通过知识转换实现的程序化内容生成（PCG-KT）

Anurag Sarkar, M atthew Guzdial、Sam Snodgrass、Adam Summerville、Tiago Machado、Gillian Smith

arXiv:2305.00644v1 [cs.AI] 1 May 2023

摘要—**我们提出“通过知识转换实现的程序化内容生成”（PCG-KT）这一概念，将其作为一种新的视角和框架，用于刻画程序化内容生成方法及方法路径。在这类方法中，内容生成依托知识转换过程实现，即转换源自一个领域的知识，以便将其应用于另一个领域。我们的研究受到近期大量程序化内容生成研究的启发，这些研究着重于通过重新利用已获得的知识来生成新颖内容。例如，这类研究包括对基于某款游戏内容训练的模型进行迁移学习，使其适应另一款游戏的内容；也包括重新组合不同的生成分布，以融合两款或更多游戏的内容。这类方法的出现，部分源于基于机器学习的程序化内容生成（PCGML）的局限，例如难以为缺少训练数据的游戏构建生成模型，以及难以为全新的游戏生成内容。本文提出一个定义和框架，用于描述这类方法，并运用该框架综述现有研究，从而在这一新的 PCG-KT 视角下对其进行分类。最后，我们指出该领域尚待解决的问题，并展望未来研究方向。**

索引词—**程序化内容生成、迁移学习、概念融合、计算创造力、转换**

I. 引言

程序化内容生成（PCG）描述了通过算法手段生成游戏内容的一系列方法。我们可以从多个视角描述和理解程序化内容生成，包括基于搜索的程序化内容生成（SBPCG） [1], 将程序化内容生成视为一种构造性方法 [2],将程序化内容生成视为价值的表示 [3], 以及将程序化内容生成视为提供不同玩家体验的手段 [4]。这些视角主要都受到以下关注点的影响：如何为单个游戏形式化并生成内容。

近期，机器学习在程序化内容生成（PCGML [5]）中的应用受到关注，这不仅揭示了一种新的内容生成技术方法，也带来了受数据导向型内容生成方法启发的新研究问题和研究方向。PCGML 方法通常专注于学习现有内容上的分布

A. Sarkar 就职于美国马萨诸塞州波士顿东北大学 Khoury计算机科学学院（电子邮箱：sarkar.an@northeastern.edu）； M. Guzdial 就职于加拿大艾伯塔省埃德蒙顿阿尔伯塔大学计算机科学系、阿尔伯塔机器学习研究所（电子邮箱：guzdial@ualberta.ca）； S. Snodgrass 就职于丹麦哥本哈根的 modl.ai（电子邮箱：sam@modl.ai）； A. Summerville 就职于美国加利福尼亚州波莫纳的加州州立理工大学波莫纳分校理学院（电子邮箱：asummerville@cpp.edu）； T. Machado 就职于美国马萨诸塞州波士顿的东北大学（电子邮箱：tiago.la.machado@gmail.com）； G. Smith 就职于美国马萨诸塞州伍斯特的伍斯特理工学院计算机科学系（电子邮箱：gmsmith@wpi.edu）

然后从该分布中采样，以生成新内容。然而，PCGML 模型只能生成其训练数据所学习到的分布范围内的内容。这引发了 PCGML 中的一些核心问题：如何为训练数据很少或完全没有的游戏创作新游戏内容？又如何生成全新的内容，既能关联现有内容，又与之有所区别？这些问题催生了更多疑问和研究方向。例如，如何生成能够继承现有内容重要特征、同时又不直接重复或抄袭它的全新内容；或者如何重新组合从现有内容中提取的知识，以生成新内容并探索新的游戏和关卡设计。

我们发现，程序化内容生成（PCG）中存在若干可能由来已久的挑战，可以通过回答上述问题来应对。其中之一是自动化游戏设计，即创作全新的电子游戏。尽管此前已有一些解决这一问题的方法 [6],，但数量仍然相对有限。这在一定程度上是因为构建此类系统需要大量设计知识 [7]。如果我们能将现有游戏中的知识转换用于创作新游戏，就能绕开这一难题。

可控性是程序化内容生成（PCG）的另一项核心问题，重点在于如何让用户借助程序化内容生成系统尽可能实现其预期目标。可控性方面的一个常见问题是，程序化内容生成器无法满足用户的需求 [8]。例如，用户可能希望生成某些内容 X。如果该程序化内容生成系统最多只能生成某些内容 Y，用户就无法令人满意地控制该系统。但如果系统具备转换 $Y \rightarrow X$ 的能力，这一问题便可迎刃而解。

本文旨在解决此类问题，并介绍一种用于描述程序化内容生成研究与实践的新视角：通过知识转换实现的程序化内容生成（PCG-KT）。我们提供了一个框架和一套术语，用于根据程序化内容生成系统在不同领域之间转换知识的方法及程度来描述这些系统。我们还综述了在该框架中处于新颖位置的现有系统，并提出了这一视角所揭示的程序化内容生成研究潜在新方向。

我们想指出，提出 PCG-KT 是为了提供一种理解程序化内容生成过程的视角，而不是将其视为一类程序化内容生成方法。与 SBPCG 或程序化内容生成机器学习不同，我们并非在界定一组由某种底层人工智能技术关联起来的程序化内容生成方法。相反，我们借助这一视角和框架，根据程序化内容生成方法涉及知识转换的程度，对它们进行比较和对照。所有程序化内容生成

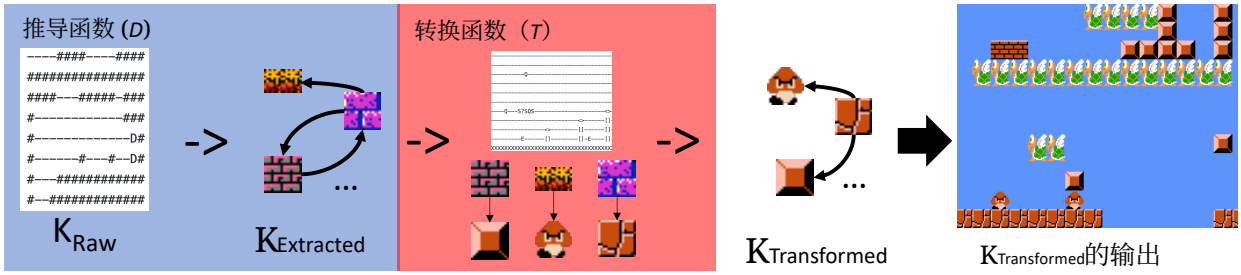


图1：将我们的框架应用于 [9]中一个示例的示意图。初始的*Kid Icarus* 关卡数据（原始知识 $\mathbb{G}_1$）通过推导函数（D）推导出马尔可夫关卡设计模型（提取的知识 $\mathbb{K}_1$）。该知识通过变换函数（T）作为输入传递，变换函数利用映射和一个马里奥关卡，为超级马力欧兄弟。关卡推导出马尔可夫关卡设计模型（转换后的知识 $\mathbb{K}_2$），该模型可用于输出新颖内容。

这些方法可以被理解为属于PCG-KT，但其中大多数涉及零转换或简单的转换。处于光谱另一端的是一些方法，它们会彻底转换推导出的知识，以近似次级知识或不存在的知识。为了清晰地介绍这些概念，本文重点讨论后者。不过，我们将PCG-KT视为一个描述性框架，而不是一组规定性条件，不将达到足够程度的知识转换作为符合PCG-KT的判定标准。

II. 基于知识转换的程序化内容生成

本节概述基于知识转换的程序化内容生成（PCG-KT）的术语体系和通用框架。我们承认，程序化内容生成涵盖范围广泛，且该领域使用的内容、知识和转换过程种类繁多。关于区分不同类型的知识转换的框架，以及现有系统如何符合这一框架的示例，请读者参阅第 III 节。

我们为 PCG-KT 提供一个形式化定义，以便识别先前的研究成果，促使人们思考这一视角如何既契合又重新诠释现有的程序化内容生成方法，并鼓励未来研究探索尚未充分研究的方向。

在定义中，我们采用以下词汇体系：

- **游戏**（$\mathbb{G}$：游戏集合；G：单个游戏 $G \subseteq \mathbb{G}$）。一个游戏或一组游戏构成领域，为知识转换过程提供定位和上下文。我们刻意不在此采用任何规定性的“游戏”定义：游戏可以是理论性的、正在开发中的或已发布的；可以是知名的，也可以鲜为人知；可以是研究用的“玩具”，也可以是成熟作品；可以是数字的，也可以是模拟的；可以是程序化生成的，也可以完全由人类创作。游戏的性质决定了能够衍生出何种知识，也影响着这些知识所处的更广泛背景和文化情境。
- **知识**（$\mathbb{K}$：从一组游戏中获得的知识集合；K：知识单元）。我们使用“知识”一词来描述从一组游戏中获得、可应用于一组新游戏的信息（经过或不经过转换）。这些信息可能采取多种不同形式，包括创作内容（精灵图、关卡地图）、功能设计方面（例如游戏机制、游戏状态），或与整体游戏体验相关的更广泛信息（例如推断出的风格启发式、自动化游戏轨迹）。知识的范围可能有所不同（例如单个精灵图或完整的规则集）。它也可以以一种独立于原游戏或目标游戏运行所使用形式的方式表示。例如，VideoGame LevelCorpus (VGLC) [10]关卡地图中使用的 ASCII 表示，就是从这些游戏中获得的知识，但并非这些游戏所使用的格式（事实上，它省略了原始关卡地图中的一些特征，例如不可交互装饰）。
- **推导**（一种从游戏中生成知识的推导函数 D：$D(\mathbb{G}_i) = \mathbb{K}_i$）。应根据可复现过程，从一组游戏中推导出知识。知识可以通过人工、自动化或某种混合流程推导得出。推导过程不必直接作用于作为制品的游戏（例如，从游戏 ROM 中提取的精灵图集）。例如，通过对游戏轨迹或游戏过程视频进行推断而得出的知识，仍然遵循与游戏相关的明确推导过程。
- **转换**（一种以知识为输入并生成知识作为输出的变换函数 T：$T(\mathbb{K}_i) = \mathbb{K}_j$）。该函数接收从一组游戏中推导出的知识，并对其进行修改，使之转而表示可从一组新游戏中推导出的知识。请注意，如果目标游戏与输入游戏足够相似，这可能是恒等变换（例如，从平台游戏中提取到的跳跃机制信息可以在这些游戏之间共享）。

通过知识转换实现程序化内容生成，是指将生成过程视为从一款（组）游戏中推导、转换并应用知识到另一款（组）游戏中的过程。更正式地说：

$$\mathbb{K}_1 = D_1(\mathbb{G}_1) \tag{1}$$

$$T(\mathbb{K}_1) = \mathbb{K}_2 \approx D_2(\mathbb{G}_2) \tag{2}$$

$$\mathbb{G}_2 \not\subseteq \mathbb{G}_1 \tag{3}$$

图 1 展示了该 PCG-KT框架的应用概览。

III. 基于知识转换的程序化内容生成框架

在本节中，我们提出一组用于描述程序化内容生成方法的方法和轴。利用上一节中介绍的定义，我们可以将许多现有的程序化内容生成方法视为表现出某种形式的知识转换。为清晰起见，我们将先讨论一个更高层次的描述类别，然后介绍其底层特征和可能的轴。

A. 知识结构

在讨论和描述这些转换方法时，考虑正在转换的知识的结构 十分重要。具体而言，我们考察转换过程三个不同阶段各自的结构：原始输入、提取的输入以及转换后的输出。

1) 原始知识： 这是最接近输入领域的知识，旨在表示输入游戏或领域本身，记作：

$$\mathbb{G}_i$$

其结构取决于转换的上下文，尤其是领域（即哪些游戏）；以及要从该领域提取的知识（例如关卡设计、故事结构、精灵图调色板、角色设计等）。例如，在许多程序化内容生成机器学习研究中，可以将关卡的基于图块的表示视为原始知识，如图 1 所示。这使得相关方法能够学习离散模式中的关卡结构（例如， [11], [12]）。或者，如果处理流程从输入关卡的图像开始，并将其转换为某种可供机器学习解析的表示（如图块），那么输入图像就会被视为原始知识。其他原始输入格式（如游戏轨迹的分段视频）则支持采用其他方法来学习关卡中的游戏玩法关系和物体放置 [13]。这与知识转换相关，因为包括转换可能输出在内的所有后续步骤都将取决于这些知识。因此，不同的原始知识格式对 PCG-KT 过程具有不同的可供性。例如，基于图块的原始知识不支持对传送门进行处理，因为其中不包含传送门。

2) 提取的知识：这是在执行转换之前，直接从原始输入知识中提取的知识，记作：

$$\mathbb{K}_i = D_i(\mathbb{G}_i)$$

其结构取决于原始输入知识、用于提取知识的方法，以及要执行的转换类型。这类知识可根据其属于直接还是潜在进行分类。在更接近直接的一端，我们可以设想一种多层 [14] 或层级式 [15] 的关卡表示，并标注关卡特征。在中间位置，我们可以采用学习得到的瓦片分布概率表 [11]。在潜在的一端，我们可以采用在这些关卡上训练的神经网络权重 [12]，或关卡的潜在向量表示 [16], [17]。

3）转换后的知识：最后，我们得到经过转换后的知识，记作：

$$T(\mathbb{K}_i) = \mathbb{K}_j \approx D_j(\mathbb{G}_j)$$

这种知识的结构可以用与转换前知识结构类似的方式来描述，因此我们不再重复相同的描述。不过，值得注意的是，尽管许多现有方法对提取的知识和转换后的知识采用相同的结构，但这并非必需。例如，我们可以设想一种方法：输入一组用于对关卡设计建模的学习所得概率表，并输出一组旨在将这些模型融合在一起的潜向量；或者输入两个经过训练、用于生成不同类型游戏内容的模型的神经网络权重，并输出适用于第三个未见过的游戏的内容。

B. 推导函数

推导函数是根据上述原始知识推导出提取的知识的过程。在知识提取方程 $\mathbb{K}_i = D_i(\mathbb{G}_i)$ 中，推导函数为 D_i。构建推导函数的方法多种多样。我们将其归纳为三大类：手工编写、混合式和自动化。这三类构成一个连续的尺度，函数既可以更多地由人工编写，也可以更少地由人工编写（或更加自动化）；完全手动和完全自动化则分别处于这一尺度的两端。

1）手动编写：手动编写的推导函数依赖人工编写和领域专业知识，从原始知识中提取知识。此类函数的一个例子是由领域专家分析一组输入关卡，然后为给定游戏制定一组设计规则（例如，洞的宽度只能是 4 个图块；洞穴关卡只能有特定的敌人，等等）。这些提取的知识随后可用于转换。另一个例子是在内容（关卡、故事、任务等）中手动标记和注释结构或模式。

2）自动化：在另一端，自动化推导函数依靠自动化规则、启发式方法或机器学习得到的信息，从原始知识中提取知识。例如，Summerville 和 Mateas [12] 以及 Snodgrass 和 Onta˜n´on [11]分别使用反向传播和出现次数统计作为提取知识的推导函数。在这些例子中，提取的知识就是训练得到的模型，可以是条件概率分布，也可以是 LSTM。如果考虑故事或任务领域，我们可以设想一种方法，从叙事文本中提取词嵌入作为提取的知识。这里的推导函数可以是简单的词语聚类方法，也可以是返回潜在向量编码的神经网络方法。

3）混合方法：这类方法结合了自动化方法与人类专业知识。实践中的大多数方法可能都属于这一类，因为即使是自动化方法，也可能依赖领域专业知识来设置规则、训练方案、标签等。以自动化方法为例：从游戏关卡中提取图块类型的条件分布，以建模设计知识。我们可以将其视为一种混合方法

如果用户定义模型应查看哪些图块邻域来提取分布，那么这种方法便是混合方法。或者，如果我们考虑在故事中标注结构或模式的手工编写方法，那么这种方法的初始输出可作为训练更自动化方法以标注新示例的起点。

C. 转换情境

转换的上下文，或转换知识将如何使用，是另一个需要考虑的重要因素。为描述这一上下文，我们提出以下几个维度，但这些维度并非面面俱到。

1) 新颖性：这描述了转换旨在近似现有领域，还是探索新领域。例如，在近似这一端，Snodgrass 和 Ontañón [9] 尝试学习从一个游戏到另一个现有游戏的关卡表示映射，其中已知一些先验知识。在另一端，Guzdial 和 Riedl [18]提出了一种探索方法，将从一组输入游戏中提取的知识进行转换和重组，以创建全新的游戏。可以根据对近似领域的先验知识掌握程度，以及所探索领域与输入领域的差异程度，将各种方法置于这一谱系之中。

2) 用法：上下文的这一方面与方法的用途有关，与其说是一个维度，不如说是对该方法在用户情境中所处位置的分类描述。例如，该方法是用于原型设计，以探索新的游戏创意；用于研究环境，以测试底层方法；还是用于制作一款供玩家体验的完整新游戏？了解上下文的这一方面，会直接影响可使用的知识类型，以及变换函数的形式。在此，用法有别于新颖性，因为对于相同的用途或目标，新颖性可以有不同程度。在设计新游戏机制的原型时，我们可能希望机制非常新颖（即与系统见过的所有先前机制都不同），也可能更关注这些机制与设计的其他方面（例如叙事、美学等）的契合程度。

D. 转换属性

这是一个涵盖转换诸多方面的宽泛类别，旨在描述知识被转换的程度，以及转换后的知识与输入知识之间的关系。这不同于定义中仅指出输入知识和输出知识，因为我们并未明确说明它们可能存在的关系。描述转换程度时，我们需要考虑转换后的知识与输入知识之间的距离。我们对距离一词的使用相当宽泛。我们不定义具体的距离度量，而是根据知识的结构和关注的属性，讨论两种距离类型。

1) 表征距离：指输入知识结构与转换后的知识之间的差异。例如，在两个潜向量之间进行插值时，表征距离相当小，因为所有知识都采用相同的编码。然而，例如，将故事图转换为游戏世界中的一组三维点时，这种距离就会大得多。请注意，许多方法的表征距离相当小，因为输入知识与转换后的知识通常具有相同的知识结构（例如，将一个瓦片集或精灵图集映射到另一个，或修改已学习的分布）。

2) 内容距离：这是输入知识与转换后的知识在内容上的差异。例如，从两个不同的 Mario 关卡中提取的精灵图集合之间的内容距离，远小于从一个 Mario 关卡和一个洛克人关卡中提取的精灵图集合之间的内容距离。如果考虑两种转换：一种通过组合 Mario 和洛克人的单个精灵图来生成一组新的精灵图；另一种则通过从 Mario 和洛克人的精灵图中随机选取一个子集来生成一组新的精灵图，那么，生成的混合精灵图集合与输入集合之间的内容距离，会大于生成的随机拼合精灵图集合与输入集合之间的距离。我们可以将此视为输入知识与转换后的知识之间共享信息量的等价体现，而不论这些信息以何种形式表示。举例来说，Sarkar 等人通过学习多个输入游戏的混合潜在表示来生成内容的方法 [17]，其内容距离高于 Snodgrass 和 Sarkar 通过二叉空间分割组合现有游戏关卡中的既有部分来生成新关卡的方法 [14]；而后者的内容距离又高于以往针对单一游戏领域运行的程序化内容生成机器学习方法。

请注意，这两种距离彼此独立。包含相同信息但表示方式不同的两组知识，表征距离会很大，但内容距离较小（例如，将游戏的关卡设计知识表示为一组关卡的基于图块的表示，与将这些关卡描述为一组用于放置物体和敌人的规则相比）。同样，两组知识也可以用相同的方式表示不同的信息。例如，在两款不同的游戏上训练的、架构相同的两个神经网络，会产生表征距离较小但内容距离较大的知识集，因为网络的权重会有所不同。

E. 转换函数

用于执行转换的函数与 PCG-KT 的其他所有方面都密切相关，因为它依赖于转换的上下文，需要各种知识结构来执行转换，而转换的属性则取决于该函数。这里我们将详细介绍描述该函数的各种方式。

1) 方法：这指的是执行转换的实际过程或方式。具体方法可以是更新概率表 [11],通过神经网络进行反向传播 [12],根据已有的提取图迭代扩展图 [18]，在潜向量之间进行插值 [16]，[17],或将来自不同领域的详细知识表示简化为单一统一的表示 [14], [17], ，等等。对于这一类别，

		Game 概念融合	**概念扩展**	**领域迁移**
知识结构	原始知识 提取的知识 转换后的知识	VGLC关卡 变分自编码器潜向量 混合变分自编码器潜向量 通过插值和标注	游戏过程视频 游戏图 概念扩展 of 游戏图	VGLC关卡 MdMC 条件分布 tion 转换后的 VGLC关卡 目标领域
推导函数	*Type* 描述	自动化 通过反向传播训练	自动化 规则学习与概率估计	自动化 估计 瓦片分布 通过观测计数
转换上下文	新颖性 用法	New 研究环境	New 研究环境	迁移 研究环境
转换属性	表示距离 内容距离	Low High	Low High	Low 中等
转换函数	方法 输入域 输出域 基数	变分自编码器的插值/演化潜向量 N 个游戏的集合 融合 N 的新游戏 多对一、多对多	概念扩展 N 个游戏的集合 表示为一个新游戏的游戏 N 游戏的组合 多对一、多对多	用于定义图块之间映射的多阶段管线 ing 图块之间的映射 目标 Game and 输入游戏 输入游戏转换为目标游戏表示 一对一、多对一

表 I：PCG-KT框架应用于三种技术：游戏混合、概念扩展和领域迁移。

所使用的函数几乎可以是程序化内容生成、机器学习、创造力、设计或人工智能文献中已采用的任何过程，包括手动方法和混合方法。明确指出转换发生时所采用的过程，有助于更清晰地界定整个转换方法。

2) 输入/输出领域：转换的输入域和输出域有助于刻画转换过程的性质。例如，在以融合不同游戏为核心的研究中，输入域是待融合游戏的集合，输出域则是最终的融合游戏。因此，此处的转换过程是将从各个游戏中提取的输入知识，转换为重新组合输入游戏属性的输出知识。类似地，在关卡设计的领域迁移中，输入域由一个或多个输入游戏以及一个目标游戏组成，输出域则由转换为目标游戏表示的输入游戏关卡组成。

3) 基数：这指的是输入源的数量和输出的数量。我们考虑四种基数类型：

1)一个输入域到一个输出域：这是一种标准的知识迁移，其中一个领域的知识被转换为用于表征另一个领域知识的形式 [9]。

2) 多个输入对应一个输出：这通常是一种概念融合方法，会以某种方式转换并组合输入领域，从而得到不同的输出域（可以是新的，也可以是已有的），如 [14], [17], [18] 所示。

3) 多个输入对应多个输出：这也可以是概念融合或近似，但更可能是一种探索性更强的方式，其中输入知识经过转换，并以各种方式重新组合，以探索不同的输出领域。例如 [19], [20] ，输入是一组游戏，但并非生成单一的融合输出，而是使用不同的标签来生成不同的融合组合，每种独特的标签配置都可以看作是将这组输入游戏映射到一个独立的融合领域，而该领域由输入游戏的独特组合构成。

4)一个输入对应多个输出：这也是一种更具探索性的变换，会转换单一领域中的知识，并可能将其抽象化，以探索多个输出域。

据我们所知，尽管已有系统先后将多个输入组合成多个输出，但尚无系统尝试同时生成彼此依赖的多个输出（例如一系列游戏、一组精灵图或一段配乐）。

为使用具体示例说明该框架，表 I 展示了将 PCG-KT框架应用于三个独立应用的结果：游戏混合 [16], 概念扩展 [18], 和领域迁移 [9], ，说明了上述讨论的特征和轴在每种情况下如何进行分类。原始知识 $\mathbb{G}_i$ 由游戏混合和领域迁移中的 VGLC关卡组成，而在概念扩展中则由游戏过程视频组成。在这三种情况下，推导函数（D_i（$\mathbb{G}_i$））都涉及一种机器学习算法（例如神经网络、贝叶斯网络或马尔可夫模型），用于获取提取的知识（$\mathbb{K}_i$）。对于游戏混合，这表现为潜向量；对于概念扩展，则表现为游戏图；对于领域迁移，则表现为条件分布表。对于变换函数（T（$\mathbb{K}_i$））， 概念融合采用潜向量插值，而概念扩展则需要一个多阶段管线，包括定义瓦片映射，以获得最终的转换后知识 $\mathbb{K}_j$。对于游戏混合，这由混合潜向量组成；对于概念扩展，则由表示新游戏的重组游戏图组成。

对于概念扩展的情况，以及领域迁移中目标领域的关卡。这些应用利用了两大类变换函数——组合创造力和迁移学习。这些内容以及表 I 中的示例将在第 V 节中进一步讨论。

IV. 相关工作

本节将讨论以往工作的相关领域，并将其与本文的关注重点进行比较。

A. 程序化内容生成

PCG-KT 是一种用于思考特定程序化内容生成过程的框架，因此，所有以往的程序化内容生成工作都属于相关领域。

一个明显的相关工作领域是通过机器学习进行程序化内容生成（PCGML） [5], ，因为我们认为 PCG-KT 能够解决这种方法面临的主要问题。不过，我们认为，PCG-KT 也适用于输入知识由人工编写的情形，例如 Permar 和 Magerko 的工作 [21]。因此，我们认为 PCG-KT 适用于 PCGML，但并不完全隶属于 PCGML。

监督式程序化内容生成机器学习尝试将以现有内容形式呈现的知识作为输入，并将其转换为更多内容（通过训练机器学习模型，再从该模型中采样）。然而，由于目标不同，在大多数情况下，我们不认为这一过程属于 PCG-KT。例如，一种程序化内容生成机器学习方法可能尝试学习关卡设计模型，以便生成适用于某个现有游戏（例如《超级马力欧兄弟》）的内容（如新关卡） [12], [22]。在这种情况下，目标明确是让输出关卡集合与原始知识实质上相似。通常的做法是基于原有的现存内容学习一个分布，然后从该分布中采样。因此，这个学习所得模型的任何输出都代表着源自原始游戏的相同知识 $\mathbb{G}$。虽然我们对 PCG-KT 的定义涵盖了输入游戏与输出目标游戏相匹配的情形，但这只代表一种最低限度或微不足道的转换。

我们还指出，PCG-KT 与大多数基于搜索的程序化内容生成（SBPCG）方法有所不同，例如遗传算法或进化搜索 [1]。在这些方法中，我们可以将搜索空间视为由作者自身的游戏设计知识所定义。因此，它源自一个理论上的游戏集合 $\mathbb{G}$；在本例中，该集合包含作者在定义搜索空间时所考虑的所有游戏。由此可知，该搜索空间中的任何输出都代表了来自 $\mathbb{G}$ 的知识。已有研究通过 SBPCG 生成了一款完整的可玩游戏 [6]。然而，即使在这些情况下，该游戏也已经存在于未改变的搜索空间中。因此，输出的游戏必然已经存在于 $\mathbb{G}$ 中。换言之，由于作者定义的游戏空间已包含输出游戏，因此并未发生任何转换，知识也保持不变。不过请注意，如果基于搜索的方法用于实现知识转换，则可以将其归类为 PCG-KT。是否适合将 PCG-KT 作为分析视角，取决于系统的目的，而非系统采用的技术方法。

B. 设计模式

在软件工程中，使软件更灵活且更易于复用的设计结构称为设计模式 [23]。设计模式的应用涉及多个学科；一般而言，设计模式描述某一环境中反复出现的问题，并将其映射到一组核心解决方案，这些方案可以重新应用和调整，而无须从头再造 [24]。由于游戏设计所涉及的活动并非完全关乎问题解决和工程技能，因此引入了游戏设计模式这一概念，以涵盖游戏机制、游戏设计和游戏艺术 [25], [26]。此前的研究曾使用游戏设计模式分析关卡设计 [27], [28], [29]和交互式叙事 [30]。Bj¨ork 和 Holopainen [25] 将游戏设计模式定义为游戏设计知识的组成部分；当它们出现在多款游戏中时便可以被识别出来（例如伙伴角色或可收集物）。已有研究尝试将这些设计模式应用于新游戏 [31]。然而，我们注意到，这些研究并未尝试将这些模式转换到其最初识别情境之外。因此，应用于某款新游戏的单个设计模式，被视为与原始游戏中的模式相同。

C. 迁移学习和领域适应

在将知识从一个领域转换到另一个领域时，PCG-KT 涉及迁移学习和领域适应；在后续章节中，我们将看到利用这些技术开展 PCG-KT 的实例。Zhuang 等 [40] 将迁移学习方法定义为：尝试利用为源任务学到的知识，并用这些源知识促进目标任务的知识学习。Pan 等 [41] 将领域适应定义为：利用不同但相关的源领域中的训练数据，解决目标领域中的学习问题。在游戏领域，除本文讨论的研究外，这类方法主要用于将智能体在一个环境中的行为迁移到另一个环境。Chaplot 等 [42]使用深度 Q 网络，将智能体在 VizDoom 环境中的导航行为从已见地图迁移到未见地图。需要注意的是，这不属于 PCG-KT，因为知识迁移发生在同一款游戏中。因此，不会为任何新游戏生成内容。Rusu 等 [43] 提出了渐进式网络，用于在不同 Atari 游戏之间迁移智能体行为；Kansky 等 [44] 则使用模式网络学习实体的基础物理模型，并将其用于训练期间未见过的 Breakout 新配置。Melnik 等 [45]学习了目标检测和基于轨迹的物理等模块，以将智能体策略从 Pong 迁移到 Breakout。类似地，研究者也尝试了多种方法，以便将一款游戏中学到的知识迁移并应用到另一款游戏中。Mittel 和 Munukutla [46]以及 Gamrian 和 Goldberg [47] 通过将已学习智能体的视觉方面与控制方面解耦来处理这一任务。 [46] 将智能体策略从 Atari Pong 迁移到 Breakout，而 [47] 则通过迁移学习构建能够玩 Breakout 和 Road Fighter 的智能体。不过需要注意的是，在 [43]-[47], 中，尽管部分有关智能体行为的知识

变换函数		论文	方法	已生成内容
Main 类别	次要类别			
组合创造力	概念概念融合	[13]	层次贝叶斯网络	关卡
		[16], [17]	变分自编码器（VAE）	关卡
		[32]	变分自编码器（VAE）	跳跃
		[19], [20]	条件变分自编码器	关卡
		[33]	图结构知识库	游戏
		[34]	手动概念融合	游戏
	概念扩展	[35]	贝叶斯网络	关卡
		[7], [18]	概率图模型	游戏
迁移学习	领域适应性	[9]	多维马尔可夫链	关卡
		[36], [14]	变分自编码器、二叉空间分割	关卡
	风格迁移	[37]	瓦片到可供性映射 自编码器/马尔可夫模型	关卡
	设计模式	[38]	基于案例的推理	关卡
	数据挖掘关联	[39]	Apriori 算法	Game 规则

表 II：第五节讨论的 PCG-KT 研究总结。

从一款游戏迁移到了另一款游戏，但迁移并未产生任何新内容，因此不符合 PCG-KT 的定义。在 [48], 中，Kim 等人介绍了 GameGAN，这是一种能够通过分别建模静态元素和动态元素来学习游戏模型的神经网络。这使得不同游戏的静态元素和动态元素可以互换，从而实现某种形式的转换。然而，这种转换幅度很小，其本质更接近于替换精灵图或背景图像，而非转换游戏结构或游戏机制。

D. 计算创造力

计算创造力（CC）是指在计算机中再现人类创造力的各种努力的统称 [49]。由于我们关注如何转换与程序化内容生成（PCG）相关的知识，以在特定创造过程中产生新颖的输出，因此这与计算创造力有关。计算创造力长期以来一直与游戏相关[50],，然而这方面的大部分工作都聚焦于“探索型”创造力，这与前文讨论的 SBPCG 密切相关。与 PCG-KT 更相关的一种创造力类型是“变革型”创造力 [51], ：概念空间（可视为类似于搜索空间或分布）通过改变表示该空间的变量而发生转换。这与我们对 PCG-KT 的定义有很强的相似之处，不过变革型创造力通常定义得宽泛得多。Boden 定义了第三种创造力：组合创造力 [51]。这种创造力指概念的组合，并启发了大量试图对其进行复现的计算过程，其中最著名的是概念融合 [52]。在心理学领域，这种方法称为概念组合 [53], ，数十年来一直作为一种普遍存在的认知过程受到研究。我们为变换函数引入的两个类别之一以组合创造力为基础。还有其他与 PCG-KT 存在一定交集的计算创造力方法。例如，新颖性搜索 [54] 和惊喜搜索 [55],分别试图在搜索空间或环境中找到新颖且令人惊喜的成员。尽管这看起来可能等同于 PCG-KT，但由于搜索空间保持不变（它所表示的知识未发生转换），我们并不

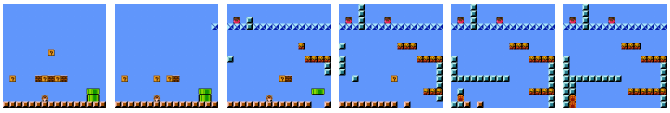


图 2：超级马力欧兄弟。与*Kid Icarus*概念融合示例。经 [16] 许可转载。

然而，我们可以设想将新颖性或惊喜搜索等方法应用于 PCG-KT，以寻求新颖或令人惊喜的转换。因此，我们认为这些方法以及其他计算创造力（CC）方法，与本文讨论的方法具有互补性。

V. 变换函数的类别

本节将综述归属于 PCG-KT 范畴的现有文献。尽管我们此前已介绍了 PCG-KT 的定义，以及设计 PCG-KT 系统时需要考虑的若干方面，但我们认为，PCG-KT 系统的主要区别在于其变换函数：也就是实际改变知识的过程。因此，我们将 PCG-KT 的变换函数分为两大类：（1）组合创造力；（2）迁移学习。第一类变换通过组合现有的程序化内容生成知识，生成能够产出新颖内容的其他程序化内容生成知识。第二类则借助迁移学习来转换知识 [40],，即从源领域获取知识，并对其进行调整，使其更适合某个目标领域。粗略来说，组合创造力对应多对多和多对一的基数关系，而迁移学习对应一对一的基数关系。我们将讨论这两类，并介绍各类中的子类别及近期的具体方法。为帮助读者了解所讨论的各项研究在 PCG-KT 框架中的定位，我们将重点说明输入知识的形式，以及应用转换过程后得到的输出知识形式。此处的输入知识和输出知识分别指此前所述变换函数的输入提取知识和输出转换后的知识（$T(\mathbb{K}_i) = \mathbb{K}_j$）。表 II 概述了本文讨论的研究。



图 3：洛克人 与 银河战士之间插值的示例。经许可转载，来源： [17]



图 4：概念融合关卡设计模型的输出示例。

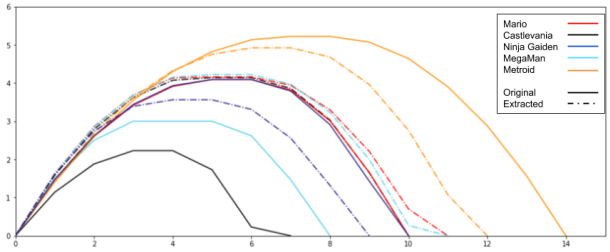


图 5：从融合域中提取的原始游戏跳跃（实线）与生成的跳跃（虚线点线）。经许可转载，来源： [56]。

A. 组合创造力

组合创造力是一种创造力，指通过以新颖、陌生的方式重新组合现有知识，创造新的制品、想法和/或知识[57]。Guzdial 和 Riedl 在早期的一项论述中借鉴了组合创造力，探讨如何解决利用监督式程序化内容生成机器学习生成新颖内容的问题 [35]。在计算层面表示组合创造力的方法有很多，其中包括融合 [58]和组合式适配 [59]，但这些方法大多尚未应用于程序化内容生成问题。因此，我们聚焦于两种具体的组合创造力方法：概念融合和概念扩展。

1）概念融合： 概念融合是一种组合创造力技术，侧重于通过组合或融合现有概念中的元素来产生新颖概念。Fauconnier 和 Turner [52] 在其“四空间”理论中对概念融合进行了形式化描述。该理论将概念融合描述为由四个空间构成：两个输入 空间，包含待融合或组合的概念；一个通用空间，两个输入概念被投射到其中，以寻找适合融合的等价点；以及一个融合空间，这些等价点被投射到其中，并由此演化出新颖的概念和模式。在接下来的章节中，我们将回顾运用概念融合的既有研究，以及如何在 PCG-KT 框架下对其进行规范。

在 [13], 中，Guzdial 和 Riedl 将概念融合应用于学习得到的关卡设计模型。这里的输入知识由表示为层次贝叶斯网络的关卡设计模型构成，因此这些模型本质上具有图结构。这使得将传统的概念融合算法应用于这些模型变得简单。输出知识是通过融合这些模型生成新关卡设计模型而得到的，这些模型又可以生成新类型的关卡。例如，将两个生成式关卡设计模型结合起来，一个模型生成水下 Mario 关卡，另一个生成城堡 Mario 关卡，从而创建一个能够生成图4所示水下城堡关卡的模型。

在 [33], Nelson 和 Mateas 概述了一种重新组合游戏机制以生成新游戏的方法。生成过程会从用户输入中提取动词和/或名词来描述主题，并据此生成合适的游戏；这一过程使用一组预先定义的抽象游戏模板，每个模板都可以通过不同的标准机制集来实现。因此，此处的输入知识包括游戏模板，以及根据用户提供的动词和名词选定并用于组合的机制集；输出知识则是通过组合所选模板和机制而生成的游戏。

在 [34], Gow 和 Corneli 提出通过融合现有游戏中的元素来生成完整游戏。在他们的框架中，输入知识由视频游戏描述语言（VGDL） [60]中的游戏规格构成。例如，他们手动将概念融合用作变换函数，作用于*Frogger* [61] 和《塞尔达传说》 [62]这两款游戏的 VGDL 规格。由此生成的输出知识是*Frolda*的 VGDL 规格；这是一款融合了 *Frogger* 的穿越危机四伏道路机制与*Zelda* 的锁与钥匙机制的新游戏。

这一框架通过 Sarkar 和 Cooper [16]以及 Sarkar 等人[17]的研究，借助机器学习得以实际运用。两者都使用了变分自编码器（变分自编码器） [63] ，其由编码器—解码器网络组成，用于学习输入数据的连续潜在表示。通过使用多个游戏的关卡进行训练，所学习到的变分自编码器潜在空间涵盖训练中使用的所有游戏，并能够采样生成混合关卡。在这些研究中，输入知识由从原始游戏关卡中提取的已学习潜在表示（取自超级马力欧兄弟。 [64], *Kid Icarus*[65], 银河战士 [66], 洛克人 [67], 《恶魔城》 [68] 和《忍者龙剑传》 [69]）以及由变分自编码器编码器和解码器学习得到的关卡与潜向量之间的映射构成。输出知识由混合关卡的潜在表示构成，这些表示通过在潜向量之间进行插值来应用概念融合，或通过演化这类

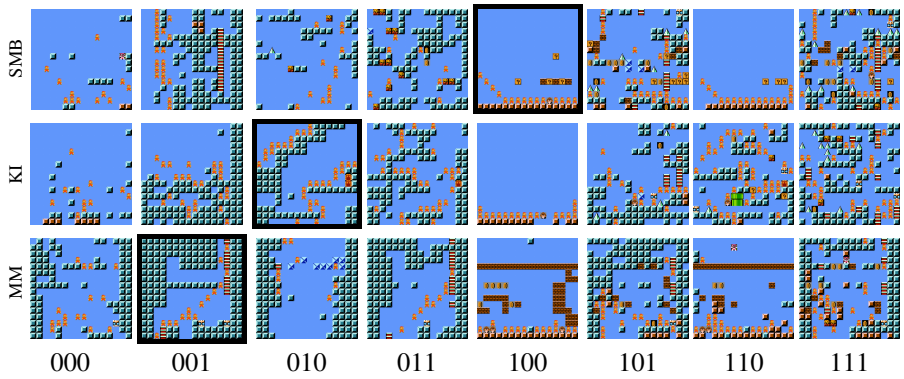


图 6：通过以融合标签为条件生成的片段（标签显示在第 3 行下方），使用来自超级马力欧兄弟 （SMB，第 1 行）、*Kid Icarus* （KI，第 2 行）和洛克人 （MM，第 3 行）的原始片段。标签中的第一、第二和第三个元素分别对应 SMB、KI 和 MM。带边框的片段为原始片段。经许可转载自 [19]

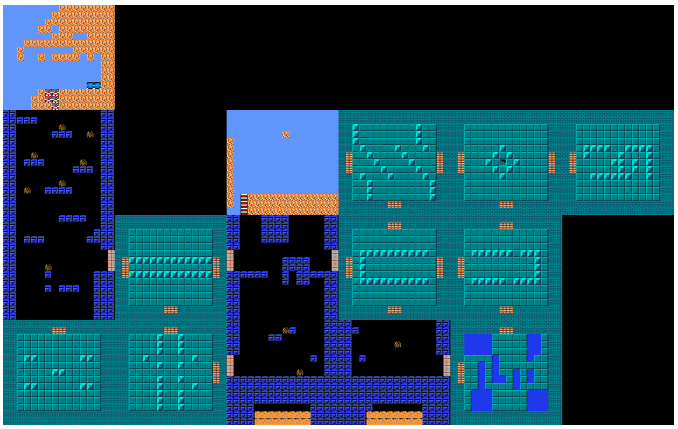


图 7：融合塞尔达传说、银河战士和洛克人关卡的示例。经许可转载自 [20]

向量获得。示例输出关卡分别如图2和图3所示。在后续研究中，Summerville 等人[56]利用生成的混合关卡中的路径提取了跳跃机制。提取出的跳跃弧线示例如图5所示。在此，输出知识还包括从混合关卡中提取的混合跳跃机制。

在后续研究中，Sarkar 等人 [19] 以及 Sarkar 和 Cooper[20] 使用了条件变分自编码器，即在变分自编码器的基础上额外使用标签来控制生成的混合关卡的属性，例如要融合的游戏组合以及方向。就 PCG-KT框架而言，这些基于条件变分自编码器的研究有一个有趣的区别：它们以多个融合域作为输出，因此呈现出多对多的基数，而基于变分自编码器的研究则是多对一的基数，因为后者只生成单个融合域作为输出。 [19] 中的示例关卡如图 6 所示。还需注意的是，通过实现对方向的控制， [20] 中的方法能够生成将地牢房间与平台游戏关卡片段相结合的关卡（例如图 7）。因此，与此前仅融合平台游戏的方法相比，该方法的输入知识与转换后的知识之间具有更大的内容距离。需要注意的是，在这些条件方法中，输入与输出知识与基于变分自编码器的方法类似，但这里概念融合的变换函数是通过标签操作实现的。

2）概念扩展： Guzdial 和 Riedl 提出了概念扩展，作为传统非潜在概念融合的替代方法，旨在实现对两个以上来源的机器学习知识进行融合[35]。其核心思路是将组合创造力过程拆分为一系列形如 $CE^X(\mathrm{A}\ F) = \alpha_0 * f_0 + \alpha_1 * f_1 ... \alpha_n * f_n$ 的函数。这里， F 是一组已有的机器学习特征，A 是一组筛选器，用于描述从各个特征中选取哪些内容，而 X 是一个目标。 X 可以是需要重现的明确知识片段，也可以是用于定义可接受性的启发式规则。因此，通过改变 F 和 A 的取值，优化方法可以在给定特征集合的各种可能组合构成的空间中进行搜索。对这些变量进行优化，便代表了概念扩展研究中的变换函数。

在 [35],Guzdial 和 Riedl 将概念扩展应用于超级马力欧兄弟 关卡设计的机器学习模型，以生成新类型的输出关卡（例如，将首领关卡和地上关卡结合起来，创建首领-地上关卡）。他们还采用了概念融合和另外两种组合创造力方法作为基线。值得注意的是，这些关卡并非被直接重新组合；相反，输入知识是学习得到的贝叶斯网络概率，用以表示关卡设计模型。将概念扩展用作变换函数后，输出结果是一个新颖的贝叶斯网络，能够生成前所未见的新关卡类型，如图 8 所示

在 [7], Guzdial 和 Riedl 将概念扩展应用于通过机器学习获得的游戏规则和游戏关卡表示，所涉游戏包括 超级马力欧兄弟、星之卡比 梦之泉物语 [70],和洛克人。这些输入知识采用知识图谱的形式，将游戏规则知识与游戏关卡知识结合起来，而二者均以游戏视频为原始知识来源。他们发现，在给定另外两款游戏以及留出游戏的部分规格说明时，概念扩展在重现留出游戏方面优于概念融合和遗传算法。然而，在这项工作中，Guzdial 和 Riedl 并未尝试生成

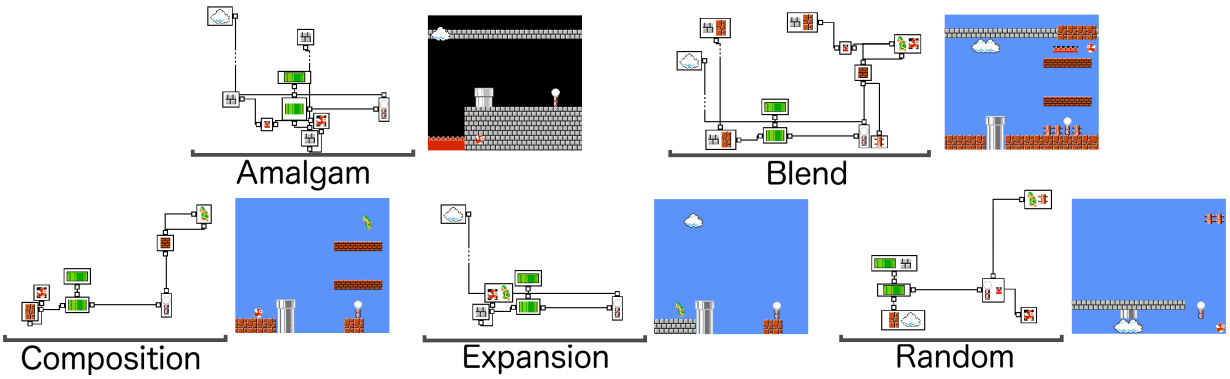


图 8：结合“城堡”和“地上”超级马力欧兄弟关卡时，不同组合创造力方法生成的不同最终模型及一个输出关卡示例。经 [35] 许可转载。

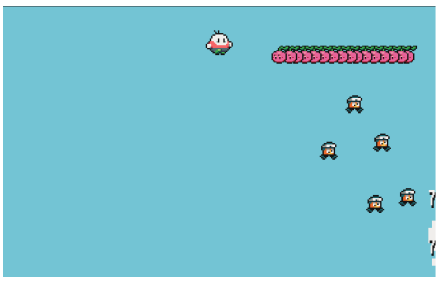


图 9：其中一个概念扩展游戏开头的屏幕截图。经 [18]许可转载。

完全新颖的游戏。相反，输出知识仅仅是基于该留出游戏的部分规格说明，对留出游戏所作的一种近似。

在 [18]，中，Guzdial 和 Riedl 展示了如何利用相同的输入知识生成新颖游戏：三个上述游戏的机器学习关卡图和规则信息。他们通过人类受试者研究，对这些游戏与其他组合创造力方法进行了比较评估。他们的变换函数采用手工编写的启发式方法和在概念扩展表示上的贪心搜索，对输出图进行优化。结果表明，概念扩展游戏的表现优于现有组合创造力方法生成的游戏，但该启发式方法针对游戏质量的设计并不理想。如何最好地引导可能组合的搜索，仍然是一个重要的开放问题。我们在图 9 中给出了其中一个游戏的屏幕截图。

B. 迁移学习

迁移学习是一种机器学习技术，专注于将一个领域中的信息或知识迁移并应用于另一个领域。在以下各节中，我们将讨论此前采用迁移学习作为转换函数的 PCG-KT 方法。

1）领域适应：这指的是将从源领域提取的知识加以转换，以补充或近似另一个目标领域的知识的方法。现有的用于关卡生成的领域适应方法侧重于寻找源领域与目标领域表征之间的映射。

在 [9], Snodgrass 和 Onta˜n´on 提出了一种方法，旨在自动定义多个平台游戏关卡领域的领域表示之间的映射。该方法的核心是使用领域之间的映射子集，将源领域关卡转换为目标领域表示，并过滤掉表现不佳的映射。通过比较转换后的输入域和目标领域中的瓦片出现频率及结构（例如， 2×2 瓦片块）来评估潜在映射。该方法的输入知识是来自输入域的瓦片关卡集合。这里的转换函数是映射管线，用于尝试定义如何将输入域中的图块映射到目标领域中的图块。应用此映射后，输出知识是一组重新表示为目标领域关卡的关卡。我们在图 10 中列出了不同映射对应的不同输出关卡示例。

不同于上述旨在将源领域转换后用于补充目标领域的工作，Snodgrass [36]和 Snodgrass 和 Sarkar [14]旨在将源领域转换为中间目标领域，然后再转换回源领域 [36] 或转换为新的次级目标领域 [14]。在这些方法中，作者将源领域的关卡转换为一种抽象领域表示，该表示旨在仅捕捉各源领域关卡的结构属性（即，将每种瓦片类型映射为实体类型或空类型）。在 [36],中，他们将抽象关卡作为关卡生成器的输入，由生成器创建该关卡的变体。这些变体通过将抽象关卡中的结构与源领域中的结构相匹配来生成，并利用源领域关卡的区域填充抽象关卡的细节。在 [14]，中，他们在抽象关卡上训练变分自编码器，并在抽象领域中生成新的关卡。随后，他们将生成的抽象关卡作为生成器的输入，让生成器混合源领域，而不是将其转换回某个单独的源领域。在这种方法中，混合关卡通过将生成的抽象关卡中的结构与所有源领域中的结构相匹配来创建，并利用不同的源领域填充生成的抽象关卡的细节。因此，

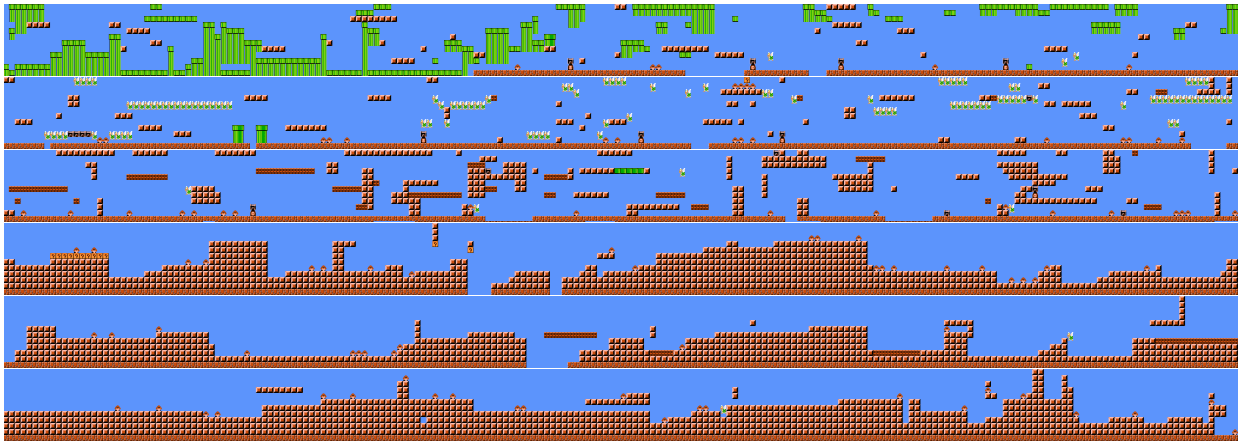


图 10：从超级马力欧兄弟到*Kid Icarus* 和*Kid Kool*的领域适应马尔可夫模型生成的输出关卡。经 [9]许可转载。

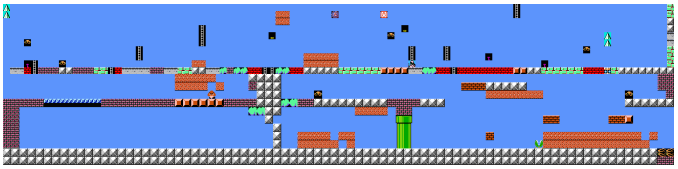
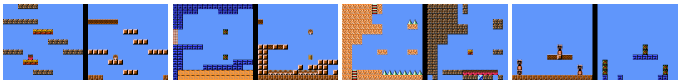


图 11：结合领域适应与概念融合方法生成的《恶魔城》关卡示例。经 [14]许可转载。



KI 转 SMB　Met 转 SMB　MM 转 KI　SMB 转 Met

图 12：涉及*Kid Icarus*（KI）、超级马力欧兄弟 （SMB）、洛克人 （MM）和银河战士（Met）的风格迁移关卡示例。经许可转载自 [37]

这里的输入知识由抽象表示中的关卡构成，这些关卡从原始输入关卡中提取而来；输出知识则由应用变换函数后获得的混合关卡构成。本例中的变换函数就是上文所述的结构匹配过程。请注意，尽管 [14] 同时包含领域适应阶段和组合创造力阶段，但我们仍将其纳入本节，因为在这种方法中，前者使后者成为可能。图 11 展示了使用这种方法生成的《恶魔城》关卡示例。

2）平台游戏风格迁移：风格迁移 [72] 是创造性机器学习在视觉艺术领域的一种热门应用，也是大量研究的主题 [73]。为了在游戏中实现风格迁移，Sarkar 和 Cooper 提出了 tile2tile[37], ，这是一种在不同平台游戏的关卡之间迁移风格的方法。在 PCG-KT 的上下文中，我们可以将游戏中的风格迁移视为提取一种游戏的风格知识，并将其转换为另一种游戏的风格知识。虽然游戏混合与风格迁移相似，但两者并不相同。风格迁移是将一个领域的风格转换为另一个领域的风格，而概念融合的目标则是取两个领域的风格和/或内容并将其结合起来，创造出

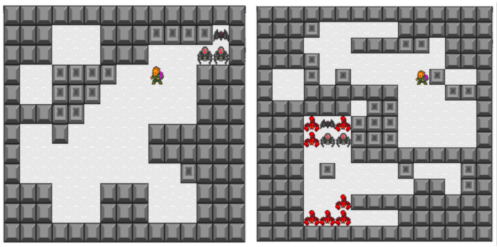


图13：由广义 3×3 结构构建的构造式生成器（左）和基于搜索的生成器（右）。经 [38] 许可转载。

新事物。tile2tile 方法通过预定义的映射，将源游戏的关卡转换为基于可供性的表示，然后使用机器学习模型，将这种可供性表示转换为目标游戏特有的图块。该方法将关卡的底层可供性和整体游戏特有的瓦片分布，分别视为与视觉艺术风格迁移中的内容和风格概念相对应。这里的输入知识由源游戏的关卡构成，输出知识则是迁移了风格并转换为目标游戏的同一批关卡，而变换函数由人工编写的瓦片到可供性映射和经过训练、用于将可供性转换为目标游戏图块的模型共同构成。图 12 展示了风格迁移后的关卡示例。

3）多游戏关卡生成的设计模式方法：在 [38]，中，Beaupre 等人采用概括与选择的方法，通过概括多个游戏中的关卡设计模式，以程序化方式生成特定游戏的新关卡。这种方法生成了不属于现有关卡集合的新游戏关卡，例如包含默认 Pacman 豆粒以外可收集物的 Pacman 关卡。他们的方法将 VGDL 中的所有游戏关卡全部纳入，并以更通用的表示形式重新表示每个关卡（例如，所有敌人现在都视为相同，所有可收集物现在也都视为相同，等等）。随后，他们从这种表示中提取 3×3 模式，并以此构建生成式和搜索式的

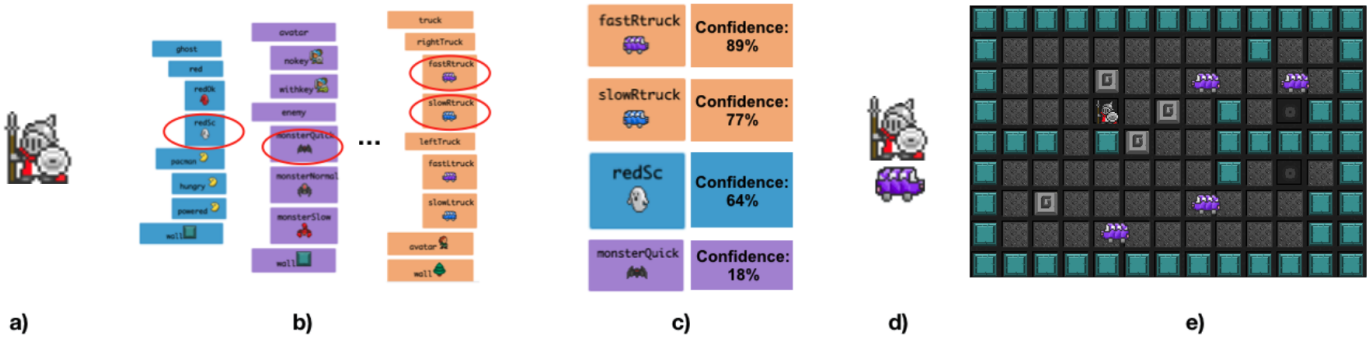


图 14：*Pitako*的推荐流程。经 [71]许可转载。在此上下文中，所有精灵图都与某种游戏行为相关联，例如射击、追逐、随机移动等。在**a)**中，用户向游戏引擎提供一张精灵图。在**b)**中，*Pitako*通过数据挖掘关联（以置信度作为排序标准），从其他游戏中筛选并列出匹配度最高的精灵图（红色圆圈）。在**c)**中，用户可以查看排名。在**d)**中，用户检查排名后选择一张新精灵图。在**e)**中，用户可以查看两张精灵图在游戏中的实际效果——一张是用户提供的，另一张是系统推荐并从其他游戏导入的。

程序化内容生成器，如图 13 所示。值得注意的是，这种方法没有使用统计机器学习，而是采用了类似于基于案例的推理的方法。在这项工作中，输入知识由从原始 VGDL 关卡中提取的通用 3×3 设计模式构成。输出知识则由生成的关卡组成，而这些关卡又是通过基于进化的变换函数，利用提取出的设计模式构建而成。

4）用于游戏规则重组的数据挖掘关联：在 [39], Machado 等人介绍了 Pitako，这是一个用于游戏机制的推荐系统，可在他们此前开发的游戏设计工具 Cicero 中使用。该系统基于 VGDL 和 GVGAI（通用电子游戏人工智能）框架 [74] 构建，并使用关联规则挖掘，向设计者建议添加游戏元素和规则。关联规则通常用于事务数据库，其形式为$X \rightarrow Y$：如果某个事务中存在项目 X，则意味着同一事务中也存在项目 Y，且具有一定概率。为将这一概念置于游戏情境中，试想《洛克人*X*》 [75] 和《超级银河战士》 [76]。两款游戏都有射击机制，也都具有“按住按钮以发射强力射击”的机制。也就是说，如果一款游戏实现了射击机制，那么它也可能实现强力射击规则。将其表述为事务，即拥有射击机制意味着同一款游戏中也存在强力射击机制。关联规则通过 Apriori 算法等数据挖掘关联算法[77] 对事务数据库进行分析后得出。为此，Pitako 使用了一个游戏目录。该系统使用 GVGAI 框架中的游戏，并对每款游戏进行深度优先树搜索，以读取所有游戏描述并将其加入目录。在目录中，每款游戏的每个游戏元素都能脱离其原始来源，并轻松插入目标游戏。目录的组织方式便于对其中的项目进行分类；这些项目是描述 VGDL 游戏所需的元素，可作为数据挖掘关联算法的输入。图 14 概述了 Pitako 的推荐流程。流程从获取设计者的游戏描述集开始。在 (a) 中，用户只有一个游戏元素。在 (b) 中，系统使用频繁项集数据挖掘，找出与设计者游戏中的元素相匹配的关联项，这些关联项来自游戏目录中的元素。红色椭圆表示来自不同游戏、可能被推荐的元素。该流程会生成一份推荐列表，并按规则的置信度从高到低排序，如 (c) 所示。设计者选取置信度最高的推荐项 (d)，并将其用作游戏中的新元素。此处展示的元素来自一款 Frogger 克隆游戏；在本例中，用户将 Frogger 和 Sokoban 的元素混合在一起。用户选取推荐项后，该游戏元素便从目录中的某款游戏迁移并插入正在开发的游戏中。因此，这里的输入知识是通过搜索从 GVGAI 游戏中提取的游戏目录，而输出知识则是设计出的游戏，其中包含用户从输入游戏中选取的元素。转换过程既包括对目录应用 Apriori 算法以生成推荐，也包括用户在设计游戏时使用这些推荐。值得注意的是，此处的转换过程属于混合主动式，与其他受调查工作中更为自动化的转换不同。

VI. 讨论

本节总结与 PCG-KT 相关的未解决问题及未来研究方向。

A. 评估

在生成过程中，如何最好地评估知识转换尚不明确。现有研究采用了表达范围分析 [78], 可玩性度量和用户研究等技术，而基于机器学习的系统则使用训练分布与生成分布之间的比较[79]。然而，这些方法关注的是评估生成输出的优劣和功能性，而非评估转换本身的属性。此外，在某些情况下，内容评估方法本身也有待改进。例如，在游戏与关卡融合的情形下，如何最好地评估融合后关卡的质量仍不明确

，以及融合所产生的、属于全新领域的关卡。目前常见的方法是，将新领域或融合领域生成的内容与原始输入领域中的内容进行比较；然而，这类比较往往只能表明是否发生了融合或转换，却无法提供有关融合与转换的程度、类型和质量的信息。这类似于我们上文提出的衡量变换函数内容距离的概念，但对转换的其他属性所能说明的内容有限。因此，未来需要开发更稳健、信息量更高的评估技术。这可能需要定义特定指标，以评估知识转换的各种属性，或许还可以利用我们在框架定义中强调的其他一些特征。

B. 扩展到多种游戏类型

大多数采用知识转换（以及一般意义上的程序化内容生成）的方法都聚焦于平台游戏。尽管使用同一类型的游戏更方便，但局限于单一类型会限制转换过程的范围和可能空间。将知识转换方法扩展到多种游戏类型和领域，有望生成新类型和更多样化的内容。可以理解的是，利用从多种游戏类型中提取的知识开展工作，会在开发知识推理技术以及以有意义的方式应用转换方面带来新的挑战。不过，若能做到这一点，便有望创造新颖的游戏体验。我们能否设计出一种变换函数，将从 Mario 这类平台游戏中提取的知识与从 Zelda 这类冒险游戏中提取的知识结合起来？如前所述，Sarkar 和 Cooper [20]在这方面迈出了一步，生成了融合平台游戏片段与地牢房间的关卡，但忽略了此类关卡中的游戏机制问题。解决这些问题，并以有助于实现有意义的组合与转换的方式从不同游戏类型和领域中提取知识，或许能让我们构建出可以融合不同游戏类型、而不仅仅是关卡或游戏的系统。例如，这样的系统能否将平台游戏的机制与冒险游戏中以钥匙和锁为核心的推进方式结合起来，生成一款银河恶魔城游戏？就 PCG-KT框架而言，这样的系统会展现出输入知识与输出知识之间内容距离更大的转换属性，因为变换函数的输出域不仅会是一个新游戏，甚至可能是一个全新的游戏类型。

C. 模型与技术的结合

在此前讨论的所有研究中，知识转换过程都是通过一种主要模型或架构来实现的，例如马尔可夫链、贝叶斯网络和变分自编码器。未来，考虑以某种有意义的方式结合这些模型的混合式知识转换过程，将会很有意思。用作推导函数的模型选择决定了提取的知识的形式。我们已经看到，这种形式可以是潜在向量表示（用于变分自编码器时），也可以是概率表（将其应用于类似的原始知识时，用于马尔可夫链时）。因此，探索如何结合这些不同的提取知识表示，以指导并影响转换过程，从而获得不同种类的输出知识，或许会很有价值。例如，用于概念扩展的模型能否与用于领域适配的马尔可夫模型结合，或与变分自编码器结合，以学习游戏图的连续潜在空间？或者，能否对不同游戏的潜向量或其他先前研究中的设计模式应用概念扩展？此类模型集成可视为分层知识转换：从一个推导函数中提取的知识被用作另一个推导函数的输入。换言之，这可以看作是将单个推导函数 D 替换为推导函数的组合 $D_n \circ D_m \circ ... \circ D_b \circ D_a$；该组合在每一步应用后都会生成中间提取知识$\mathbb{K}_a$，$\mathbb{K}_b$...$\mathbb{K}_m$，$\mathbb{K}_n$ 。研究如此逐步提取的知识的特征，以及它们可能如何影响转换后的知识，也将是未来研究的另一个富有前景的方向。同样，我们可以设想由变换函数 $T_n \circ T_m \circ ... \circ T_a$ 构成的组合，对给定的提取知识依次进行转换，并在每一步生成新的转换后知识。这可以表现为一个系统：先生成游戏融合（通过概念融合），再将融合游戏的风格迁移到另一款游戏中（通过领域迁移）。

D. 设计工具与可控性

未来工作的一项重要方向，是提高知识转换过程的可控性，并通过面向用户的设计工具，让更多人能够使用 PCG-KT 模型。例如，可以考虑前文讨论过的概念融合相关研究。虽然这些方法通过指定目标函数和受监督的机器学习模型来实现可控性，但使用者需要具备机器学习和进化算法方面的经验与知识，因此实际上只有研究人员和专家才能使用这类方法。此外，本文讨论的若干方法，尤其是基于机器学习的方法，可以视为黑箱：它们接收输入（以及可选的一组约束），然后生成输出，却几乎不为用户参与生成或转换过程留出空间，因此用户的创作控制权十分有限。未来的设计工具可以针对这些问题提供解决方案，例如，将模型学习到的潜在设计空间可视化，让用户交互式地探索这些空间并引导生成过程。总的来说，未来工作需要着力开发工具，让更多不同类型的用户能够控制这些模型（例如，提供用户界面元素和交互界面，而不是要求用户定义独热编码标签并编写进化算法的适应度函数）。这类工具可以借鉴大量现有的协同创作系统 [80], ，尤其是基于机器学习的系统，例如用于制作 Mario 关卡的 Morai Maker [81] 和 TOAD-GUI [82]、Schrum 等人开发的工具 [83]（用于交互式探索经训练的 GAN 的潜在空间）

，这些 GAN 使用 Mario 和 Zelda 训练；此外还有 Lode Encoder [84]，这是一款基于变分自编码器、用于生成 *LodeRunner* [85]关卡的工具。请注意，在所有这些例子中，工具都只适用于单一游戏或领域，因此不能视为 PCG-KT 工具。事实上，前文讨论过的 Cicero 和 Pitako 工具是少数 PCG-KT 设计工具中的代表，这也说明未来有必要开发更多此类工具。Sarkar 和 Cooper [86] 为这类工具提供了蓝图，但截至撰写本文时，尚无具体实现。提高 PCG-KT 方法的可控性，并通过易用的工具将其付诸实践，可以进一步推动这一领域的发展，促进 PCG-KT 方法更广泛地应用于游戏和关卡设计。除了帮助经验较少的用户使用的工具，未来工作还可以探索将用户知识融入转换过程的工具。这类工具可视为创造力支持工具 [87]：用户输入来自不同领域的知识，工具则让用户能够从多种自动化推导函数中选择并应用、创建自己的手工编写推导函数，或指出输入知识中适宜转换的部分。由此，这类工具能够帮助将用户的创造性见解融入 PCG-KT 过程，推动形成更具协同创造性、以人为本的 PCG-KT，而非本文综述的那些更自动化的方法。

七、结论

本文介绍了通过知识转换实现的程序化内容生成（PCG-KT），这是程序化内容生成研究的一种新视角，侧重于通过在不同领域之间转换知识来生成内容的方法。我们为 PCG-KT 提供了正式定义，概述了一个用于在这一视角下刻画相关研究的框架，讨论了若干符合该框架的现有研究实例，并以未来研究方向作结。此前关于基于搜索的程序化内容生成 [1] 和程序化内容生成机器学习（PCGML） [5] 的综述论文，成功地凸显了特定的一类程序化内容生成方法，并指出了未来研究方向，从而启发了后续研究。例如，本文重点介绍的若干研究直接受到 PCGML 论文中建议的启发。我们希望本文也能取得类似成效。通过强调程序化内容生成这一新分支，我们希望推动这一有趣且令人振奋的新方向上的进一步研究。

参考文献

[1] J. Togelius、G. N. Yannakakis、K. O. Stanley 和 C. Browne，“基于搜索的程序化内容生成”，载于欧洲进化计算应用会议论文集。Springer，2010年，第141–150页。[2] S. Risi 和 J. Togelius，“通过程序化内容生成提高机器学习的通用性”，《自然·机器智能》，第2卷，第8期，第428–436页，2020年。[3] G. Smith，“我们看重程序化内容生成中的什么？”，载于第十二届数字游戏基础国际会议论文集，2017年，第1–2页。[4] G. N. Yannakakis 和 J. Togelius，“经验驱动的程序化内容生成”，IEEE《情感计算汇刊》，第2卷，第3期，第147–161页，2011年。[5] A. Summerville、S. Snodgrass、M. Guzdial、C. Holmg˚ard、A. K. Hoover、A. Isaksen、A. Nealen 和 J. Togelius，“通过机器学习实现的程序化内容生成（PCGML）”，IEEE《游戏汇刊》，第10卷，第3期，第257–270页，2018年。

[6] M. Cook、S. Colton 和 J. Gow，“angelina 电子游戏设计系统——第一部分”，IEEE《计算智能与游戏人工智能汇刊》，第9卷，第2期，第192–203页，2016年。[7] M. Guzdial 和 M. Riedl，“通过概念扩展实现自动化游戏设计”，载于《第14届AAAI人工智能与交互式数字娱乐会议论文集》，2018年。[8] M. Guzdial、N. Liao、J. Chen、S.-Y. Chen、S. Shah、V. Shah、J. Reno、G. Smith 和 M. O. Riedl，“朋友、协作者、学生、管理者：人工智能驱动的游戏关卡编辑器设计如何影响创作者”，载于《2019年CHI计算系统中的人为因素会议论文集》，2019年，第1–13页。[9] S. Snodgrass 和 S. Onta˜n´on，“二维电子游戏关卡程序化内容生成中的领域迁移方法”，载于《第12届人工智能与互动数字娱乐会议论文集》，2016年。[10] A. J. Summerville、S. Snodgrass、M. Mateas 和 S. O. Villar，“VGLC：电子游戏关卡语料库”，《第七届程序化内容生成研讨会论文集》，2016年。[11] S. Snodgrass 和 S. Onta˜n´on，“学习使用马尔可夫模型生成电子游戏地图”，IEEE《计算智能与游戏人工智能汇刊》，第9卷，第4期，第410–422页，2016年。[12] A. Summerville 和 M. Mateas，“作为字符串的超级马力欧：通过长短期记忆网络生成平台游戏关卡”，载于《DiGRA与FDG第一届国际联合会议论文集》，2016年。[13] M. Guzdial 和 M. Riedl，“学习融合电脑游戏关卡”，载于《第七届国际计算创造力会议论文集》，2016年。[14] S. Snodgrass 和 A. Sarkar，“通过示例驱动的二叉空间划分和变分自编码器，结合草图实现多领域关卡生成与融合”，载于《第15届数字游戏基础国际会议论文集》，2020年。[15] S. Snodgrass 和 S. Onta˜n´on，“用于二维电子游戏地图生成的层次化MdMC方法”，载于《AAAI人工智能与互动数字娱乐会议论文集》，第11卷，第1期，2015年。[16] A. Sarkar、Z. Yang 和 S. Cooper，“使用变分自编码器实现游戏间可控关卡融合”，载于AIIDE游戏实验人工智能研讨会，2019年。[17] A. Sarkar、A. Summerville、S. Snodgrass、G. Bentley 和 J. Osborn，“探索通过路径与可供性实现平台游戏间的关卡融合”，载于《AAAI人工智能与互动数字娱乐会议论文集》，2020年。[18] M. Guzdial 和 M. O. Riedl，“概念性游戏扩展”，IEEE《游戏汇刊》，第14卷，第1期，第93–106页，2021年。[19] A. Sarkar、Z. Yang 和 S. Cooper，“条件关卡生成与游戏混合”，载于AIIDE游戏实验人工智能研讨会，2020年。[20] A. Sarkar 和 S. Cooper，“使用条件变分自编码器实现地牢与平台游戏关卡的融合和生成”，载于《IEEE游戏会议（CoG）论文集》，2021年。[21] J. Permar 和 B. Magerko，“通过概念融合生成交互式叙事的认知脚本”，载于《第九届人工智能与互动数字娱乐会议》，2013年。[22] S. Snodgrass 和 S. Onta˜n´on，“使用马尔可夫链生成地图的实验”，载于《数字游戏基础国际会议论文集》，2014年。[23] F. Buschmann、R. Meunier、H. Rohnert、P. Sommerlad 和 M. Stal，《面向模式的软件架构：模式体系》。约翰·威立父子出版公司，2008年，第1卷。[24] C. Alexander，《模式语言：城镇、建筑与建造》。牛津大学出版社，1977年。[25] S. Bj¨ork、S. Lundgren 和 J. Holopainen，“游戏设计模式”，载于《数字游戏研究会议论文集》，2003年。[26] B. Kreimeier。（2002年3月）游戏设计模式的价值。 [在线]。网址：https://www.gamedeveloper.com/design/the-case-for-game-design-patterns [27] S. Dahlskog 和 J. Togelius，“模式与程序化内容生成：重新审视世界1-1中的马力欧”，载于《第一届游戏设计模式研讨会论文集》，2012年，第1–8页。[28] K. Hullett 和 J. Whitehead，“第一人称射击游戏关卡中的设计模式”，载于《第五届数字游戏基础国际会议论文集》。计算机协会，2010年，第78–85页。[29] A. Khalifa、F. de Mesentier Silva 和 J. Togelius，“二维游戏中的关卡设计模式”，载于《IEEE游戏会议（CoG）论文集》，2019年。[30] B. Mears 和 J. Zhu，“叙事驱动型游戏中无台词玩家角色的设计模式”，载于《第12届数字游戏基础国际会议论文集》，2017年。

[31] K. Hullett，“第一人称射击游戏关卡中设计模式与设计者意图之间的因果关系”，载于《第二届游戏设计模式研讨会（DPG）论文集》，第13卷，2013年。[32] A. Summerville、J. Osborn、C. Holmg˚ard 和 D. W. Zhang，“通过交互式观察自动识别机制：跳跃”，载于《第十二届数字游戏基础国际会议论文集》，2017年，第1–10页。[33] M. J. Nelson 和 M. Mateas，“迈向自动化游戏设计”，载于意大利人工智能协会大会。Springer，2007年，第626–637页。[34] J. Gow 和 J. Corneli，“通过概念融合生成新颖游戏”，载于《AAAI人工智能与互动数字娱乐会议论文集》，2015年。[35] M. J. Guzdial 和 M. O. Riedl，“通过机器学习实现程序化内容生成的组合创造力”，载于《第32届AAAI人工智能会议研讨会》，2018年。[36] S. Snodgrass，“通过示例驱动的二叉空间划分，从草图生成关卡”，载于《AAAI人工智能与互动数字娱乐会议论文集》，第15卷，第1期，2019年，第73–79页。[37] A. Sarkar 和 S. Cooper，“tile2tile：学习用于平台游戏风格迁移的游戏滤镜”，载于《AAAI人工智能与互动数字娱乐会议论文集》，2022年。[38] S. Beaupre、T. Wiles、S. Briggs 和 G. Smith，“用于多游戏关卡生成的设计模式方法”，载于《第14届AAAI人工智能与交互式数字娱乐会议论文集》，2018年。[39] T. Machado、D. Gopstein、A. Nealen 和 J. Togelius，“Pitako——在Cicero中推荐游戏设计元素”，载于《IEEE游戏会议（CoG）论文集》。IEEE，2019年，第1–8页。[40] F. Zhuang、Z. Qi、K. Duan、D. Xi、Y. Zhu、H. Zhu、H. Xiong 和 Q. He，“迁移学习综合综述”，《IEEE论文集》，第109卷，第1期，第43–76页，2020年。[41] S. J. Pan、I. W. Tsang、J. T. Kwok 和 Q. Yang，“通过迁移成分分析实现领域适应”，IEEE《神经网络汇刊》，第22卷，第2期，第199–210页，2010年。[42] D. S. Chaplot、G. Lample、K. M. Sathyendra 和 R. Salakhutdinov，“三维环境中的迁移深度强化学习：一项实证研究”，载于NeurIPS深度强化学习研讨会，2016年。[43] A. A. Rusu、N. C. Rabinowitz、G. Desjardins、H. Soyer、J. Kirkpatrick、K. Kavukcuoglu、R. Pascanu 和 R. Hadsell，“渐进式神经网络”，arXiv预印本 arXiv:1606.04671，2016年。[44] K. Kansky、T. Silver、D. A. M´ely、M. Eldawy、M. L´azaro-Gredilla、X. Lou、N. Dorfman、S. Sidor、S. Phoenix 和 D. George，“模式网络：基于直觉物理生成式因果模型的零样本迁移”，载于《第34届国际机器学习会议论文集》第70卷。JMLR.org，2017年，第1809–1818页。[45] A. Melnik、S. Fleer、M. Schilling 和 H. Ritter，“用于Atari球类游戏的功能学习模块”，载于ICLR研讨会专题，2018年。[46] A. Mittel 和 P. S. Munukutla，“使用竞争式强化学习实现Atari游戏间的视觉迁移”，载于《2019年IEEE/CVF计算机视觉与模式识别会议研讨会（CVPRW）》，2019年，第499–501页。[47] S. Gamrian 和 Y. Goldberg，“通过图像到图像转换，为相关强化学习任务实现迁移学习”，载于《国际机器学习会议》。PMLR，2019年，第2063–2072页。[48] S. W. Kim、Y. Zhou、J. Philion、A. Torralba 和 S. Fidler，“学习用GameGAN模拟动态环境”，载于《IEEE/CVF计算机视觉与模式识别会议论文集》，2020年，第1231–1240页。[49] S. Colton 和 G. A. Wiggins，“计算创造力：最后的前沿？”，载于《欧洲人工智能会议论文集》。IOS出版社，2012年，第21–26页。[50] A. Liapis、G. N. Yannakakis 和 J. Togelius，“计算游戏创造力”，载于《国际计算创造力会议论文集》，2014年。[51] M. A. Boden，“创造力”，载于《人工智能》。爱思唯尔，1996年，第267–291页。[52] G. Fauconnier 和 M. Turner，“概念整合网络”，《认知科学》，第22卷，第2期，第133–187页，1998年。[在线]。网址：http://onlinelibrary.wiley.com/doi/10.1207/s15516709cog2202 1/pdf[53] C. Gagn´e 和 E. Shoben，“主题关系对修饰语—名词组合理解的影响”，《实验心理学杂志：学习、记忆与认知》，第23卷，第71–87页，1997年。[54] J. Lehman 和 K. O. Stanley，“新颖性搜索与目标的问题”，载于《遗传编程理论与实践》第九卷。Springer，2011年，第37–56页。

[55] D. Gravina、A. Liapis 和 G. Yannakakis，“惊喜搜索：超越目标与新颖性”，载于《遗传与进化计算会议论文集》，2016年，第677–684页。[56] A. Summerville、A. Sarkar、S. Snodgrass 和 J. Osborn，“从混合平台游戏关卡中提取物理特性”，载于 AIIDE 游戏实验人工智能研讨会，2020年。[57] M. A. Boden，“创造力的计算机模型”，《AI Magazine》，第30卷第3期，第23–23页，2009年。[58] S. Ontan´on 和 E. Plaza，“融合体：一种组合多个案例解决方案的形式化方法”，载于《国际基于案例推理会议论文集》。Springer，2010年，第257–271页。[59] K. Badie 和 M. T. Mahmoudi，“基于案例中组件之间语义关系的基于案例推理组合式适应”，载于2017年IEEE智能系统与应用创新国际会议（INISTA），2017年，第449–454页。[60] T. Schaul，“一种可扩展的电子游戏描述语言”，IEEE《计算智能与游戏人工智能汇刊》，第6卷第4期，第325–331页，2014年。[61] 科乐美，“Frogger”，游戏 [街机版], 1981。[62] 任天堂，《塞尔达传说》，游戏 [任天堂娱乐系统], 1986。[63] D. Kingma 和 M. Welling，“自编码变分贝叶斯”，载于《第2届国际学习表征会议（ICLR）论文集》，2013年。[64] 任天堂，《超级马力欧兄弟》，游戏 [任天堂娱乐系统], 1985。[65] ——，“Kid Icarus”，游戏 [任天堂娱乐系统], 1986。[66] ——，“银河战士”，游戏 [任天堂娱乐系统], 1986。[67] 卡普空，《洛克人》，游戏 [任天堂娱乐系统], 1987。[68] 科乐美，《恶魔城》，游戏，1986年。[69] 特库摩，《忍者龙剑传》，游戏，1988年。[70] H. Laboratory，《星之卡比 梦之泉物语》，游戏 [任天堂娱乐系统], 1993。[71] T. Machado、D. Gopstein、A. Wang、O. Nov、A. Nealen 和 J. Togelius，“评估辅助新手游戏设计师的推荐系统”，载于《AAAI人工智能与互动数字娱乐会议论文集》，第15卷第1期，2019年，第167–173页。[72] L. A. Gatys、A. S. Ecker 和 M. Bethge，“艺术风格的神经算法”，arXiv预印本 arXiv:1508.06576，2015年。[73] Y. Jing、Y. Yang、Z. Feng、J. Ye、Y. Yu 和 M. Song，“神经风格迁移：综述”，《IEEE可视化与计算机图形学汇刊》，2019年。[74] D. Perez-Liebana、J. Liu、A. Khalifa、R. D. Gaina、J. Togelius 和 S. M. Lucas，“通用电子游戏人工智能：用于评估智能体、游戏和内容生成算法的多轨框架”，IEEE《游戏汇刊》，第11卷第3期，第195–214页，2019年。[75] 卡普空，《洛克人X》，游戏 [超级任天堂], 1993。[76] 任天堂，《超级银河战士》，游戏 [超级任天堂], 1994。[77] R. Agrawal、R. Srikant 等，“挖掘关联规则的快速算法”，载于《第20届超大型数据库国际会议（VLDB）论文集》，第1215卷，1994年，第487–499页。[78] G. Smith 和 J. Whitehead，“分析关卡生成器的表达范围”，载于《2010年游戏程序化内容生成研讨会论文集》，2010年。[79] A. Summerville，“拓展表达范围：程序化内容生成的评估方法”，载于《第14届AAAI人工智能与交互式数字娱乐会议论文集》，2018年。[80] G. N. Yannakakis、A. Liapis 和 C. Alexopoulos，“混合主动式协同创作”，载于《数字游戏基础国际会议论文集》，2014年。[81] M. Guzdial、N. Liao 和 M. Riedl，“通过机器学习进行协同创作式关卡设计”，AIIDE游戏实验人工智能研讨会，2018年。[82] F. Schubert、M. Awiszus 和 B. Rosenhahn，“Toad-GAN：一种用于基于令牌的游戏少样本关卡生成的灵活框架”，IEEE《游戏汇刊》，2021年。[83] J. Schrum、J. Gutierrez、V. Volz、J. Liu、S. Lucas 和 S. Risi，“在生成对抗网络的潜在关卡设计空间中进行交互式进化与探索”，载于《遗传与进化计算会议论文集》，2020年，第148–156页。[84] D. Bhaumik、A. Khalifa 和 J. Togelius，“Lode Encoder：人工智能约束的协同创作”，载于《IEEE游戏会议（CoG）论文集》，2021年。[85] D. Smith，“Lode Runner”，游戏，1983年。[86] A. Sarkar 和 S. Cooper，“迈向基于创造性机器学习的游戏设计（GDCML）”，载于《IEEE游戏会议（CoG）论文集》，2020年。[87] B. Shneiderman，“创造力支持工具：加速发现与创新”，《ACM通讯》，第50卷，32，2007 12，20第。