29] Chao-YuanWu、Christoph Feichtenhofer、HaoqiFan、Kaiming He、PhilippKrahenbuhl 和 Ross Girshick。《用于详细视频理解的长期特征库》。发表于*IEEE/CVF*计算机视觉与模式识别会议（*CVPR*）论文集，2019年6月。