



(12) 发明专利申请

(10) 申请公布号 CN 112783781 A

(43) 申请公布日 2021.05.11

(21) 申请号 202110122283.7

(22) 申请日 2021.01.28

(71) 申请人 网易(杭州)网络有限公司

地址 310052 浙江省杭州市滨江区长河街
道网商路599号4幢7层

(72) 发明人 胡玥 宋研 吴悦晨 陈赢峰

范长杰 胡志鹏

(74) 专利代理机构 北京超凡宏宇专利代理事务
所(特殊普通合伙) 11463

代理人 钟扬飞

(51) Int. Cl.

G06F 11/36 (2006.01)

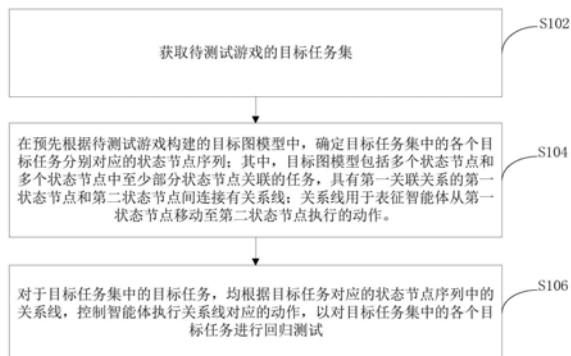
权利要求书2页 说明书15页 附图6页

(54) 发明名称

游戏测试方法、装置、电子设备及存储介质

(57) 摘要

本发明提供了游戏测试方法、装置、电子设备及存储介质。其中,该方法包括:获取待测试游戏的目标任务集;在预先根据待测试游戏构建的目标图模型中,确定目标任务集中的各个目标任务分别对应的状态节点序列;其中,目标图模型包括多个状态节点和多个状态节点中至少部分状态节点关联的任务,具有第一关联关系的第一状态节点和第二状态节点间连接有关系线;关系线用于表征智能体从第一状态节点移动至第二状态节点执行的动作;对于目标任务集中的目标任务,均根据目标任务对应的状态节点序列中的关系线,控制智能体执行关系线对应的动作,以对目标任务集中的各个目标任务进行回归测试,提高了游戏测试效率,从而提高了游戏质量和用户游戏体验度。



1. 一种游戏测试方法,其特征在于,所述方法包括:

获取待测试游戏的目标任务集;

在预先根据所述待测试游戏构建的目标图模型中,确定所述目标任务集中的各个目标任务分别对应的状态节点序列;其中,所述目标图模型包括多个状态节点和多个所述状态节点中至少部分状态节点关联的任务,具有第一关联关系的第一状态节点和第二状态节点间连接有关系线;所述关系线用于表征智能体从所述第一状态节点移动至所述第二状态节点执行的动作;

对于所述目标任务集中的目标任务,均根据所述目标任务对应的状态节点序列中的关系线,控制所述智能体执行所述关系线对应的动作,以对所述目标任务集中的各个目标任务进行回归测试。

2. 根据权利要求1所述的游戏测试方法,其特征在于,所述目标图模型的训练过程包括:

获取所述待测试游戏的任务训练集;其中,所述任务训练集包括多个训练任务;

控制所述待测试游戏中的智能体完成每个所述训练任务,并记录完成每个所述训练任务对应的状态节点;

在原始图模型中建立每个所述训练任务与记录的所述状态节点之间的关联关系,得到所述目标图模型;其中,所述原始图模型包括所述智能体在所述待测试游戏中的多个状态节点,具有关联关系的第一状态节点和第二状态节点间连接有关系线;所述关系线用于表征所述智能体从所述第一状态节点移动至所述第二状态节点执行的动作。

3. 根据权利要求2所述的游戏测试方法,其特征在于,所述状态节点配置有每种动作类型对应的动作参数,所述控制所述待测试游戏中智能体完成每个所述训练任务的步骤,包括:

对于处于当前状态节点的所述智能体,确定所述智能体需要执行的当前动作对应的目标动作类型;

从所述当前状态节点中获取所述目标动作类型对应的动作参数;

控制所述智能体按照所述目标动作类型和所述目标动作类型对应的动作参数执行所述当前动作,以使所述智能体从所述当前状态节点移动至下一个状态节点。

4. 根据权利要求3所述的游戏测试方法,其特征在于,所述确定所述智能体需要执行的当前动作对应的目标动作类型的步骤,包括:

如果所述智能体需要执行的当前动作对应多个动作类型,按照所述动作类型对应的置信系数对多个所述动作类型排序;

根据多个所述动作类型的排序结果选择所述当前动作对应的目标动作类型。

5. 根据权利要求4所述的游戏测试方法,其特征在于,所述确定所述智能体需要执行的当前动作对应的目标动作类型的步骤之前,所述方法还包括:

应用预先训练好的语言模型确定所述训练任务中每种所述动作类型的置信系数。

6. 根据权利要求5所述的游戏测试方法,其特征在于,所述语言模型配置有衰减因子,所述应用预先训练好的语言模型确定所述训练任务中每种所述动作类型的置信系数的步骤,包括:

将所述训练任务输入预先训练好的语言模型,得到所述语言模型输出的所述训练任务

中每种所述动作类型的置信系数;或者,

响应针对于所述衰减因子的调整操作,调整所述语言模型的衰减因子,将所述训练任务输入调整所述衰减因子后的语言模型,得到所述语言模型输出的所述训练任务中每种所述动作类型的置信系数。

7. 根据权利要求1所述的游戏测试方法,其特征在于,所述在预先根据所述待测试游戏构建的目标图模型中,确定所述目标任务集中的各个目标任务分别对应的状态节点序列的步骤,包括:

控制所述智能体完成每个所述目标任务,并记录每个所述目标任务对应的状态节点;

在所述目标图模型中查找每个所述目标任务对应的状态节点,得到所述目标任务集中的各个目标任务分别对应的状态节点序列。

8. 根据权利要求1所述的游戏测试方法,其特征在于,所述根据所述目标任务对应的状态节点序列中的关系线,控制所述智能体执行所述关系线对应的动作的步骤,包括:

基于所述目标任务对应的初始状态节点序列中的关系线,生成目标状态节点序列;其中,所述目标状态节点序列为所述初始状态节点序列的子集;

控制所述智能体执行所述目标状态节点序列中的关系线对应的动作。

9. 根据权利要求1所述的游戏测试方法,其特征在于,所述方法还包括:

响应于所述目标任务集测试完成,生成测试报告,并将所述测试报告进行显示。

10. 一种游戏测试装置,其特征在于,所述装置包括:

获取模块,用于获取待测试游戏的目标任务集;

确定模块,用于在预先根据所述待测试游戏构建的目标图模型中,确定所述目标任务集中的各个目标任务分别对应的状态节点序列;其中,所述目标图模型包括多个状态节点和多个所述状态节点中至少部分状态节点关联的任务,具有第一关联关系的第一状态节点和第二状态节点间连接有关系线;所述关系线用于表征智能体从所述第一状态节点移动至所述第二状态节点执行的动作;

回归测试模块,用于对于所述目标任务集中的目标任务,均根据所述目标任务对应的状态节点序列中的关系线,控制所述智能体执行所述关系线对应的动作,以对所述目标任务集中的各个目标任务进行回归测试。

11. 一种电子设备,包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现上述权利要求1-9任一项所述的游戏测试方法的步骤。

12. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器运行时执行上述权利要求1-9任一项所述的游戏测试方法的步骤。

游戏测试方法、装置、电子设备及存储介质

技术领域

[0001] 本发明涉及游戏技术领域,尤其是涉及游戏测试方法、装置、电子设备及存储介质。

背景技术

[0002] 近几年,游戏产业蓬勃发展,市面上有各种各样的游戏以及庞大的玩家群体。面对如此多的玩家,游戏的质量得到保证是非常有必要的。游戏中的漏洞bug会严重破坏玩家的游戏体验以及对公司造成巨大的经济损失。因此,游戏公司会有专门的质量保证团队以对上线的游戏进行测试,查找是否有bug,保证游戏质量。

[0003] 游戏测试中比较重要的任务回归测试是当游戏版本迭代更新的时候,需要对游戏中的任务系统进行测试,以此来保证所有的任务仍然是可以完成的,从而没有由于某些数值或者功能的改变而发生变化。现有的测试方法主要是测试人员将游戏中的任务如主线任务和支线任务等都玩一遍,如果可以完成任务则说明这个任务是没有问题的。由于测试的过程比较重复枯燥,且,测试任务量巨大,因此,现有的人工测试方法将耗费大量的人力和时间,影响了游戏的测试效率,从而难以有效的保证游戏的质量。

发明内容

[0004] 有鉴于此,本发明的目的在于提供游戏测试方法、装置、电子设备及存储介质,以缓解上述问题,提高了游戏的测试效率。

[0005] 第一方面,本发明实施例提供了一种游戏测试方法,该方法包括:获取待测试游戏的目标任务集;在预先根据待测试游戏构建的目标图模型中,确定目标任务集中的各个目标任务分别对应的状态节点序列;其中,目标图模型包括多个状态节点和多个状态节点中至少部分状态节点关联的任务,具有第一关联关系的第一状态节点和第二状态节点间连接有关系线;该关系线用于表征智能体从第一状态节点移动至第二状态节点执行的动作;对于目标任务集中的目标任务,均根据目标任务对应的状态节点序列中的关系线,控制智能体执行关系线对应的动作,以对目标任务集中的各个目标任务进行回归测试。

[0006] 结合第一方面,本发明实施例提供了第一方面的第一种可能的实施方式,其中,上述目标图模型的训练过程包括:获取待测试游戏的任务训练集;其中,任务训练集包括多个训练任务;控制待测试游戏中的智能体完成每个训练任务,并记录完成每个训练任务对应的状态节点;在原始图模型中建立每个训练任务与记录的状态节点之间的关联关系,得到目标图模型;其中,原始图模型包括智能体在待测试游戏中的多个状态节点,具有关联关系的第一状态节点和第二状态节点间连接有关系线;关系线用于表征智能体从第一状态节点移动至第二状态节点执行的动作。

[0007] 结合第一方面的第一种可能的实施方式,本发明实施例提供了第一方面的第二种可能的实施方式,其中,上述状态节点配置有每种动作类型对应的动作参数,上述控制待测试游戏中智能体完成每个训练任务的步骤,包括:对于处于当前状态节点的智能体,确定智

能体需要执行的当前动作对应的目标动作类型;从当前状态节点中获取目标动作类型对应的动作参数;控制智能体按照目标动作类型和目标动作类型对应的动作参数执行当前动作,以使智能体从当前状态节点移动至下一个状态节点。

[0008] 结合第一方面的第二种可能的实施方式,本发明实施例提供了第一方面的第三种可能的实施方式,其中,上述确定智能体需要执行的当前动作对应的目标动作类型的步骤,包括:如果智能体需要执行的当前动作对应多个动作类型,按照动作类型对应的置信系数对多个动作类型排序;根据多个动作类型的排序结果选择当前动作对应的目标动作类型。

[0009] 结合第一方面的第三种可能的实施方式,本发明实施例提供了第一方面的第四种可能的实施方式,其中,上述确定智能体需要执行的当前动作对应的目标动作类型的步骤之前,该方法还包括:应用预先训练好的语言模型确定训练任务中每种动作类型的置信系数。

[0010] 结合第一方面的第四种可能的实施方式,本发明实施例提供了第一方面的第五种可能的实施方式,其中,上述语言模型配置有衰减因子,上述应用预先训练好的语言模型确定训练任务中每种动作类型的置信系数的步骤,包括:将训练任务输入预先训练好的语言模型,得到语言模型输出的训练任务中每种动作类型的置信系数;或者,响应针对于衰减因子的调整操作,调整语言模型的衰减因子,将训练任务输入调整衰减因子后的语言模型,得到语言模型输出的训练任务中每种动作类型的置信系数。

[0011] 结合第一方面,本发明实施例提供了第一方面的第六种可能的实施方式,其中,上述在预先根据待测试游戏构建的目标图模型中,确定目标任务集中的各个目标任务分别对应的状态节点序列的步骤,包括:控制智能体完成每个目标任务,并记录每个目标任务对应的状态节点;在目标图模型中查找每个目标任务对应的状态节点,得到目标任务集中的各个目标任务分别对应的状态节点序列。

[0012] 结合第一方面,本发明实施例提供了第一方面的第七种可能的实施方式,其中,上述根据目标任务对应的状态节点序列中的关系线,控制智能体执行关系线对应的动作的步骤,包括:基于目标任务对应的初始状态节点序列中的关系线,生成目标状态节点序列;其中,目标状态节点序列为初始状态节点序列的子集;控制智能体执行目标状态节点序列中的关系线对应的动作。

[0013] 结合第一方面,本发明实施例提供了第一方面的第八种可能的实施方式,其中,该方法还包括:响应于目标任务集测试完成,生成测试报告,并将测试报告进行显示。

[0014] 第二方面,本发明实施例还提供一种游戏测试装置,该装置包括:获取模块,用于获取待测试游戏的目标任务集;确定模块,用于在预先根据待测试游戏构建的目标图模型中,确定目标任务集中的各个目标任务分别对应的状态节点序列;其中,目标图模型包括多个状态节点和多个状态节点中至少部分状态节点关联的任务,具有第一关联关系的第一状态节点和第二状态节点间连接有关系线;关系线用于表征智能体从第一状态节点移动至第二状态节点执行的动作;回归测试模块,用于对于目标任务集中的目标任务,均根据目标任务对应的状态节点序列中的关系线,控制智能体执行关系线对应的动作,以对目标任务集中的各个目标任务进行回归测试。

[0015] 第三方面,本发明实施例提供了一种电子设备,包括存储器、处理器及存储在存储

器上并可在处理器上运行的计算机程序,处理器执行计算机程序时实现上述第一方面的游戏测试方法的步骤。

[0016] 第四方面,本发明实施例提供了一种计算机可读存储介质,计算机可读存储介质上存储有计算机程序,计算机程序被处理器运行时执行上述第一方面的游戏测试方法的步骤。

[0017] 本发明实施例带来了以下有益效果:

[0018] 本发明实施例提供了游戏测试方法、装置、电子设备及存储介质,通过预先根据待测试游戏构建的目标图模型,确定各个待测试的目标任务对应的状态节点序列,以便控制智能体执行每个目标任务对应的状态节点序列中的关系线对应的动作,从而实现对各目标任务回归测试,与现有的人工回归测试方法相比,减少了人力和时间消耗,提高了任务回归测试效率,进而提高了游戏的质量和用户游戏体验度。

[0019] 本发明的其他特征和优点将在随后的说明书中阐述,并且,部分地从说明书中变得显而易见,或者通过实施本发明而了解。本发明的目的和其他优点在说明书以及附图中所特别指出的结构来实现和获得。

[0020] 为使本发明的上述目的、特征和优点能更明显易懂,下文特举较佳实施例,并配合所附附图,作详细说明如下。

附图说明

[0021] 为了更清楚地说明本发明具体实施方式或现有技术中的技术方案,下面将对具体实施方式或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施方式,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0022] 图1为本发明实施例提供了一种游戏测试方法的流程图;

[0023] 图2为本发明实施例提供了一种原始图模型的结构示意图;

[0024] 图3为本发明实施例提供了一种智能体完成任务的示意图;

[0025] 图4为本发明实施例提供了一种SIAT方法的测试框架;

[0026] 图5为本发明实施例提供了一种游戏测试方法的原理图;

[0027] 图6为本发明实施例提供的另一种游戏测试方法的流程图;

[0028] 图7为本发明实施例提供了一种平均提升的实验结果示意图;

[0029] 图8-a为本发明实施例提供了一种迷你游戏上的语言模型的搜索时间的实验结果示意图;

[0030] 图8-b为本发明实施例提供了一种倩女幽魂手游上的语言模型的搜索时间的实验结果示意图;

[0031] 图8-c为本发明实施例提供了一种逆水寒端游上的语言模型的搜索时间的实验结果示意图;

[0032] 图9为本发明实施例提供了一种游戏测试装置的示意图;

[0033] 图10为本发明实施例提供了一种电子设备的结构示意图。

具体实施方式

[0034] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合附图对本发明的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0035] 在游戏测试中，一个任务的完整测试过程包括以下步骤：首先玩家接取任务，并根据该任务对应的任务栏描述寻找任务相关的NPC (Non Player Character, 非玩家角色)；寻找到待测试任务相关的NPC之后，与NPC进行交谈，并完成任务，然后接取下一个待测试任务。在实际测试中，游戏中待测试的任务数量较多，导致测试任务量巨大，耗费大量的人力和时间，从而影响了游戏的测试效率。此外，还可以通过测试人员编写脚本来完成游戏任务回归测试，虽然可以缓解测试所需的时间和人力，但针对不同任务类型的回归测试，缺少通用性，如每一个任务都需要编写一个特定的脚本；当任务有变化时则需要修改脚本，或者有新任务来的时候也要重新编写脚本，导致任务回归测试效率较低，且，这是一个需要长期维护的过程，影响了游戏的质量和用户的游戏体验度。

[0036] 针对上述问题，本发明实施例提供了游戏测试方法、装置、电子设备及存储介质，通过预先根据待测试游戏构建的目标图模型，确定各个待测试的目标任务对应的状态节点序列，以便控制智能体执行每个目标任务对应的状态节点序列中的关系线对应的动作，从而实现对各个目标任务的回归测试，与现有的人工回归测试方法相比，减少了人力和时间消耗，提高了任务回归测试效率，进而提高了游戏的质量和用户游戏体验度。

[0037] 为便于对本实施例进行理解，下面首先对本发明实施例提供的一种游戏测试方法进行详细介绍。

[0038] 本发明实施例提供了一种游戏测试方法，执行主体为服务器，如图1所示，该方法包括以下步骤：

[0039] 步骤S102，获取待测试游戏的目标任务集。

[0040] 其中，目标任务集中包括待测试游戏的多个目标任务，每个目标任务通过对应的任务栏描述表示，如任务栏描述为“找到NPC1并询问最近是否发生了一些奇怪的事情”，对应的目标任务为寻找NPC1，并与NPC1交流，以确定最近是否发生了一些奇怪的事情，因此，每个目标任务对应的任务栏描述包含完成该目标任务的动作序列。

[0041] 步骤S104，在预先根据待测试游戏构建的目标图模型中，确定目标任务集中的各个目标任务分别对应的状态节点序列；其中，目标图模型包括多个状态节点和多个状态节点中至少部分状态节点关联的任务，具有第一关联关系的第一状态节点和第二状态节点间连接有关系线；关系线用于表征智能体从第一状态节点移动至第二状态节点执行的动作。

[0042] 具体地，上述目标图模型的训练过程包括：首先获取待测试游戏的任务训练集；其中，任务训练集包括多个训练任务；然后，控制待测试游戏中的智能体完成每个训练任务，并记录完成每个训练任务对应的状态节点；最后在原始图模型中建立每个训练任务与记录的状态节点之间的关联关系，得到目标图模型，从而将训练任务与原始图模型关联起来，加速了确定目标图模型中目标任务对应的状态节点序列。

[0043] 其中，原始图模型包括智能体在待测试游戏中的多个状态节点，如图2中的圆形节点，具有关联关系的两个状态节点间连接有关系线，例如具有关联关系的第一状态节点和

第二状态节点间连接有关系线;该关系线用于表征智能体从第一状态节点移动至第二状态节点执行的动作,其中,关系线的箭头方向为智能体执行的动作指示方向。因此,在原始图模型中,状态节点组成了智能体在游戏环境中的状态空间,状态节点间的关系线则组成了智能体在游戏环境中的动作空间。

[0044] 在实际应用中,可以将游戏任务测试的过程转化为一个确定性的MDP (Markov Decision Process, 马尔科夫决策过程),并根据该确定性的马尔科夫决策问题,构建上述原始图模型。具体地,游戏任务测试的过程可以建模成MDP,将用来测试的智能体看成玩家agent,游戏环境则是environment,智能体可以将获取到的当前环境和自身的一些状态作为state,当前可以执行的动作为action。如图3所示,该智能体像玩家一样在游戏环境中通过与环境互动来完成任务,在完成任务的同时完成了对任务的测试,即这个过程是一个接收到状态,执行动作,并接收下一个状态,直至任务完成为止,因此,游戏任务的回归测试可以看作一个序列决策问题,从而可以将游戏任务的自动回归测试转化为确定性的马尔可夫决策过程。

[0045] 在游戏环境中,智能体在 t 时刻接收到状态信息 $s_t \in S$,其中, S 表示状态空间,即智能体在待测试游戏的游戏环境中的所有状态;状态信息则表示智能体所处的一种状态,该状态从游戏内获取,包含UI (User Interface, 用户界面) 信息、任务信息、环境信息、玩家信息等;UI信息包括窗口和对话框等,任务信息则包含任务id (identity document, 帐号)、任务栏描述和任务完成进度等;环境信息则包含智能体周边的信息如NPC和怪物信息如怪物位置信息等;玩家信息则包括玩家位置信息如智能体位置信息、玩家血量信息以及玩家是否处于对话和/或剧情中等。然后,智能体根据状态信息 s_t 选择一个适当的动作 $a_t \in A$ 在游戏环境中执行,并接收下一个状态信息 s_{t+1} ,直至任务完成;其中, a_t 表示智能体在状态信息 s_t 需要执行的动作, A 表示动作空间。

[0046] 此外,智能体在与游戏环境的互动过程中,不同的行为会获取游戏环境给出的奖励,而智能体的目标就是在游戏环境中不断试错,使得找到的一种策略(即完成任务的一系列的行动)的累积奖励最大。由于每一个动作由两部分组成:动作类型和动作参数,其中,动作类型例如移动、攻击和点击NPC等,动作参数是指与动作类型对应的参数,例如移动这个动作类型的参数可以是具体的位置信息,如NPC的id等,有的动作类型的动作参数还可以是缺省(默认值),例如点击对话框这个动作类型则不需要具体的参数。当执行正确的动作类型后,智能体的状态是会发生变化的,如果 s_{t+1} 和 s_t 一样,则说明智能体是在同样的状态下尝试其他动作,此时智能体继续尝试其他动作直至状态发生改变。

[0047] 此外,动作是基于状态产生的,其中动作类型是固定的,动作参数从状态里提取,存在一个动作类型可以和多个动作参数组合产生多个动作的情况,因此,一个状态下的动作空间很大,随机探索或者按顺序一个一个尝试动作是非常消耗时间的,一般需要很久才尝试到一个正确的动作。如果 s_{t+1} 和 s_t 不一样,则说明智能体的状态发生变化,即智能体执行了正确的动作,此时智能体继续收下一个状态并尝试执行动作,反复执行这个循环过程直至任务完成。当任务完成,则说明已经找到一条可以完成任务的动作序列,因此,智能体是在巨大的状态-动作空间中确定最优策略 $S \rightarrow A$,使得智能体能够快速完成任务。

[0048] 现有方法中也有采用人工智能进行自动化任务回归测试,例如采用RL (Reinforcement Learning, 强化学习)的方式构建一个玩游戏的智能体,该智能体在游戏

环境中通过互动来测试任务是否可完成。以及,还可以通过强化学习算法指导智能体如何对游戏环境进行探索,如DQN (Deep Q Network,深度Q网络)、DDPG (Deep Deterministic Policy Gradient,深度确定性策略梯度算法)和强化学习Q-learning等算法,虽然解脱了部分人力,但是,由于游戏系统的复杂性,智能体在一个状态下可以尝试的动作很多,直到找到完成任务的动作组合,这个过程是非常耗时的,导致任务回归测试的效率较低。

[0049] 而本发明实施例中则是采用SIAT (Self-Improving Automated Testing,自提升的自动化测试)方法,该方法包括目标图模型的构建阶段和目标图模型的规划和利用阶段。基于如图4所示的SIAT方法的测试框架,下面详细介绍SIAT方法。

[0050] 其中,在目标图模型的构建阶段,主要是控制智能体在游戏环境中尝试各种动作直至训练任务的完成,以得到目标图模型。具体的,控制智能体完成每个训练任务,并记录完成每个训练任务对应的状态节点,以在原始图模型中建立每个训练任务与记录的状态节点之间的关联关系,得到目标图模型,从而实现目标图模型中多个状态节点中至少部分状态节点关联有任务,在目标任务的回归测试中,可以缩短智能体在动作空间的探索过程,达到加速回归测试的效果。

[0051] 在游戏环境中,智能体根据找到完成任务的所有路径得出最优策略。由于该过程是一个确定性的MDP,在游戏环境中探索的过程中会动态地构建一个有向图,其中,图的节点表示状态节点,有向边表示动作。由于状态-动作空间巨大,很难构建完整的图,因此,需要构建一个能够完成任务的图模型,如图4中构建的目标图模型。

[0052] 在目标图模型的训练过程中,原始图模型的状态节点配置有每种动作类型对应的动作参数,在控制待测试游戏中智能体完成每个训练任务的过程中,对于处于当前状态节点的智能体,确定智能体需要执行的当前动作对应的目标动作类型;从当前状态节点中获取目标动作类型对应的动作参数;控制智能体按照目标动作类型和目标动作类型对应的动作参数执行当前动作,以使智能体从当前状态节点移动至下一个状态节点。为了快速确定智能体需要执行的当前动作对应的目标动作类型,本发明实施例还基于计算方法和文本信息指导方法确定目标动作类型,即对动作空间的探索加速是在动作类型层面上的优化和加速,最后基于目标动作类型对应的动作参数进行组合构成智能体需要执行的当前动作。

[0053] 其中,基于计数方是基于动作在动作空间中被选择的次数,由于原始图模型中并没有任务和包含的动作序列的对应的关系,因此,通过计数方法进行训练,以便产生训练数据,从而确保每个动作都将被选择,避免重复选择已经选过的动作。具体地,选择动作的策略如下:

$$[0054] \quad \pi(a_t | s_t) = \arg \max_a U(s_t, a) \quad (1)$$

[0055] 其中, s_t 表示t时刻的状态信息即t时刻的状态节点, $\pi(a_t | s_t)$ 表示t时刻的状态节点 s_t 处选择动作 a_t 的动作选择策略, a_t 表示状态节点 s_t 处需要执行的动作, a 表示动作, $U(s_t, a)$ 表示动作 a 在状态节点 s_t 的得分。

[0056] 具体地, $U(s, a)$ 可以根据下式确定:

$$[0057] \quad U(s, a) = \begin{cases} P(s, a) & \sum_b N(s, b) = 0 \\ P(s, a) \sqrt{\frac{\sum_b N(s, b)}{1 + N(s, a)}} & otherwise \end{cases} \quad (2)$$

[0058] 其中, $U(s, a)$ 表示动作 a 在状态节点 s 的得分, $P(s, a)$ 表示在状态节点 s 选择动作 a 的概率, $N(s, a)$ 表示动作 a 在状态节点 s 被选择的次数, $N(s, b)$ 表示动作 b 在状态节点 s 被选择的次数, b 表示状态节点 s 处所有的动作。

[0059] 以及, 根据下式计算 $P(s, a)$:

$$[0060] \quad P(s, a) = 1/M \quad (3)$$

[0061] 其中, $P(s, a)$ 表示在状态节点 s 选择动作 a 的概率, M 表示状态节点 s 处所有可执行动作的个数。

[0062] 当智能体完成训练任务后, 将得到完成该训练任务的一系列状态和动作, 如图4中所示的 $(s_1, a_1, s_2, a_2, \dots, s_n, a_n)$, 并在原始图模型中查找得到每个状态对应的状态节点, 以及每个动作对应的关系线, 从而得到的目标图模型中部分状态节点与训练任务具有关联关系。

[0063] 此外, 由于游戏中存在巨大的状态-动作行为空间, 智能体在每个状态查找执行动作是非常低效的, 因此, 还可以通过文本信息指导方法确定智能体在每个状态节点的目标动作类型, 即通过训练任务对应的任务栏描述指导智能体快速的确定正确的目标动作类型。具体地, 根据训练任务对应的任务栏描述给每个动作 $a \in A$ 赋予一个置信系数 $C(s, a)$, 以便智能体优先选择具有高置信系数的动作类型。

[0064] 其中, 在目标图模型的训练过程中, 确定智能体需要执行的当前动作对应的目标动作类型的过程如下: 如果智能体需要执行的当前动作对应多个动作类型, 则按照动作类型对应的置信系数对多个动作类型排序; 并根据多个动作类型的排序结果选择当前动作对应的目标动作类型。例如, 训练任务为: 找到NPC1并询问最近是否发生了一些奇怪的事情, 则智能体在开始执行该训练任务时, 动作类型可能是移动至NPC1, 也可能是点击NPC1, 此时通过比较移动至NPC1的置信系数和点击NPC1的置信系数, 将置信系数比较大的作为智能体对应的目标动作类型。

[0065] 为了确定每种动作类型对应的置信系数, 该还包括: 应用预先训练好的语言模型确定训练任务中每种动作类型的置信系数。在其中一种可能的确定置信系数的方式中, 将训练任务输入预先训练好的语言模型, 得到语言模型输出的训练任务中每种动作类型的置信系数。具体地, 置信系数可以根据下式计算:

$$[0066] \quad P(s, a) = \text{softmax}(C(s, a) + 1/M) \quad (4)$$

[0067] 其中, $P(s, a)$ 表示在状态节点 s 选择动作 a 的概率, M 表示状态节点 s 处所有可执行动作的个数, $C(s, a)$ 表示状态节点 s 处动作 a 的置信系数。

[0068] 上述语言模型模型是一个神经网络, 可以仅仅由几个完全连接的层和一个 softmax 层组成, 也可以是其他的模型。该语言模型的训练数据来自SIAT中的训练任务集, 没有人工注释, 如图4中的训练集。在完成第一轮动作空间探索后, 可以得到每个训练任务对应的动作序列, 作为任务栏描述对应的标签进行语言模型训练, 即一个任务栏描述对应一个动作作为一组样本数据, 任务栏描述是特征, 动作类型是对应的标签。

[0069] 但是, 在实际应用中, 一个游戏任务对应有一个任务栏描述, 智能体需要根据任务栏描述执行多个动作来完成该游戏任务, 即任务栏描述和所需的动作之间的映射并非是一一对一的, 例如一个任务栏描述可以映射多个动作; 此外, 任务栏描述中的误导性文字描述可能会导致智能体执行完全错误的行为, 这些在语言模型训练过程中, 均会给语言模型带来

噪声,导致语言模型输出的置信系数精确较低。

[0070] 为了缓解语言模型中噪音导致的置信系数精度较低的问题,在另一种可能的置信系数确定方式中,上述语言模型配置有衰减因子,响应针对于衰减因子的调整操作,调整语言模型的衰减因子,将训练任务输入调整衰减因子后的语言模型,得到语言模型输出的训练任务中每种动作类型的置信系数。其中,可以根据下式计算置信系数:

$$[0071] \quad P(s, a) = \text{softmax}(C(s, a) \lambda^{N(s, a)} + 1/M) \quad (5)$$

[0072] 其中, $P(s, a)$ 表示在状态节点 s 选择动作 a 的概率, M 表示状态节点 s 处所有可执行动作的个数, $C(s, a)$ 表示状态节点 s 处动作 a 的置信系数, $N(s, a)$ 表示动作 a 在状态节点 s 被选择的次数, λ 表示衰减因子,且,满足 $\lambda \in [0, 1]$ 。

[0073] 因此,通过在语言模型中配置衰减因子,可以降低语言模型多次选择同一动作类型的置信系数,避免了在目标图模型训练过程中智能体一直选择一个动作类型,从而减少目标图模型在实际应用中选择动作类型的时间。

[0074] 综上,对于上述训练过程得到的目标图模型,确定目标任务集中的各个目标任务分别对应的状态节点序列的过程如下:控制智能体完成每个目标任务,并记录每个目标任务对应的状态节点;在目标图模型中查找每个目标任务对应的状态节点,得到目标任务集中的各个目标任务分别对应的状态节点序列,并基于每个目标任务对应的状态节点序列组成的路径对每个目标任务进行回归测试,与现有的人工测试方法相比,提高了回归测试中确定智能体路径的效率,从而加速了目标任务的回归测试。

[0075] 步骤S106,对于目标任务集中的目标任务,均根据目标任务对应的状态节点序列中的关系线,控制智能体执行关系线对应的动作,以对目标任务集中的各个目标任务进行回归测试。

[0076] 在目标图模型中,根据查找到的目标任务的状态节点序列,可以得到完成该目标任务的一条可行的路径,但是在该路径中可能存在多个没有意义的子路径,无意义的子路径是指智能体探索过程中许多无意义的动作,比如(1)移动到与目标任务不相关的NPC上,并点击不相关的NPC进行对话;(2)移动至一个新的位置并很快返回;(3)智能体攻击超过目标任务要求的怪物等,从而在对目标任务进行回归测试时,降低了测试效率。

[0077] 因此,为了提高测试效率,对于确定的各个目标任务分别对应的状态节点序列,还需在目标图模型中根据状态节点序列中的部分状态节点寻找一条最短路径,如图4中基于目标图模型规划得到的从开始至结束的实线路径。具体地,基于目标任务对应的初始状态节点序列中的关系线,生成目标状态节点序列;其中,目标状态节点序列为初始状态节点序列的子集;这里初始状态节点即目标图模型中查找到的目标任务对应的状态节点序列,包括多种状态节点子序列,从而在目标图模型中根据每个目标任务的初始状态节点序列进行路径规划,以得到最优路径作为完成目标任务的最优策略即目标状态节点序列以及目标状态节点序列中每个目标状态节点之间的动作,并通过该最优路径完成目标任务的回归测试,即控制智能体执行目标状态节点序列中的关系线对应的动作。

[0078] 具体地,在对目标任务进行回归测试时,为了实现在游戏版本迭代时目标任务依旧可以完成而不会出现bug,则在目标图模型中寻找与智能体当前状态相同的状态节点,如状态节点 s ,并将状态节点 s 选择动作 a 的概率 $P(s, a)$ 分配给动作空间中对应的动作 a ,以使智能体按照状态节点 s 处选择动作 a 的动作选择策略 $\pi(a|s)$ 选择动作并快速完成测试任务。

以及,在找到最优路径之后,为了增加状态节点下最优动作 a_i 的优先级,设置在状态节点 s 选择动作 a_i 的概率 $P(s, a_i)$ 为1,在状态节点 s 选择其他动作 a 的概率 $P(s, a)$ 则为 $1/M$ 。

[0079] 此外,如果新游戏版本改变了一些内容影响了目标任务相关的地方,那么智能体在完成目标任务时的部分状态对应的状态节点可能不在目标图模型中,或者状态节点在目标图模型中,但规划得到的最优路径却是错误的。此时,对于第一种情况,当前动作空间中所有动作的概率如在状态节点 s 选择动作 a 的概率 $P(s, a)$ 则将根据上述公式5计算;对于第二种情况,当前动作空间中所有动作的概率如在状态节点 s 选择动作 a 的概率 $P(s, a)$ 则将根据目标图模型中的值分配,但是状态节点 s 处选择动作 a 的动作选择策略 $\pi(a_t | s_t)$ 给出的最优动作可能是错误的,由于在状态节点 s 选择其他动作 a 的概率 $P(s, a)$ 为 $1/M$,因此,智能体在一定尝试之后会有概率选择其他动作进行探索。

[0080] 随着SIAT对回归测试的需求越来越多,生成的样本数据也越来越多,其中包含的知识也越来越多。当新的目标任务到来时,根据在目标图模型规划并收集数据后对语言模型进行迭代训练语言模型,以提高语言模型的性能在状态-行动空间中的探索,即SIAT在目标图模型和语言模型的自我监督训练方式下可以得到提高。此外,SIAT为训练语言模型生成数据,并且指导智能体探索动作以完成目标任务,其他类似的结构可能存在协同适应的问题,但是对于SIAT,由于智能体探索结束的唯一标志是完成任务,因此,如果当前目标图模型中智能体执行的动作是错误的,目标任务就不会完成,智能体仍然会选择执行其他动作,直到目标任务完成,故通过目标图模型中的状态节点序列进行路径规划,即便无效的动作具有较高的置信系数,也可以忽略掉,以使智能体按照规划的最优路径对目标任务进行回归测试,从而提高了回归测试的效率。

[0081] 本发明实施例提供的游戏测试方法,通过预先根据待测试游戏构建的目标图模型,确定各个待测试的目标任务对应的状态节点序列,以便控制智能体执行每个目标任务对应的状态节点序列中的关系线对应的动作,从而实现各个目标任务的回归测试,与现有的人工回归测试方法相比,减少了人力和时间消耗,提高了任务回归测试效率,进而提高了游戏的质量和用户游戏体验度。

[0082] 在其中一种可能的实施例中,如图5所示,上述游戏测试方法可以分为训练流程和回归测试流程,其中,在训练流程中,如图5中的上半部分,当训练服务器获取到一个新的游戏任务,该游戏任务包括主线任务和支线任务等,并基于这个游戏任务对原始图模型进行训练,即控制智能体在游戏环境中完成这个游戏任务,得到这个任务包含的状态和节点,并在原始图模型中查找每个状态,得到目标图模型;此时目标图模型中包括该游戏任务对应的多组状态节点以及每个状态节点处执行的动作,即完成这个任务的多条路径。

[0083] 在回归测试流程中,如图5中的下半部分,根据回归测试服务器刚接收到任务时,智能体的状态作为第一个状态,上述游戏任务完成的状态作为最后一个状态,然后在整个目标图模型中进行路径规划,得到一条从起点到终点的最短路径即目标动作序列,之后当游戏版本更新时对该游戏任务进行回归测试时,控制智能体按照最短路径执行,实现了对该游戏任务的回归测试。需要说明的是,上述训练服务器和回归测试服务器可以为同一服务器,也可以为不同服务器,具体可以根据实际情况进行设置。

[0084] 此外,当上述目标任务回归测试完成后,该方法还包括:响应于目标任务集测试完成,生成测试报告,并将测试报告进行显示,以便当待测试游戏的版本进行更新时,开发人

员及时掌握更新后游戏中各个任务的情况,从而保证游戏的质量,进而提高用户的游戏体验度。

[0085] 在图1的基础上,本发明实施例还提供了另一种游戏测试方法,如图6所示,该方法包括以下过程描述:

[0086] (61) 问题建模:将任务回归测试问题转化为一个确定性的马尔科夫决策问题。其中,将用于进行回归测试的智能体可以作为一个agent,游戏世界作为环境,任务情况和环境情况作为状态,当前可以进行的操作如移动,点击npc,杀怪等行为(同时也是玩家可以操作的行为)作为动作;

[0087] (62) 基于任务栏描述构建目标图模型:针对确定性的马尔科夫决策过程问题,构建原始图模型,并基于训练任务对应的任务栏描述对原始图模型进行训练,得到目标图模型;在训练过程中,智能利用训练任务对应的任务栏描述的文本信息对动作空间里的动作赋予不同的分数,然后根据分数的高低进行动作的选择,可以大大缩短在动作空间中的探索过程,以此达到加速的目的,从而缩短了智能体完成训练任务的探索时间,进而缩短了目标图模型的构建时间,当有新的任务需要进行测试时,则在目标图模型构建时就可以缩短时间;

[0088] (63) 基于目标图模型的规划和利用:在上述构建的目标图模型的基础上,采用规划的方法找到一条智能体可以完成任务的最短路径,从而在进行任务回归测试的时候,利用找到的最短路径上的动作序列指导智能体行动,从而可以更快地完成任务。因此,在目标图模型上寻找最短路径有利于后续版本更新迭代后对任务进行回归测试,智能体完成任务的动作序列越短,则完成回归测试越快,从而加速了游戏的测试效率。

[0089] (64) 模拟环境和真实游戏环境的测试结果:即发明人通过在模拟环境和真实游戏环境进行实验,验证了本发明实施例中游戏测试方法的有效性。

[0090] 为了便于理解,这里以一个迷你游戏和两个MMORPG(Massive Multiplayer Online Role-Playing Game,大型多人在线角色扮演游戏)(如倩女幽魂手游和逆水寒端游)进行实验,以评估上述SIAT方法的性能。其中,迷你游戏是M2MORPG的缩小版,由经验丰富的游戏开发人员开发,它是一个很好的游戏测试研究平台,具有双重优势:(1)支持研究者轻松设计自己的游戏任务;(2)支持插入bug,以便验证各种测试方法的有效性。而两个真实游戏则是非常受欢迎的商业MMORPG,它们都有很多日常活跃用户,在这两款游戏中有许多任务,比如主要故事任务,支线故事任务等;同时,这两个真实游戏每周都会有版本更新,从而需要对游戏任务进行大量的回归测试。

[0091] 基于此,发明人设计了两个实验来验证SIAT的有效性和自我提升能力。其中,第一个实验是通过比较上述SIAT方法、随机方法和基线方法,并根据三种方法在相同的任务上的探索时间来验证SIAT方法的有效性;第二个实验则是将不同的语言模型的探索时间进行比较,并通过对相同任务不断增加样本数据来训练,从而验证了当被测试任务数量增加时,语言模型学习的知识越多,SIAT方法获得的性能也越好,两个实验的具体过程如下:

[0092] 实验一:探索时间的对比

[0093] 该实验主要是为了验证SIAT方法确实能够缩短智能体的探索时间,具体地,基于现有的语言模型F(d),在迷你游戏、倩女幽魂手游和逆水寒端游这三种游戏环境中,发明人选择一些新的任务进行实验,并使用SIAT方法、基线方法和随机法三种方法分别进行动作

的选择,直到任务完成,以比较不同方法的性能。

[0094] 其中,对于SIAT方法,衰减因子 λ 设置为0.2, λ 的值对实验结果影响不大,在每个游戏环境中,发明人使用不同的随机种子计算平均结果;同时,发明人为每个任务设置了20分钟的时间阈值,如果任务不能在时间阈值内完成,则该任务将被跳过,具体的实验结果和分析详见表1:

[0095] 表1

游戏	任务数	基线方法	SIAT方法	随机方法	提升
迷你游戏	12	0.519	0.169	0.833	67.4%
倩女幽魂手游	17	572.277	212.162	—	62.9%
逆水寒端游	10	909.191	463.322	—	49.0%

[0097] 由上述表1中记录的三种方法在这三种游戏环境下探索时间的平均值可知,其中,探测时间的测量单位是秒,空白(-)则表示该方法不能在时间阈值内完成任务,最后一列中的值是表示SIAT方法相比基线方法的改进百分比。此外,虽然随机方法使用0.833秒完成了12个迷你游戏任务,但是该方法对于真实对的游戏,则在时间阈值时间内不能完成任务。

[0098] 在上述三种游戏环境下,SIAT方法的探索时间比其他方法都要短;特别地,SIAT方法在迷你游戏中比基线方法改进了67.4%,节省了一半的时间;在另外两个复杂的MMORPG游戏中,SIAT方法比基线方法改进了大约50%,因此,SIAT方法对缩短探索时间起到了一定的作用。

[0099] 实验二:自我提升的验证

[0100] 该实验主要是为了验证SIAT方法的性能随着学习到的语言模型知识的增加而提高。其中,具体的实验设置包括:使用逐渐增加的样本数据对语言模型 $F(d)$ 进行多次训练,并对SIAT方法进行性能评估,以验证SIAT方法是在自我改进。

[0101] 在该实验中,发明人在SIAT方法未测试的任务中随机选择了几组任务(Q_1, Q_2, \dots, Q_m);首先,使用基于计数的方法测试第一组任务 Q_1 ,直到 Q_1 中的所有任务都被智能体完成,然后将 Q_1 的基于目标图模型规划后生成的样本数据 D_1 进行采集,并用这些数据训练语言模型 $F(d)$,得到 F_1 。对于第二组任务 Q_2 ,发明人使用SIAT方法和基线方法分别完成 Q_2 中的任务并计算探索时间,此时,SIAT方法还使用模型 F_1 计算动作的置信系数,以决定智能体执行的动作,并基于目标图模型重新进行规划,得到最短路径,之后同样生成样本数据 D_2 ,最后发明人一起使用 D_1 和 D_2 训练语言模型 $F(d)$,得到模型 F_2 。由于对语言模型 $F(d)$ 的训练是从头开始的,且,训练成本不大,故也可以基于模型 F_1 ,使用 D_2 训练,得到模型 F_2 ,然后用基于模型 F_2 的SIAT方法和基线方法分别探索路径,以完成 Q_2 中的任务,并计算探索时间;以此类推,发明人在两个MMORPG中总共执行了7套任务,在迷你游戏中执行了5套任务。

[0102] 因此,发明人将基于不同的语言模型进行动作探索的方法进行对比,以查看随着模型的改变,探索时间是否减少,性能是否提升。在实际应用中,由于任务组中每个任务的探索时间是不同的,故发明人以任务组的平均提升作为评价标准,从而使结果更加清晰。其中,根据下式计算平均提升:

$$[0103] \quad I_j = \frac{\sum_{i=1}^{|Q_j|} \frac{T_i^{\text{baseline}} - T_i^{\text{SIAT}}}{T_i^{\text{baseline}}}}{|Q_j|} \times 100\% \quad (6)$$

[0104] 其中, I_j 表示 Q_j 任务组的平均改进, T_i^{baseline} 表示基线方法对 Q_j 任务组中第 i 个任务的探索时间, T_i^{SIAT} 表示 SIAT 方法对 Q_j 任务组中第 i 个任务的探索时间。其中, 每个任务组的平均改进用于表征使用不同的语言模型对应的 SIAT 方法相对于基线方法的改进。

[0105] 此外, 发明人还通过基线方法和不同语言模型的 SIAT 方法对相同的任务进行探索, 以比较两种方法的探索时间。对比试验结果如图 7 所示, 从左至右依次分别是迷你游戏、倩女幽魂手游和逆水寒端游这三种游戏环境的试验结果, 对于三个游戏环境下的任务组 Q_1 , 平均改进都是 0%, 因为只有基线方法在 Q_1 上实验, 而 SIAT 方法则是在 Q_2 上实验。在实际应用中, 平均改进都大于 0%, 这意味着 SIAT 方法的性能优于基线方法。

[0106] 以及, 最左边的图形中, 折线向上的趋势表明 SIAT 方法的性能在提高。特别地, Q_5 和 Q_7 在倩女幽魂手游环境上的平均提升低于之前的值, 这是因为之前的训练数据中没有出现完成任务所需的全部知识, 使得语言模型的性能下降。然而, 语言模型将在使用这些新任务生成的数据进行训练后得到改进, 相似的结果出现在逆水寒端游环境上的 Q_3 、 Q_5 和 Q_7 。

[0107] 对于基于不同语言模型的 SIAT 方法和基线方法在相同任务上的探索时间对比的实验结果, 其中, 图 8-a 为迷你游戏上的语言模型的搜索时间, 图 8-b 为倩女幽魂手游上的语言模型的搜索时间, 图 8-c 为逆水寒端游上的语言模型的搜索时间, 由此可知, SIAT 方法的每个语言模型都优于基线方法; 同时, 随着语言模型迭代, 探索时间整体呈下降趋势, 因此, 可以得出如下结论: 早期任务生成的数据在大多数情况下有利于后一种模型构建, 可以减少时间成本, 这也体现了在游戏任务回归测试过程中的自我改进行为。此外, 如前所述, 文本描述栏中噪声的存在可能会导致语言模型输出错误以及语言模型可能不适合未来的任务, 因为它用之前收集的数据训练的, 而新的任务可能完全不同。所有这些不利因素都会在今后的测试中造成更高的时间成本, 并且语言模型可能只有在训练了这些新任务之后才能给出正确的预测, 这也解释了为什么有些模型的探索时间比之前的模型要长, 如逆水寒端游环境中, 模型 4 比模型 3 的探索时间要长。

[0108] 在上述方法实施例的基础上, 本发明实施例还提供一种游戏测试装置, 如图 9 所示, 该装置包括依次连接的获取模块 91、确定模块 92 和回归测试模块 93, 其中, 各个模块的功能如下:

[0109] 获取模块 91, 用于获取待测试游戏的目标任务集;

[0110] 确定模块 92, 用于在预先根据待测试游戏构建的目标图模型中, 确定目标任务集中的各个目标任务分别对应的状态节点序列; 其中, 目标图模型包括多个状态节点和多个状态节点中至少部分状态节点关联的任务, 具有第一关联关系的第一状态节点和第二状态节点间连接有关系线; 关系线用于表征智能体从第一状态节点移动至第二状态节点执行的动作;

[0111] 回归测试模块 93, 用于对于目标任务集中的目标任务, 均根据目标任务对应的状态节点序列中的关系线, 控制智能体执行关系线对应的动作, 以对目标任务集中的各个目标任务进行回归测试。

[0112] 本发明实施例提供的游戏测试装置, 通过预先根据待测试游戏构建的目标图模

型,确定各个待测试的目标任务对应的状态节点序列,以便控制智能体执行每个目标任务对应的状态节点序列中的关系线对应的动作,从而实现对各个目标任务的回归测试,与现有的人工回归测试方法相比,减少了人力和时间消耗,提高了任务回归测试效率,进而提高了游戏的质量和用户游戏体验度。

[0113] 在其中一种可能的实施例中,上述目标图模型的训练过程包括:获取待测试游戏的任务训练集;其中,任务训练集包括多个训练任务;控制待测试游戏中的智能体完成每个训练任务,并记录完成每个训练任务对应的状态节点;在原始图模型中建立每个训练任务与记录的状态节点之间的关联关系,得到目标图模型;其中,原始图模型包括智能体在待测试游戏中的多个状态节点,具有关联关系的第一状态节点和第二状态节点间连接有关系线;关系线用于表征智能体从第一状态节点移动至第二状态节点执行的动作。

[0114] 在另一种可能的实施例中,上述状态节点配置有每种动作类型对应的动作参数,上述控制待测试游戏中智能体完成每个训练任务,包括:对于处于当前状态节点的智能体,确定智能体需要执行的当前动作对应的目标动作类型;从当前状态节点中获取目标动作类型对应的动作参数;控制智能体按照目标动作类型和目标动作类型对应的动作参数执行当前动作,以使智能体从当前状态节点移动至下一个状态节点。

[0115] 在另一种可能的实施例中,上述确定智能体需要执行的当前动作对应的目标动作类型,包括:如果智能体需要执行的当前动作对应多个动作类型,按照动作类型对应的置信系数对多个动作类型排序;根据多个动作类型的排序结果选择当前动作对应的目标动作类型。

[0116] 在另一种可能的实施例中,上述确定智能体需要执行的当前动作对应的目标动作类型之前,该装置还包括:应用预先训练好的语言模型确定训练任务中每种动作类型的置信系数。

[0117] 在另一种可能的实施例中,上述语言模型配置有衰减因子,上述应用预先训练好的语言模型确定训练任务中每种动作类型的置信系数,包括:将训练任务输入预先训练好的语言模型,得到语言模型输出的训练任务中每种动作类型的置信系数;或者,响应针对于衰减因子的调整操作,调整语言模型的衰减因子,将训练任务输入调整衰减因子后的语言模型,得到语言模型输出的训练任务中每种动作类型的置信系数。

[0118] 在另一种可能的实施例中,上述确定模块92还用于:控制智能体完成每个目标任务,并记录每个目标任务对应的状态节点;在目标图模型中查找每个目标任务对应的状态节点,得到目标任务集中的各个目标任务分别对应的状态节点序列。

[0119] 在另一种可能的实施例中,上述回归测试模块93还用于:基于目标任务对应的初始状态节点序列中的关系线,生成目标状态节点序列;其中,目标状态节点序列为初始状态节点序列的子集;控制智能体执行目标状态节点序列中的关系线对应的动作。

[0120] 在另一种可能的实施例中,该装置还包括:响应于目标任务集测试完成,生成测试报告,并将测试报告进行显示。

[0121] 本发明实施例提供的游戏测试装置,与上述实施例提供的游戏测试方法具有相同的技术特征,所以也能解决相同的技术问题,达到相同的技术效果。

[0122] 本发明实施例还提供一种电子设备,包括处理器和存储器,存储器存储有能够被处理器执行的机器可执行指令,处理器执行机器可执行指令以实现上述游戏测试方法。

[0123] 参见图10所示,该电子设备包括处理器100和存储器101,该存储器101存储有能够被处理器100执行的机器可执行指令,该处理器100执行机器可执行指令以实现上述游戏测试方法。

[0124] 进一步地,图10所示的电子设备还包括总线102和通信接口103,处理器100、通信接口103和存储器101通过总线102连接。

[0125] 其中,存储器101可能包含高速随机存取存储器(RAM,Random Access Memory),也可能还包括非不稳定的存储器(non-volatile memory),例如至少一个磁盘存储器。通过至少一个通信接口103(可以是有线或者无线)实现该系统网元与至少一个其他网元之间的通信连接,可以使用互联网,广域网,本地网,城域网等。总线102可以是ISA(Industrial Standard Architecture,工业标准结构总线)总线、PCI(Peripheral Component Interconnect,外设部件互连标准)总线或EISA(Enhanced Industry Standard Architecture,扩展工业标准结构)总线等。上述总线可以分为地址总线、数据总线、控制总线等。为便于表示,图10中仅用一个双向箭头表示,但并不表示仅有一根总线或一种类型的总线。

[0126] 处理器100可能是一种集成电路芯片,具有信号的处理能力。在实现过程中,上述方法的各步骤可以通过处理器100中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器100可以是通用处理器,包括中央处理器(Central Processing Unit,简称CPU)、网络处理器(Network Processor,简称NP)等;还可以是数字信号处理器(Digital Signal Processor,简称DSP)、专用集成电路(Application Specific Integrated Circuit,简称ASIC)、现场可编程门阵列(Field-Programmable Gate Array,简称FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本发明实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本发明实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成,或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器,闪存、只读存储器,可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器101,处理器100读取存储器101中的信息,结合其硬件完成前述实施例的方法的步骤。

[0127] 本实施例还提供一种机器可读存储介质,机器可读存储介质存储有机器可执行指令,机器可执行指令在被处理器调用和执行时,机器可执行指令促使处理器实现上述游戏测试方法。

[0128] 本发明实施例所提供的游戏测试方法、装置和电子设备的计算机程序产品,包括存储了程序代码的计算机可读存储介质,所述程序代码包括的指令可用于执行前面方法实施例中所述的方法,具体实现可参见方法实施例,在此不再赘述。

[0129] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统 and 装置的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0130] 另外,在本发明实施例的描述中,除非另有明确的规定和限定,术语“安装”、“相连”、“连接”应做广义理解,例如,可以是固定连接,也可以是可拆卸连接,或一体地连接;可以是机械连接,也可以是电连接;可以是直接相连,也可以通过中间媒介间接相连,可以是两个元件内部的连通。对于本领域的普通技术人员而言,可以根据具体情况理解上述术语在本

发明中的具体含义。

[0131] 所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用，可以存储在一个处理器可执行的非易失的计算机可读取存储介质中。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备）执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U盘、移动硬盘、只读存储器（ROM, Read-Only Memory）、随机存取存储器（RAM, Random Access Memory）、磁碟或者光盘等各种可以存储程序代码的介质。

[0132] 在本发明的描述中，需要说明的是，术语“中心”、“上”、“下”、“左”、“右”、“竖直”、“水平”、“内”、“外”等指示的方位或位置关系为基于附图所示的方位或位置关系，仅是为了便于描述本发明和简化描述，而不是指示或暗示所指的装置或元件必须具有特定的方位、以特定的方位构造和操作，因此不能理解为对本发明的限制。此外，术语“第一”、“第二”、“第三”仅用于描述目的，而不能理解为指示或暗示相对重要性。

[0133] 最后应说明的是：以上所述实施例，仅为本发明的具体实施方式，用以说明本发明的技术方案，而非对其限制，本发明的保护范围并不局限于此，尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：任何熟悉本技术领域的技术人员在本发明揭露的技术范围内，其依然可以对前述实施例所记载的技术方案进行修改或可轻易想到变化，或者对其中部分技术特征进行等同替换；而这些修改、变化或者替换，并不使相应技术方案的本质脱离本发明实施例技术方案的精神和范围，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应所述以权利要求的保护范围为准。

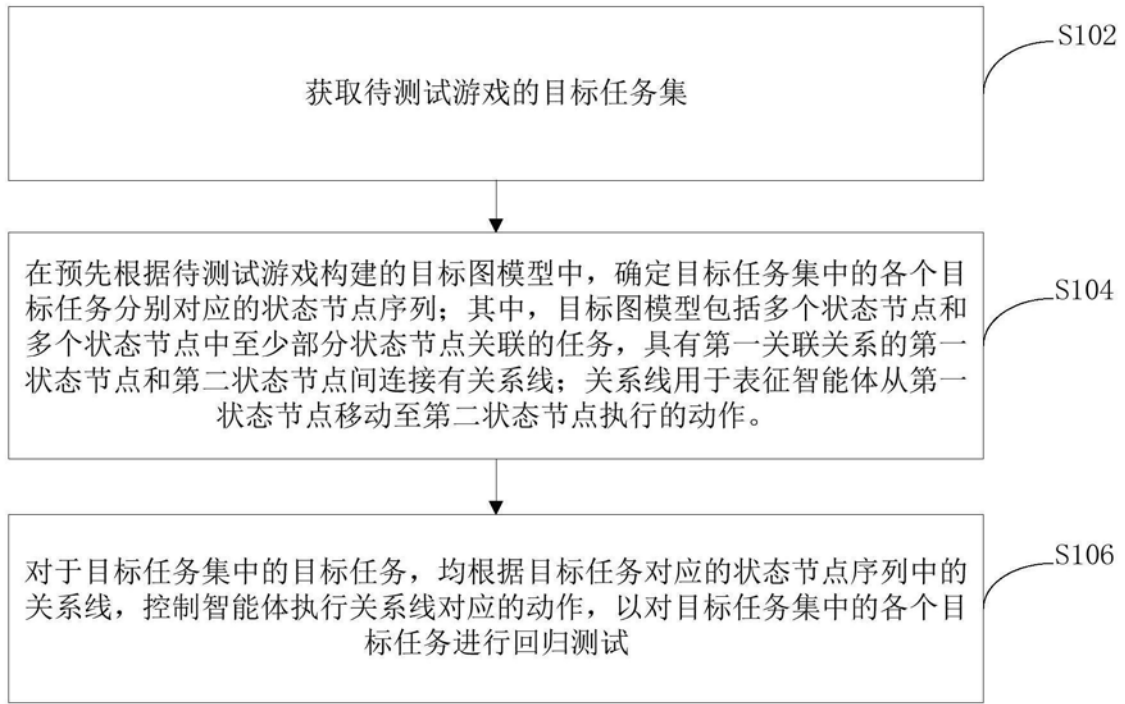


图1

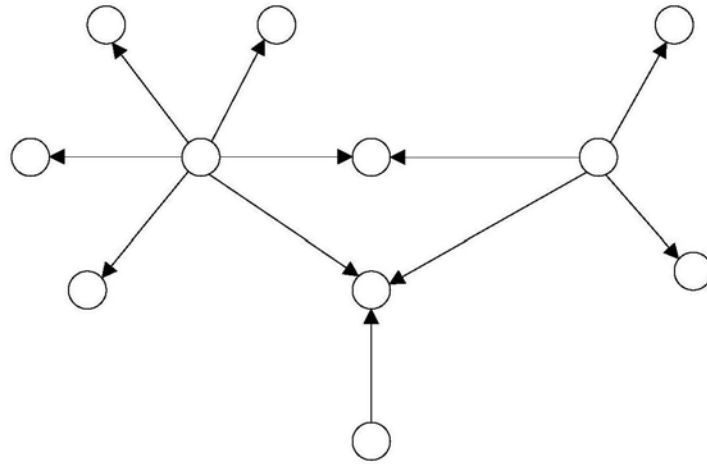


图2

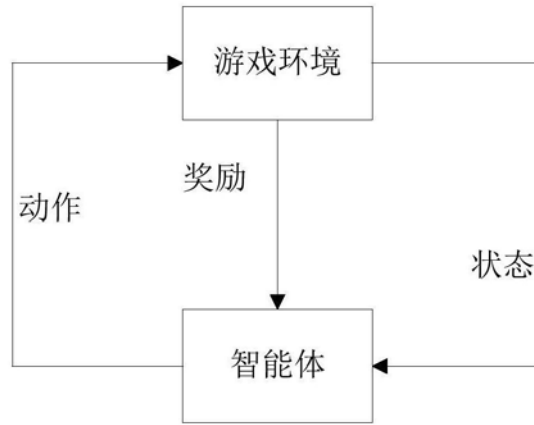


图3

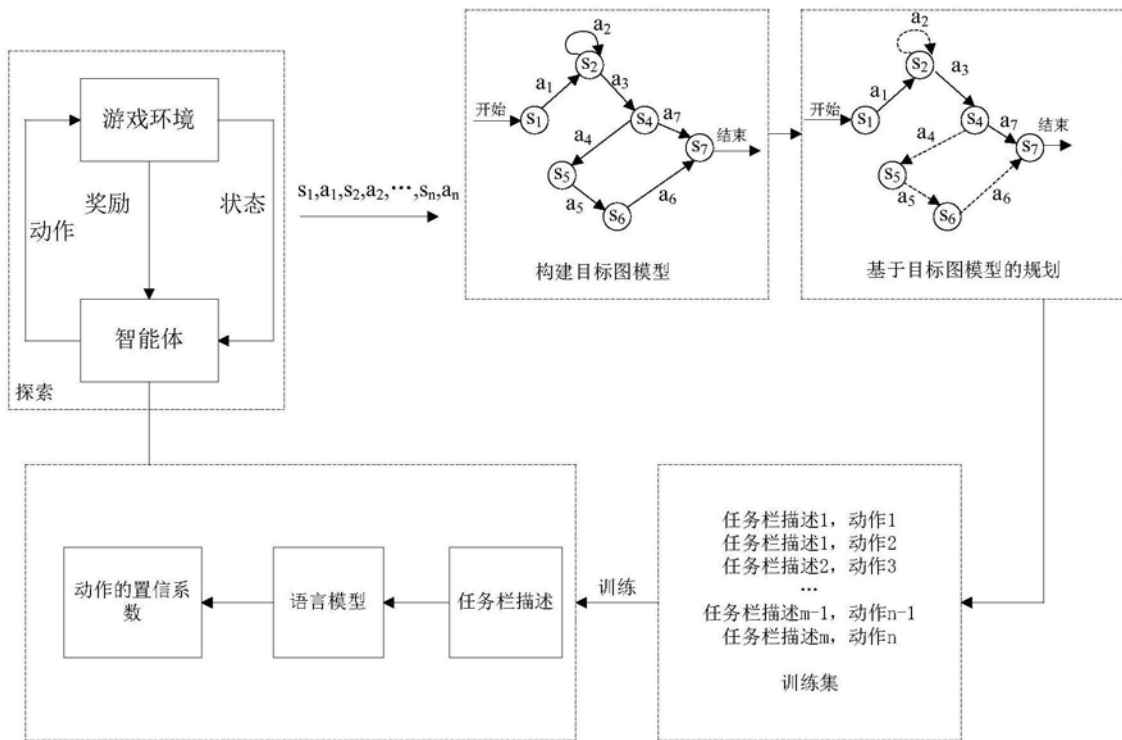


图4

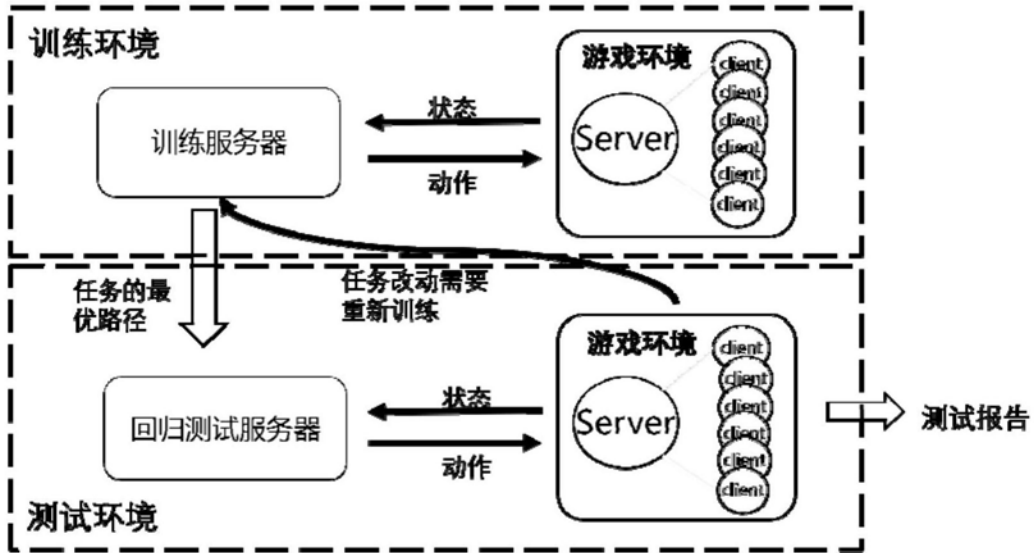


图5

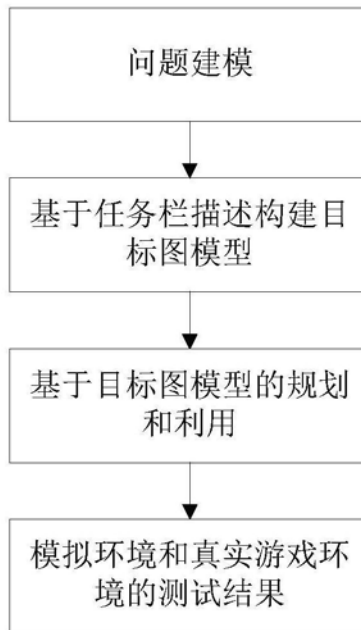


图6

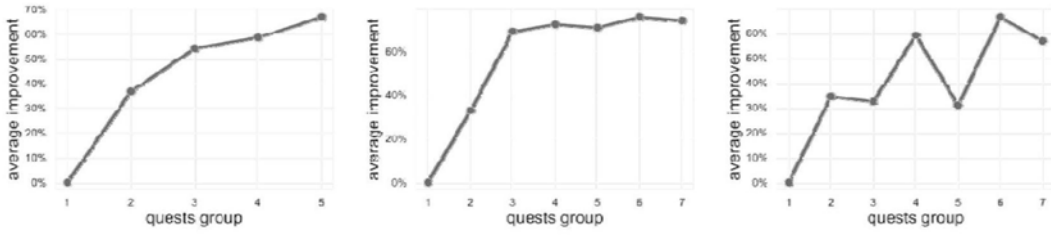


图7

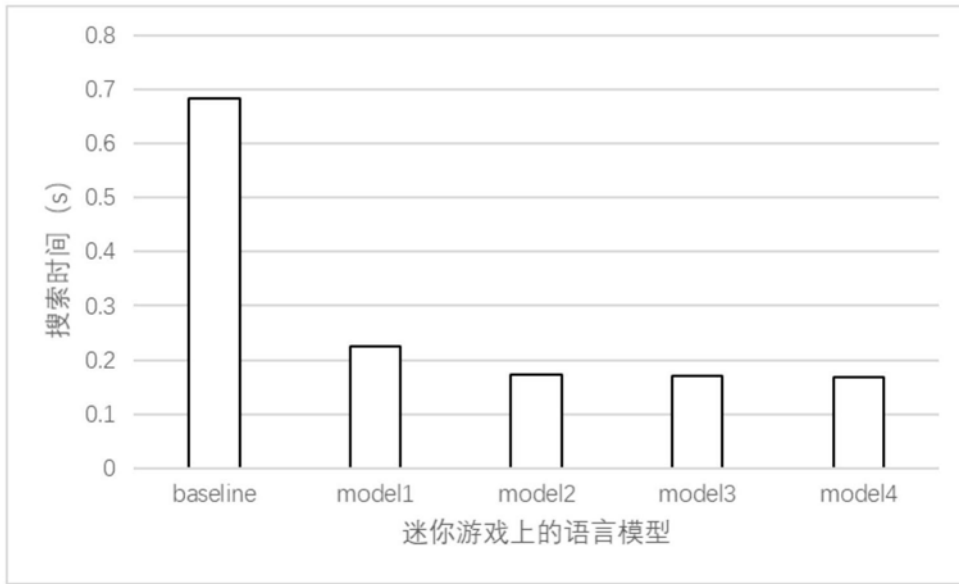


图8-a

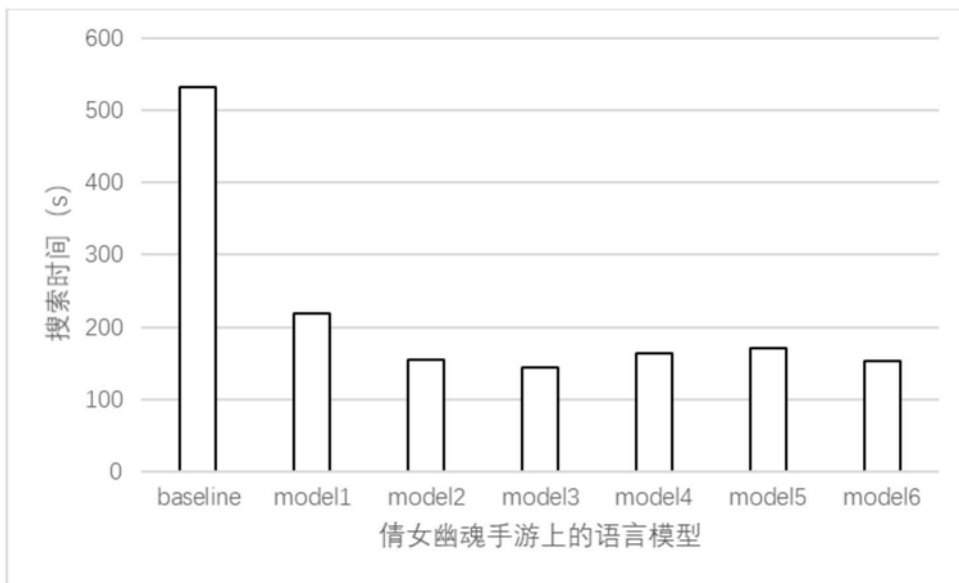


图8-b

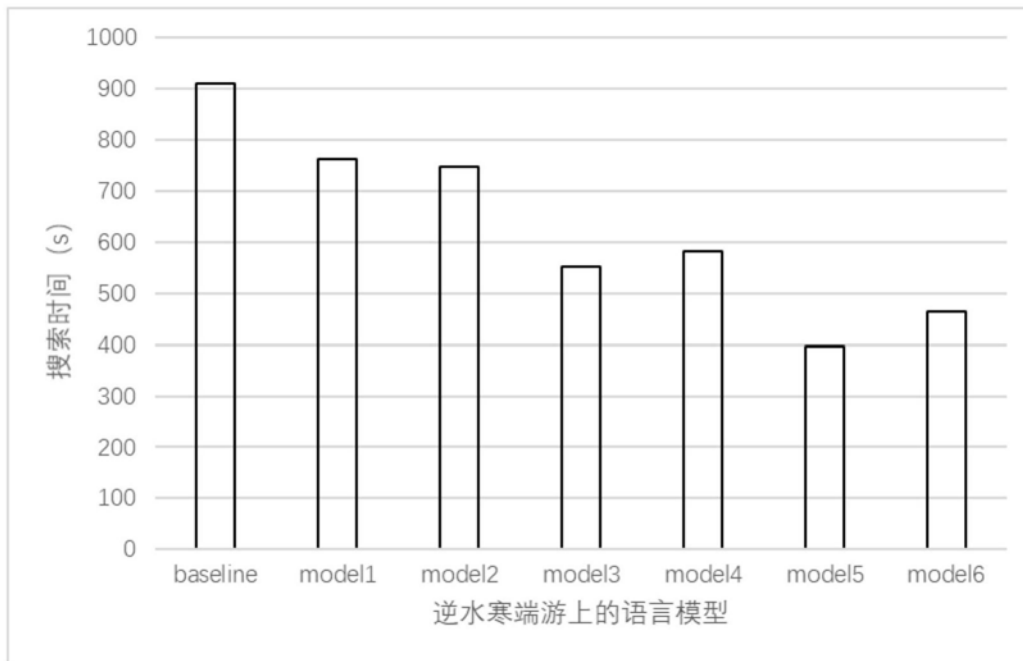


图8-c



图9

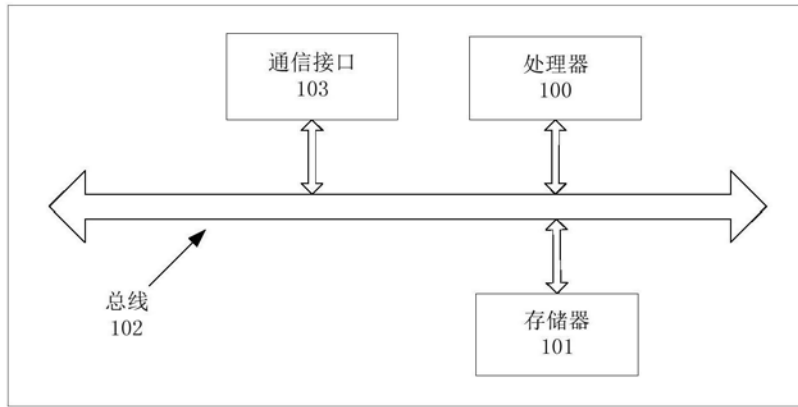


图10