arXiv:2305.02255v1 [cs.AI] 3 May 2023

场景生成的上下文推理技术报告

Loris Bozzato[1], Thomas Eiter[2], Rafael Kiesel[2], 和 Daria Stepanova[3]

[1] 布鲁诺·凯斯勒基金会，Via Sommarive 18，38123 特伦托，意大利
[2] 维也纳工业大学逻辑与计算研究所，Favoritenstraße 9-11，
A-1040 维也纳，奥地利[3] 博世人工智能中心，德国伦宁根

摘要。 本文是我们先前工作的延续。在先前工作中，我们开发了MR-CKR 框架，用于在多关系层级组织的上下文之间进行知识覆盖推理。推理通过结合代数度量的 ASP 实现，从而能够灵活地定义偏好关系。本文展示如何将我们的理论工作应用于真实的自动驾驶车辆场景数据。本文旨在将 MR-CKR 应用于生成具有挑战性的场景，以用于自动驾驶车辆学习。实际上，自动驾驶车辆学习模型所用的大多数场景数据都描述常见情形，因此可能难以捕捉特定情形发生的案例（例如，行人穿过路口时受到部分遮挡）。MR-CKR 模型允许利用此类数据的多维特性（例如时间和空间）来组织数据。通过在多个上下文中进行推理，并结合不同的场景本体，可以验证和配置场景。我们介绍了一个语义引导的数据生成框架，该框架结合了 MR-CKR 和代数度量。该框架已在一个概念验证原型中实现，并通过若干案例展示了场景生成。

1 引言与动机

测试和评估是自动驾驶车辆（AV）开发与部署中的重要环节。为了全面评估 AV 的性能，至关重要的是在安全关键场景中测试 AV 的感知系统。这类场景在自然驾驶环境中很少发生，但在实际中仍有可能出现。因此，针对性、系统地生成此类极端案例成为一个重要问题。现有研究大多关注为 AV 的感知系统生成对抗样本，这些方法通常只对输入进行非常简单的扰动（例如改变车辆的颜色或位置）；相比之下，针对基于本体且具有特定上下文的复杂场景生成所开展的工作十分有限（例如在雨天傍晚儿童遛狗）。这正是我们希望在本微型项目中研究的问题。

具体而言，我们将感兴趣的任务定义如下：给定一个来自已知数据集的现有场景（由场景图表示），我们希望生成一组作为当前场景变体的新场景，并且这些场景：

1. **具有现实感：**也就是说，与描述场景中对象的本体保持一致（例如，交通标志通常不会移动）；2. **有趣：**也就是说，它们满足某种语义限制，该限制表明场景对我们的预测模型而言，例如是“危险的”或具有挑战性的（例如，在街道中央看到一只猫就需要采取特殊行动）；3. **相似：**也就是说，将原始场景转换为生成场景只需要进行小幅变化。

我们提出使用符号方法，基于现有场景图和场景的语义定义生成有效且具有挑战性的场景。具体而言，MR-CKR [3,2] 是一种对此很有用的形式化方法。在这里，本体知识被置于上下文中，因此在不同上下文中可能具有不同的解释（可能产生非单调效应）。这意味着 MR-CKR 能够帮助我们生成<style id='2'>具有现实感的场景，因为它能够处理描述 AC 领域的背景本体。此外，我们还可以拥有关于<style id='4'>有趣性的不同上下文相关概念。MR-CKR 还允许我们通过在不同上下文中对场景关联不同的独立语义限制来表达这一点。

MR-CKR 的另一项优势是，它可以转换为答案集编程（ASP）。ASP 是一种声明式编程语言，可用于轻松表达并高效求解复杂的逻辑问题。

除了关注真实感和趣味性，我们还关心相似度。因此，我们需要一种方法来衡量生成场景与最初作为起点的原始场景有多相似。所谓的代数度量 [9] 在这里非常有用。它是 ASP 领域中的一个通用框架，可用于衡量与解相关的量。因此，它也能够表达我们所需的场景相似度度量。

我们的主要贡献如下：

– 我们提出了一种用于语义引导数据生成的新型框架，可用于生成对抗数据或训练数据。– 通过将 MR-CKR 与代数度量相结合来构建框架，我们获得了一种高度灵活的方法，并可通过转换为 ASP 高效求解。– MR-CKR 使我们能够：(i) 纳入本体背景知识，确保生成数据的真实性；(ii) 将何种生成输入才算有趣这一概念置于上下文中。– 代数度量能够最大化原始数据与生成数据之间的相似度。– 我们面向自动驾驶领域场景生成的原型有意保持精简，但展现出良好的前景。

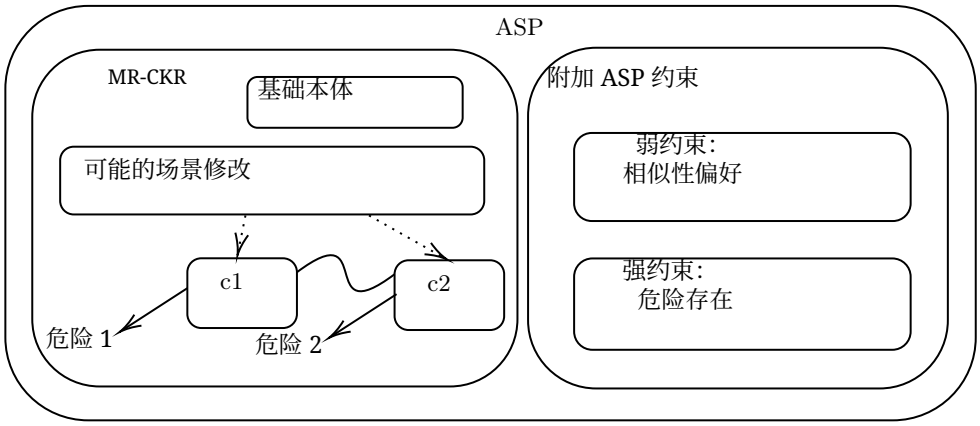

图 1. 该通用框架使用 ASP，根据 MR-CKR 所定义的（可能相关的）危险类型，生成（相似的）危险场景。

相关工作。 生成神经模型的对抗性或具有挑战性的示例是一项重要问题，近年来在工业界[4] 和研究领域 [13,16,6,10]都受到关注。这些研究同样重视生成与原始输入相似且真实的输入。然而，这里采用的是数值方法，而非符号方法来生成新输入并验证生成输入的真实性。例如， [13] 使用图像的微小数值扰动； [16] 则通过优化来最小化输入数据的数值变化，从而使网络给出不同的预测。我们发现，与我们的工作最接近的是 [10]，该方法通过尽可能少地替换字符，为自然语言处理生成对抗文本。不过，尽管这种最小替换优化可视为一种符号方法，研究并未验证新生成文本的真实性。

2 框架概述

在介绍我们如何生成具有挑战性的新训练场景描述的技术细节之前，我们先对框架的整体结构进行概述。

为此，我们考虑图 1 中描述的模式。在这里，我们看到我们使用一个 MR-CKR，一方面定义可能的场景修改，另一方面定义不同的上下文（此处为 $c1$ 和 $c2$），用于指定场景可能具有危险性或趣味性的情形。理想情况下，这些不同类型的危险对应于神经网络工程师针对当前神经网络性能不佳所做的诊断。例如，在自动驾驶场景中，我们可能会观察到，当不仅有一个停车标志时，汽车未能停在正确的位置。

[4] https://www.efemarai.com/

标志，同时还有指定汽车应停位置的停车线标记。因此，我们希望以这样的方式修改场景，使其同时包含停车标志和停车线标记。

总体而言，目标是生成更多我们认为该网络也会表现不佳的场景，以便获得可用于训练神经网络的对抗样本，并期望借此提升它在这些因训练数据不足而难以应对的情境中的表现。给定不同上下文中的危险定义（即基于不同诊断的定义），这些定义可能通过特化或其他方式关联起来；随后，我们便可在 ASP 中获得等效编码，进而得到模型，也就是符合 MR-CKR 中所含基础本体的逼真生成场景。

此外，我们还添加了更多 ASP约束，以确保对场景的修改能够使最终场景（a）具有危险性（使用强约束），并且（b）尽可能与给定的初始场景相似（使用表达代数度量的弱约束）。

将这两点结合起来，我们便可获得逼真且危险的场景，同时尽可能贴近初始场景。此外，不同的上下文使我们能够指定不同类型的目标危险，从而使每个上下文中都包含一个涵盖相应目标危险的生成场景。

下文中，我们将形式化地阐述如何使用 MR-CKR 生成场景，并通过代数度量测量其相似度，以此具体说明我们的抽象构想。

3 MR-CKR 中场景生成问题的形式化

我们首先正式介绍 MR-CKR 框架，并提供一种利用 MR-CKR 进行场景生成的解决方案。

3.1 MR-CKR 定义

我们采用描述逻辑的常规定义（例如，参见 [1] 以了解简介）。下面总结 MR-CKR的主要定义（如 [2]中所介绍）。

我们考虑一种基于 DL 签名 Σ 的通用描述语言 $\mathcal{L}_\Sigma$，该签名由一组概念名称 NC、角色名称 NR 和个体名称 NI 组成。

考虑一个非空集合 $\mathbf{N} \subseteq$ NI，其中包含上下文名称。上下文关系是上下文上的任意严格序 $\prec_i \subseteq \mathbf{N} \times \mathbf{N}$。定义上下文关系的一种方法是使用上下文维度 [4,14], ，即与每个上下文相关联的一组上下文“坐标”：例如，在场景描述中，这些坐标可以表示一天中的时间、位置类型或场景中发生的情境。于是，上下文结构由与上下文相关联的特征（维度）的序积定义，对应于一个上下文关系。

在 MR-CKR 中，上下文内的公理可以指定为可废止的（即在出现例外时可以被推翻），其依据是构成上下文结构的某个上下文关系。

定义 1（r-可废止公理）。给定一个定义在$\mathbf{N}$ 上的上下文关系集合 $\mathcal{R}$ 以及描述语言 $\mathcal{L}_\Sigma$，r-可废止公理是形如$\mathrm{D}_r(\alpha)$的任意表达式，其中 α 是 $\mathcal{L}_\Sigma$ 的公理，且 $\prec_r \in \mathcal{R}$。

我们允许在上下文的局部语言中使用 r-可废止公理：

定义 2（上下文语言）。给定上下文名称集合 $\mathbf{N}$，对于每种描述语言 $\mathcal{L}_\Sigma$，我们定义 $\mathcal{L}_{\Sigma,\mathbf{N}}$ 为 $\mathcal{L}_\Sigma$ 的扩展，其中：*(i)* $\mathcal{L}_{\Sigma,\mathbf{N}}$ 包含 $\mathcal{L}_\Sigma$ 中的 *r*- 可废止公理集合；*(ii)* 如果 X 是 $\mathcal{L}_\Sigma$ 中的概念（分别为角色），且$\mathsf{c} \in \mathbf{N}$，那么 $\mathit{eval}(X,\mathsf{c})$ 是 $\mathcal{L}_{\Sigma,\mathbf{N}}$ 中的概念（分别为角色）。

多关系上下文知识表示由基于 $\mathcal{R}$ 中的上下文关系的全局上下文结构，以及与每个局部上下文相关联的一组描述逻辑知识库组成。

定义3（多关系简单上下文知识表示）。一个 多关系简单上下文知识表示（sCKR）基于 Σ 和 $\mathbf{N}$ 是一个结构 $\mathfrak{K} = \langle \mathfrak{C}, \mathrm{K}_\mathbf{N} \rangle$ 其中：

– $\mathfrak{C}$ 是一个结构 （$\mathbf{N}$ $\prec_1$、...、 $\prec_m$），其中每个 $\prec_i$ 都是 N 上的上下文关系，并且 – 对于每个上下文名称 c $\in\mathbf{N}$，K_c 是$\mathcal{L}_\Sigma$上的一个 *DL* 知识库，$\mathbf{N}$。

示例*1*。 我们提供一个 MR-CKR 的简单示例，以更好地说明可废止公理的预期用法。考虑由以下元素组成的 sCKR $\mathfrak{K} = \langle \mathfrak{C}, \{\mathrm{K}_1, \mathrm{K}_2\} \rangle$：

$$\begin{aligned} \mathfrak{C} &= \{\mathsf{c}_2 \prec_c \mathsf{c}_1\} \\ \mathrm{K}_1 &= \{\mathrm{D}_c(\mathit{Dog} \sqsubseteq \neg \mathit{DangerousAnimal})\} \\ \mathrm{K}_2 &= \{\mathit{Dog} \sqsubseteq \mathit{DangerousAnimal}, \mathit{Dog}(d)\} \end{aligned}$$

直观地说，我们希望认识到，在更特定的上下文 c_2, 中，狗被视为危险动物，因此 c_1中的更一般可废止公理不适用于实例 d Dog。

MR-CKR 的解释是与每个上下文相关联的一族描述逻辑解释。

定义 4（简单上下文知识表示解释）。对$\mathcal{L}_{\Sigma,\mathbf{N}}$ 的解释是一族$\mathfrak{I} = \{\mathcal{I}(\mathsf{c})\}_{\mathsf{c}\in\mathbf{N}}$ 解释，其中 $\Delta^{\mathcal{I}(\mathsf{c})} = \Delta^{\mathcal{I}(\mathsf{c}')}$ 且 $a^{\mathcal{I}(\mathsf{c})} = a^{\mathcal{I}(\mathsf{c}')}$，对每个 $a \in \mathrm{NI}$ 和 c，$\mathsf{c}' \in \mathbf{N}$。

$\mathcal{L}_\Sigma$ 中概念和角色表达式的解释，$\mathbf{N}$ 通过将标准解释扩展到 eval 表达式而获得：对于每个 c $\in$ $\mathbf{N}$,

$eval(X,\mathsf{c}')^{\mathcal{I}(\mathsf{c})} = X^{\mathcal{I}(\mathsf{c}')}$。我们采用 [3] 提供的公理实例化定义：给定一个公理 $\alpha \in \mathcal{L}_\Sigma$，其一阶逻辑翻译为 $\forall \mathbf{x}.\phi_\alpha(\mathbf{x})$，用个体元组 $\mathbf{e}$ 对 NI 中的个体进行实例化，记作 $\alpha(\mathbf{e})$，即将 α 特化为 $\mathbf{e}$，也就是 $\phi_\alpha(\mathbf{e})$，具体取决于 α 的类型。

对于上下文 c 和上下文关系 r，冲突假设 是一个二元组 $\langle \alpha,\mathbf{e}\rangle$，满足 $\alpha(\mathbf{e})$ 是 α 的公理实例化，且 $\mathsf{c}' \succeq_{-r} \mathsf{c}'' \succ_r \mathsf{c}$。对于 $\langle \alpha, \mathbf{e}\rangle$，冲突集 是一个可满足的 ABox 断言集合 S，使得 $S \cup \{\alpha(\mathbf{e})\}$ 不可满足。

定义 5（CAS 解释）。*CAS* 解释 是一个结构$\mathfrak{I}_{CAS} = \langle \mathfrak{I}, \chi\rangle$，其中 $\mathfrak{I}$ 是一个解释，并且满足：对于每个 χ_i，对于 $i \in \{1,\ldots,\overline{m}\}$，将每个 $\mathsf{c} \in \mathbf{N}$ 映射到一个集合 $\chi_i(\mathsf{c})$，该集合包含上下文 c 和上下文关系 $\prec_i$的冲突假设。

定义 6（CAS 模型）。给定一个多关系 *sCKR* $\mathfrak{K}$，若 *CAS* 解释$\mathfrak{I}_{CAS} = \langle \mathfrak{I}, \chi\rangle$ 满足以下条件，则称其为 $\mathfrak{K}$ 的CAS 模型 （记为 $\mathfrak{I}_{CAS} \models \mathfrak{K}$）：

(i) 对于每个 $\alpha \in K_{\mathsf{c}}$ （严格公理）以及 $\mathsf{c}' \preceq_* \mathsf{c}$， $\mathcal{I}(\mathsf{c}') \models \alpha$；*(ii)* 对于每个 $D_i(\alpha) \in K_{\mathsf{c}}$ 以及 $\mathsf{c}' \preceq_{-i} \mathsf{c}$， $\mathcal{I}(\mathsf{c}') \models \alpha$；*(iii)* 对于每个 $D_i(\alpha) \in K_{\mathsf{c}}$ 以及 $\mathsf{c}'' \prec_i \mathsf{c}' \preceq_{-i} \mathsf{c}$，若 $\langle \alpha,\mathbf{d}\rangle \notin \chi_i(\mathsf{c}'')$，则 $\mathcal{I}(\mathsf{c}'') \models \phi_\alpha(\mathbf{d})$。

我们为每种关系在冲突假设集上定义了一个局部偏好：

（*LP*）。 $\chi_i^1(\mathsf{c}) > \chi_i^2(\mathsf{c})$，如果对于每个在上下文 $\mathsf{c}_1 \succeq_{-i} \mathsf{c}_{1b} \succ_i \mathsf{c}$ 中具有 $\mathrm{D}_i(\alpha_1)$ 的 $\langle \alpha_1, \mathbf{e}\rangle \in \chi_i^1(\mathsf{c}) \setminus \chi_i^2(\mathsf{c})$，都存在一个在上下文 $\mathsf{c}_2 \succeq_{-i} \mathsf{c}_{2b} \succ_i \mathsf{c}$ 中具有 $\mathrm{D}_i(\alpha_2)$ 的 $\langle \alpha_2, \mathbf{f}\rangle \in \chi_i^2(\mathsf{c}) \setminus \chi_i^1(\mathsf{c})$，使得 $\mathsf{c}_{1b} \succ_i \mathsf{c}_{2b}$。

直观地说，如果 $\chi_i^1(\mathsf{c})$ 在更具体的上下文中用更一般上下文中的“成本更低”的例外，替换了 $\chi_i^2(\mathsf{c})$ 的“成本更高”的例外，则 $\chi_i^1(\mathsf{c})$ 优选于 $\chi_i^2(\mathsf{c})$。

如果对于每个 $c \in \mathrm{NI}$，$c^{\mathcal{I}_1} = c^{\mathcal{I}_2}$ 都成立，则两个描述逻辑解释 $\mathcal{I}_1$ 和 $\mathcal{I}_2$ 是 NI-同余的。通过考虑所有上下文解释 $\mathcal{I}(\mathsf{c}) \in \mathfrak{I}$，这一概念可扩展到 CAS 解释 $\mathfrak{I}_{CAS} = \langle \mathfrak{I}, \chi\rangle$。

定义 7（论证）。如果存在某个冲突集 $S_{\langle \alpha,\mathbf{e}\rangle,\ \mathsf{c}}$，使得对于所有与 $\mathfrak{I}_{CAS}$*NI* 同余的 $\mathfrak{I}'_{CAS} = \langle \mathfrak{I}', \chi\rangle$（属于 $\mathfrak{K}$），都有 $\mathcal{I}'(\mathsf{c}) \models S_{\langle \alpha,\mathbf{e}\rangle,\ \mathsf{c}}$成立，则称 *CAS* 模型 $\mathfrak{I}_{CAS}$中的 $\langle \alpha,\mathbf{e}\rangle \in \chi_i(\mathsf{c})$ 是有论证支持的。如果 *sCKR* $\mathfrak{K}$ 的 *CAS* 模型 $\mathfrak{I}_{CAS}$ 中每个 $\langle \alpha,\mathbf{e}\rangle \in \chi$ 都在 $\mathfrak{K}$中有论证支持，则称该模型有论证支持。

我们通过组合各关系的偏好关系来定义模型偏好关系：它是模型上的全局字典序，其中每个 $\prec_i$ 定义第 i 个位置上的排序。

（*MP*）。 $\mathfrak{I}^1_{CAS} = \langle \mathfrak{I}^1, \chi_1^1, \ldots, \chi_m^1\rangle$ 优选于 $\mathfrak{I}^2_{CAS} = \langle \mathfrak{I}^2, \chi_1^2, \ldots, \chi_m^2\rangle$，当且仅当

(i) 存在 $i \in \{1,\ldots,m\}$ 以及某个 $\mathsf{c} \in \mathbf{N}$ 使得 $\chi_i^1(\mathsf{c}) > \chi_i^2(\mathsf{c})$ 且 not$\chi_i^2(\mathsf{c}) > \chi_i^1(\mathsf{c})$，并且不存在任何上下文 $\mathsf{c}' \neq \mathsf{c} \in \mathbf{N}$ 使其满足 $\chi_i^1(\mathsf{c}') < \chi_i^2(\mathsf{c}')$ 且 not $\chi_i^2(\mathsf{c}') < \chi_i^1(\mathsf{c}')$。

(ii) 对于每个 $j < i \in \{1, \ldots, m\}$， $\chi_j^1 \approx \chi_j^2$ 都成立（即 (i) 或其逆命题对 $\prec_j$ 不成立）。

定义 8（CKR 模型）。解释 $\mathfrak{I}$ 是 *sCKR*$\mathfrak{K}$ 的CKR 模型（符号表示为 $\mathfrak{I} \models \mathfrak{K}$），如果满足：*(i)* $\mathfrak{K}$ 存在某个有依据的 *CAS* 模型 $\mathfrak{I}_{CAS} = \langle \mathfrak{I}, \chi \rangle$；*(ii)* 不存在优选于 $\mathfrak{I}_{CAS}$的有依据的 $\mathfrak{I}'_{CAS} = \langle \mathfrak{I}', \chi' \rangle$。

示例 *2*。使用上述语义机制，我们可以展示如何解释前一个示例中的 sCKR。具体而言，我们可以考虑 CAS-解释 $\mathfrak{I}_{CAS} = \langle \mathfrak{I}, \chi \rangle$，其中 $\chi(\mathsf{c}_2) = \{\langle$狗 $\sqsubseteq \neg$危险动物 $d\rangle\}$。这意味着，如果 c_1 的 $\mathfrak{I}_{CAS}$ 可废止公理不应用于 c_2，中唯一的狗，则它是一个 CAS 模型，正如预期的那样。请注意，这样的 CAS 模型也是有依据的，因为冲突假设允许冲突集 {狗(d)、危险动物(d)}：因此，鉴于不存在关于偏好关系而言最小的其他可选 CAS 模型，所考虑的解释也属于 CKR 模型。上述定义的偏好关系有助于优先选择最特定上下文中的可废止公理：例如，如果 c_2 中的狗 $\sqsubseteq$危险动物 被定义为可废止的（相对于上述可废止公理的相同上下文关系），则下方的上下文 c_2 将优先选择更特定的公理，从而选择在上层上下文 c_1 的更一般公理上作出例外的解释。

作为在 MR-CKR 上实现推理的方法，在 [2] 中我们提供了将 MR-CKR 翻译为 ASP 逻辑程序的方法：具体而言，我们展示了这种翻译可用于在给定上下文中进行实例检查和查询回答。

3.2 MR-CKR 中的场景生成

遵循图 1 所体现的直观结构，在我们的架构中，MR-CKR 的作用是基于一个通用的场景本体，定义我们希望生成的场景的逻辑约束。

鉴于其多上下文结构，MR-CKR 有助于为基础场景的内容及其针对不同目标诊断所做的修改提供复杂表示（上下文化）。

关于第二个方面，上下文的基本组织可以定义如图 2 所示。

该结构的上下文通过上下文关系 $\succ_{sim}$ 联系起来，该关系表示相似度关系：上层上下文*Exchange* 以可废止公理的形式包含可在诊断场景中修改的公理。在*Base*上下文中，我们假设其中包含基础场景的描述以及场景描述本体的基础公理。随后，上下文诊断-*1*、.....、诊断-N提供了我们希望建模的基础场景的不同修改。对这些不同修改进行建模所需的公理类型，取决于我们希望允许在场景修改中进行的更改类型（添加、删除等）：后续章节将通过讨论特定修改，进一步介绍这类公理。

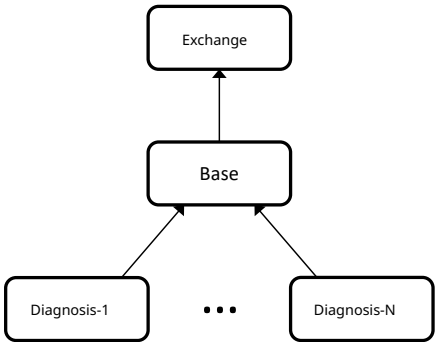

图 2. 用于场景修改的上下文通用结构

关于场景上下文化，我们可以利用 MR-CKR 的多关系特性，进一步定义场景发生的上下文。

示例 *3.* 图 3 展示了这种上下文化的一个示例。

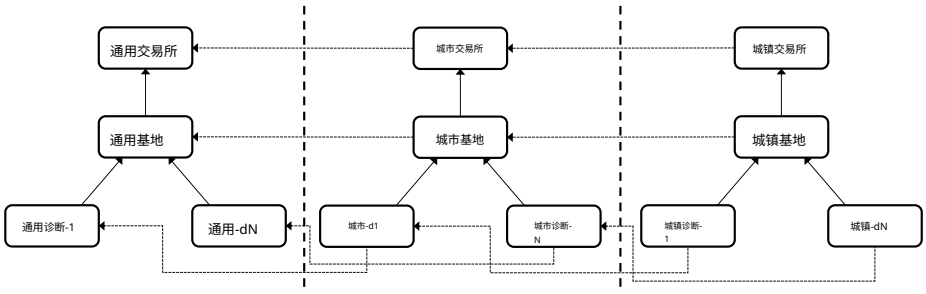

图 3. 用于场景表示的 MR-CKR 通用结构示例

在这个上下文结构中，水平箭头所表示的关系体现了场景根据位置的具体程度进行的特化：从适用于一般场景的公理出发，我们可以添加适用于城市场景的更多逻辑约束，然后再添加适用于城镇场景的约束。请注意，这一方向与上文所述的 Base 上下文结构正交。

在通过此类框架对场景进行建模后，我们希望使用 MR-CKR 到 ASP 的转换来生成诊断上下文的可能模型：这些模型随后对应于备选的生成场景。不过，我们现在需要一种方法来衡量生成场景与感兴趣场景之间的相似度：正如我们将在后续章节中详述的，这可以通过代数度量轻松实现。

4 基于代数度量的相似度形式化表示

如果我们想基于一个起始场景生成新场景，就需要优化相似度度量。为了衡量相似度，我们可以使用代数度量。代数度量背后的直观思想是，它们能够衡量与某种解释或模型相关的量。为了在统一的框架下衡量不同的量，我们采用半环这一代数结构，它支持多种不同形式的计算。

4.1 预备知识

我们介绍代数度量及其必要的预备知识。

定义 9（幺半群）。一个幺半群 $\mathcal{M} = (M, \otimes, e_\otimes)$ 由集合 M 上的一个结合二元运算 $\otimes$ 以及一个幺元 $e_\otimes$构成，该幺元也称为单位元。这里，集合 M 上的二元运算是一个函数 $\otimes : M \times M \to M$，它将来自 M 的数值对映射到 M中的一个值。我们使用中缀记法$m_1 \otimes m_2$，将二元运算 $\otimes$ 应用于数值对(m_1, m_2) $m_1, m_2 \in M$。

如果对于所有值 $m \in M$，均满足以下条件，则值 $e_\otimes \in M$ 是集合 M 上二元运算 $\otimes$ 的幺元：

$$e_\otimes \otimes m = m = m \otimes e_\otimes.$$

此外，二元运算 $\otimes$ *on* M 是 结合的，如果对于所有 m、m'、$m'' \in M$，都有

$$m \otimes (m' \otimes m'') = (m \otimes m') \otimes m''.$$

幺半群的一些例子包括

– 字符串 $= (\{0,1\}^*, \odot, \varepsilon)$，由二进制字符串在连接 $\odot$和空字符串 ε 下构成的集合是一个非交换、非幂等且不可逆的幺半群，– $\mathcal{P}(A) = (2^A, \cup, \emptyset)$，集合 A 的子集在并集 $\cup$ 下构成的集合是一个交换、幂等且不可逆的幺半群，– $\mathcal{P}(A) = (2^A, \cap, A)$，集合 A 的子集在交集 $\cap$ 下构成的集合是一个交换、幂等且不可逆的幺半群，– $\mathbb{Z} = (\mathbb{Z}, +, 0)$，整数在加法 $+$ 下构成的集合是一个交换、非幂等且可逆的幺半群。

基于幺半群，我们引入半环。

定义 10（半环）。半环 $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$ 是一个非空集合 R，配备了两个二元运算 $\oplus$ 和 $\otimes$，分别称为加法和乘法，并满足

– $(R, \oplus)$ 是以 $e_\oplus$ 为单位元的交换幺半群，– $(R, \otimes)$ 是以 $e_\otimes$ 为单位元的幺半群，

– 乘法对加法满足左右分配律，即对于所有 $r, r', r'' \in R$，都有

$$r \otimes (r' \oplus r'') = r \otimes r' \oplus r \otimes r''$$
$$(r' \oplus r'') \otimes r = r' \otimes r \oplus r'' \otimes r$$

– 乘以 $e_\oplus$ 会将 R消去，即对于所有 $r \in R$，都有

$$r \otimes e_\oplus = e_\oplus = e_\oplus \otimes r.$$

半环的一些例子如下

– $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$，布尔半环，其中析取和合取分别作为加法和乘法，– $\mathbb{F} = (\mathbb{F}, +, \cdot, 0, 1)$，对于 $\mathbb{F} \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$，这是 $\mathbb{F}$中数构成的半环，其运算为加法和乘法，– $\mathcal{P}(A) = (2^A, \cup, \cap, \emptyset, A)$，这是 A的幂集上的半环，其运算为并集和交集，以及 – $\mathcal{R}_{\min, +} = (\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$，最小加半环。

另一份半环列表（附有应用说明）见 [12]。

为了将半环的定量方面与逻辑的定性方面联系起来，我们使用加权逻辑。加权逻辑最初由 [8]在二阶设定下引入。这里，我们只介绍命题逻辑的受限版本。

定义 11（语法）。设 $\mathcal{V}$ 为一个命题变量集合，$\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$ 为一个半环。关于 $\mathcal{R}$ 的加权（命题）公式具有由文法给出的形式 α：

$$\alpha ::= k \mid v \mid \neg v \mid \alpha + \alpha \mid \alpha * \alpha$$

其中 $k \in R$ 且 $v \in \mathcal{V}$。
我们可以根据一种解释对加权公式求值，从而得到一个半环中的值。

定义 12（语义）。给定一个加权命题公式 α，其取值于半环 $\mathcal{R} = (R, \oplus, \otimes, e_\oplus, e_\otimes)$，命题变量取自 $\mathcal{V}$，以及一个解释 $\mathcal{I}$（即 $\mathcal{V}$ 的一个子集），则 α 关于 $\mathcal{I}$ 在 $\mathcal{R}$ 上的语义 $[\![\alpha]\!]_{\mathcal{R}}(\mathcal{I})$ 定义如下：

$$[\![k]\!]_{\mathcal{R}}(\mathcal{I}) = k$$
$$[\![v]\!]_{\mathcal{R}}(\mathcal{I}) = \begin{cases} e_\otimes & v \in \mathcal{I} \\ e_\oplus & otherwise. \end{cases} \quad (v \in \mathcal{V})$$
$$[\![\neg v]\!]_{\mathcal{R}}(\mathcal{I}) = \begin{cases} e_\oplus & v \in \mathcal{I} \\ e_\otimes & otherwise. \end{cases} \quad (v \in \mathcal{V})$$
$$[\![\alpha_1 + \alpha_2]\!]_{\mathcal{R}}(\mathcal{I}) = [\![\alpha_1]\!]_{\mathcal{R}}(\mathcal{I}) \oplus [\![\alpha_2]\!]_{\mathcal{R}}(\mathcal{I})$$
$$[\![\alpha_1 * \alpha_2]\!]_{\mathcal{R}}(\mathcal{I}) = [\![\alpha_1]\!]_{\mathcal{R}}(\mathcal{I}) \otimes [\![\alpha_2]\!]_{\mathcal{R}}(\mathcal{I}).$$

我们定义了代数度量，以结合答案集程序的定性语言与加权逻辑的定量语言。

定义 13（代数度量）。一个代数度量 $\mu = \langle \Pi, \alpha, \mathcal{R} \rangle$由一个答案集程序 Π、一个加权公式 α和一个半环 $\mathcal{R}$组成。于是，在 μ 下，答案集 $\mathcal{I} \in \mathcal{AS}(\Pi)$ 的权重定义为

$$\mu(\mathcal{I}) = [\![\alpha]\!]_{\mathcal{R}}(\mathcal{I}).$$

此外，对于来自 Π 的原子 a，其（原子）查询结果为

$$\mu(a) = \bigoplus_{\mathcal{I} \in \mathcal{AS}(\Pi), a \in \mathcal{I}} \mu(\mathcal{I}),$$

Π 的总体权重查询结果为

$$\mu(\Pi) = \bigoplus_{\mathcal{I} \in \mathcal{AS}(\Pi)} \mu(\mathcal{I}).$$

直观而言，这里的想法是：对于代数度量 $\mu = \langle \Pi, \alpha, \mathcal{R} \rangle$，半环 $\mathcal{R}$决定定量计算的模式，Π 给出逻辑背景理论，用以确定哪些解释是解，而α 则为每个答案集 $\mathcal{I}$ 赋予一个半环上的权重；具体做法是根据解释 $\mathcal{I}$中原子公式是否成立，在半环上进行计算。

4.2 场景的相似度

广义而言，我们可以通过三种方式修改场景：

（i）对象添加、（ii）对象删除，或（iii）对象修改。

我们的目标是为这些操作指定成本。然后，我们就可以通过执行一系列操作并将其成本相加，计算从一个场景得到另一个场景需要付出多大成本。成本越高，两个场景的相似度就越低。

（i）和（ii）的效果很明确，且可选方案不多。唯一的可能性是区分不同复杂程度对象的添加/删除，并为更复杂的对象指定更高的成本。必要时，之后可以进一步探讨这种方案。目前，我们假设添加和删除的固定成本分别为 cost(Add) 和 cost(Del)。

然而，对于对象修改，我们有不同的选择：

– 位移（例如，强制两个对象发生重叠），– 类别变化，以及 – 属性变化（即移除某个属性、添加某个属性，或修改某个属性的值）。

对于类别变化，添加一些限制是合理的，否则，我们可以通过对象修改，在同一步骤中实现删除和加法。为此，我们断言，只有当 class C 和 class C' 具有合理的共同超类 C_s 时，才允许将前者交换为后者，即，对于一个不属于 owl: $Thing$ 或类似类别的超类 C_s，必须满足 $C \sqsubseteq C_s$ 和 $C' \sqsubseteq C_s$。在这里，我们选择了一系列合理的超类，包括 车辆、动物和街道标志。

我们为每项修改指定的成本如下：

– **位移：** 要么取原位置与新位置之间的距离乘以一个常数因子，要么取固定成本 cost($Disp$)。– **类别变化：** 将类别 C 替换为 C' 时，若其最低共同超类为 C_s，成本就是推导出 $C \sqsubseteq C_s$ 和 $C' \sqsubseteq C_s$所需使用的最少描述逻辑公理数。例如，考虑以下公理集：

$$\text{刺猬} \sqsubseteq \text{危险动物}, \text{老虎} \sqsubseteq \text{危险动物}$$

这里，对于 $C = \text{刺猬}$ $C' = \text{老虎}$，以及 $C_s = \text{危险动物}$，我们需要一条描述逻辑公理来推导出 $\text{刺猬} \sqsubseteq \text{危险动物}$，还需要一条描述逻辑公理来推导出 $\text{老虎} \sqsubseteq \text{危险动物}$。因此，将老虎替换为刺猬的成本为 2。

– **属性变化：** 我们为删除、加法和修改分别指定一个常数成本 cost($PDel$)，cost($PAdd$)，以及 cost($PMod$)。令 cost($PDel$) = cost($PAdd$) > cost($PMod$) 是合理的。

基于这些关于如何度量相似度的假设，我们可以继续进行其度量建模。

4.3 使用代数度量度量相似度

代数度量由三部分组成：指定逻辑约束的程序、指定如何度量权重的加权公式，以及指定所度量权重类型的半环。对于半环，使用 $\mathcal{R}_{\min,\ +}$ 是合理的。这样，我们就可以对解释所产生的不同成本求和；如果存在不同选项，则选择可能的最低成本。

逻辑约束主要由 MR-CKR 给出。然而，为了指定加权公式，我们需要能够使用一些原子公式，以指明执行了哪些修改才得到模型中的场景。因此，我们确保程序的签名包含以下谓词：

– 加法(C, I)，表示将个体 I 添加到类别 C；– 删除(C, I)，表示从类别 C中删除个体 I；– 位移(I)（分别表示位移(I, D)），表示个体 I 的位移（分别表示位移距离为 D）；

– classVar(I, C, C')，表示个体 I 在原始场景中属于类别 C，但现在属于类别 C'；– propertyVar(I, T)，表示个体 I的某项属性发生了类型为 T的修改。

使用上述谓词，我们可以轻松地指定加权公式 α_{cost}，用于衡量将原始场景转换为修改后场景的成本，如下所示：

$$
\begin{aligned}
&\Pi_{\text{class } c,\ \text{个体 } i}(\text{加法}(c,\ i) * \text{成本}(Add) + \neg\text{加法}(c,\ i)) \\
&* \Pi_{\text{类别 } c,\ \text{个体 } i}(\text{删除}(c,\ i) * \text{成本}(Del) + \neg\text{删除}(c,\ i)) \\
&* \Pi_{\text{个体 } i}(\text{位移}(i) * \text{成本}(Disp) + \neg\text{位移}(i)) \\
&* \Pi_{\text{类别 } c,\ c',\ \text{个体 } i}(\text{classVar}(i,\ c,\ c') * \text{dist}(c,\ c') + \neg\text{classVar}(i,\ c,\ c')) \\
&* \Pi_{\text{个体 } i,\ t\in\{PDel,PAdd,PMod\}}(\text{propertyVar}(i,\ t) * \text{成本}(t) + \neg\text{propertyVar}(i,\ t))
\end{aligned}
$$

每一行分别处理一种修改类型的成本。这里，dist(c, c')表示两个类别 c 和 c'之间的距离，如上文所述。我们可以针对每个上下文静态地计算这些距离。

示例 *4*。例如，考虑以下解释

$$\mathcal{I} = \{\text{addition}(RollingContainer, i_1), \text{deletion}(Child, i_2), \text{deletion}(Child, i_3)\}.$$

直观地说，这意味着我们将对象 i_1 添加到概念 $RollingContainer$中，并从概念 儿童中移除对象 i_2 和 i_3。

我们预计，这种解释的成本为 cost(Add) + 2 ·cost(Del)，因为我们向一个概念添加了一个对象，并移除了两个对象。

据此，我们得到$[\![\alpha_{cost}]\!]_{\mathcal{R}_{\min,+}}(\mathcal{I} = \text{cost}(Add) + 2 \cdot \text{cost}(Del)$。理由如下。首先，注意到 α_{cost}定义的最后三行等于 $e_\otimes$，因为对于所有 i, c, c', t，¬ displacement(i) ¬classVar(i, c, c') ¬propertyVar(i, t)均成立，因此

$$
\begin{aligned}
&[\![\text{位移}(i) * \text{成本}(Disp) + \neg\text{位移}(i)]\!]_{\mathcal{R}} \\
&= [\![\text{位移}(i) * \text{成本}(Disp)]\!]_{\mathcal{R}} \\
&= [\![\text{位移}(i)]\!]_{\mathcal{R}} \\
&= e_\oplus \otimes [\![\text{成本}(Disp)]\!]_{\mathcal{R}} \\
&= e_\oplus \oplus e_\otimes = e_\otimes
\end{aligned}
\qquad
\begin{aligned}
&{}_{\text{最小值 }+}(\mathcal{I}) \\
&{}_{\text{最小值 }+}(\mathcal{I}) \oplus [\![\neg\text{位移}(i)]\!]_{\mathcal{R}} \quad {}_{\text{最小值 }+}(\mathcal{I}) \\
&{}_{\min\ +}\ (\mathcal{I}) \otimes [\![\text{成本}(Disp)]\!]_{\mathcal{R}\ \min\ +}\ (\mathcal{I}) \oplus e_\otimes \\
&{}_{\text{最小值 }+}\ {}^{(\mathcal{I})\ \oplus e_\otimes}
\end{aligned}
$$

最后两行也可以观察到同样的情况。

另一方面，基于相同的推理，由于解释 $\mathcal{I}$ con- 包含加法($RollingContainer\ i_1$)，第一行的计算结果为成本(Add)，第二行的计算结果为成本(Del)⊗成本(Del)，因此得到

$$\text{cost}(Add) \otimes \text{cost}(Del) \otimes \text{cost}(Del).$$

由于我们使用了$\mathcal{R}_{\min,\ +}$ 该操作 $\otimes$ 是 $+$，因此我们获得了cost(Add)+2·cost(Del)作为最终成本，正如预期。

4.4 向弱约束的转换

虽然代数度量是为程序的答案集指定定量度量的有用工具，但目前大多数 ASP 求解器不支持优化代数度量的权重。不过，我们使用的代数度量可以转换为弱约束 [5]。

回想一下，直观地说，加权公式 α_{cost} 对应于对场景所执行修改的成本求和。例如，对于

$$(\text{addition}(c,i) * \text{cost}(Add) + \neg\text{addition}(c,i))$$

如果 $\neg$addition(c,i)成立，则成本为 0；如果 addition(c,i)成立，则成本为 cost(Add)。

这正是可以使用如下形式的弱约束进行建模和优化的成本类型

$$:\sim a_1,\ldots,a_n,\text{not } b_1,\ldots,\text{not } b_m.[C,t_1,\ldots,t_k],$$

其中， a_i 和 b_j 是原子公式， C 是成本， t_l 是项。该弱约束表示，满足 $a_1,\ldots,a_n$ 但不满足 $b_1,\ldots,b_m$ 会产生 C 的成本。项 $t_1,\ldots,t_k$ 直观地将不同的弱约束分组。也就是说，如果存在多个具有相同项的弱约束，则只有成本最高的那个会被触发。

这意味着，我们可以使用弱约束

$$:\sim\text{addition}(C,I).[\text{cost}(Add),add,C,I]$$

为每个个体 i 和概念 c 添加成本 cost(Add)，使得加法(c,i) 成立，即向概念 c 添加 i 。

我们可以对 α_{cost} 的其他因素进行相同的处理，将其转换为带有弱约束的 ASP。

5 自动驾驶场景生成实现原型

我们使用自动驾驶中的一个示例实现并测试了我们的方法。在此，我们重构并略微扩展了基础本体，来自[7,15]。[5] 它包含不同的场景，每个场景都标注了其中包含的对象。所包含的对象都标注了相关信息，例如其类型。此外，本体还包含公理，用于添加关于本体中不同概念之间关系的额外知识。总体而言，基础本体包含超过300万条公理，涉及41个概念、100多个场景以及5万多个对象的知识。

在我们的原型中，我们将自身限制于有限的场景变化，即从概念中加法和删除对象，并为每项分配成本 1。不过，也可以毫无问题地加入上文详述的其他修改。

[5] https://github.com/boschresearch/ad_cskg

5.1 概述

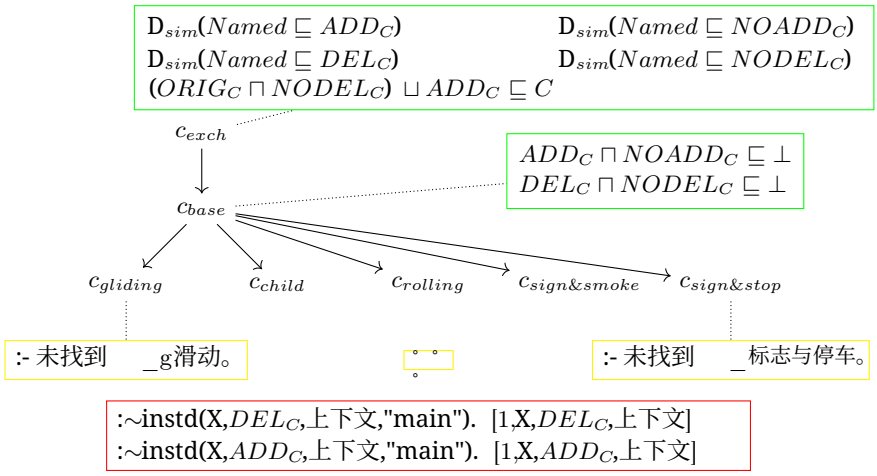

图 4. 我们用于自动驾驶原型的 MR-CKR 和附加约束。 c_i 表示不同的上下文，上下文之间的箭头表示更高的特异性。属于某个上下文的本体公理和附加 ASP 约束分别显示在绿色框和黄色框中，并以虚线连接到相应的上下文。红色框中包含用于优化相似度的附加弱约束。

我们在图4中概述了MR-CKR及其与ASP约束的相互作用。我们将逐步介绍不同部分。

对于网络故障的诊断，我们使用以下内容（采用本体中已有的类别）：

1. 场景中包含一个属于类别 $GlidingOnWheels$ 的物体。由于缺少否定示例而具有危险性。2. 场景中包含一个属于类别 $Child$ 的物体。由于相较于其他人类，其行为难以预测而具有危险性。3. 场景中包含一个属于类别 $RollingContainer$ 的物体，但不包含属于类别 $Human$ 的物体。由于滚动容器的行为难以预测而具有危险性。4. 场景中包含一个属于类别 $Sign$ 的物体和一个属于类别 $Smoke$ 的物体。由于烟雾导致标志更难识别而具有危险性。5. 场景中包含一个属于类别 $Sign$ 的物体和一个属于类别 $StopLineMarking$ 的物体。由于网络无法正确预测车辆会在停止线处停车而具有危险性。

这意味着每个诊断 i 都有一个上下文 c_i 。

回顾图 1 中的整体框架。为了对可能的场景修改进行建模，我们有一个交换上下文 c_{exch}，其中为本体中的每个原始（可修改）概念 C 都包含以下默认公理：

$$\mathsf{D}_{sim}(Named \sqsubseteq ADD_C)\ \mathsf{D}_{sim}(Named \sqsubseteq NOADD_C)\ \mathsf{D}_{sim}(Named \sqsubseteq DEL_C)\ \mathsf{D}_{sim}(Named \sqsubseteq NODEL_C)\ (ORIG_C \sqcap NODEL_C) \sqcup ADD_C \sqsubseteq C$$

这里，

– $Named$ 是一个概念，我们根据原始本体定义该概念，其中包含所有可修改的对象；– ADD_C 是一个概念，表示应添加到概念 C 中的个体；– $NOADD_C$ 是一个概念，表示不应添加到概念 C 中的个体；– DEL_C 是一个概念，表示应从概念 C中移除的个体；– $NODEL_C$ 是一个概念，表示不应从概念 C 中移除的个体；– $ORIG_C$ 是一个概念，表示在原始本体中属于概念 C 的个体。

因此，对于每个可修改个体，我们断言应将其添加到概念 C中（或不添加），或从中移除（或不移除）。如果它们原本属于 C，且未被移除，或者被添加到 C，那么根据最后一条公理，它们就应属于 C。显然，单独来看，这并不合理，因为每个可修改个体都会属于每个概念。因此，我们添加一个基础上下文 c_{base}，其中包含本体中每个（可修改）概念 C 的以下公理：

$$ADD_C \sqcap NOADD_C \sqsubseteq \bot$$
$$DEL_C \sqcap NODEL_C \sqsubseteq \bot$$

这些公理确保我们要么添加（或移除）某个个体，要么不添加（或不移除）该个体，但不能两者兼有。然后，如果就$\succ_{sim}$而言，c_{exch} 的具体程度低于 c_{base}，我们就可以在 c_{base} 中覆盖 c_{exch} 的默认规则，从而满足 c_{base} 中的不相交要求。

为了让诊断的上下文 c_i 能够访问修改的可能性，我们随后声明，每个上下文 c_i 就 $\succ_{sim}$而言都比 c_{base} 更具体。

剩下的就是附加 ASP 约束，它们（i）确保修改幅度最小；（ii）确保在给定上下文中存在该诊断。

对于 (i)，我们对每个可修改的概念 C 使用以下规则：

: ~ instd(X, DEL_C, 上下文, "main"). [1,X, DEL_C, 上下文]:
~ instd(X, ADD_C, 上下文, "main"). [1,X, ADD_C, 上下文]

这确保每次添加或删除一个 X 到 C时，都会增加 1 的惩罚。请注意，该惩罚适用于每个上下文。

对于（ii），我们只需添加确保能够推导出该诊断的约束。例如，对于上下文 C_3 中的第三个诊断，我们需要推导出概念 $RollingContainer$ 中存在一个对象，而概念 $Human$ 中不存在对象，因此添加以下规则

found_rolling_no_人类_1 :- instd(X, $RollingContainer$, C_3, "main"). :- not found_rolling_no_人类_1. found_rolling_no_人类_2 :- instd(X, $Human$, C_3, "main"). :- found_rolling_no_ 人类 _2.

示例*5*。 假设我们的输入场景包含四个对象 i_1，...， i_4 以及

$$Child(i_2), Child(i_3), Car(i_4)$$

保持。

由于本体公理 $Child \sqsubseteq Human$，我们可以推导出 $Human(i_2)$ 和 $Human(i_3)$。

因此，在上下文 C_3中，我们需要一个滚动容器但不需要人类，所以需要从 $Child$ 概念中移除 i_2 和 i_3 ，并向$RollingContainer$ 概念中添加一个对象。因此，一个潜在的修改（仅限于上下文C_3）可通过以下解释表示

$$\begin{aligned}\mathcal{I} = \{&\text{instd}(i_1, ADD_{RollingContainer}, C_3, \text{“}main\text{”}),\\ &\text{instd}(i_2, DEL_{Child}, C_3\,\text{“}main\text{”}),\\ &\text{instd}(i_3, DEL_{Child}, C_3, \text{“}main\text{”})\}.\end{aligned}$$

如前一个示例所述，其成本为 cost(Add)+2·cost(Del)。由于我们将加法和删除的成本设为 1，因此该成本为 3。

由于不存在成本更低的修改， C_3 的一个可能生成场景由以下内容组成

$$RollingContainer(i_1), Car(i_4),$$

也就是说，其中包含一个滚动容器 i_1，没有人类，但有一辆汽车 i_4。

5.2 实际实现

为了将上文概述的思路转化为一个可通过 ASP 求解器（例如 Clingo [11], ）求解问题的形式化编码，我们按以下步骤进行:

1. 构建 MR-CKR 编码 2. 将 MR-CKR 编码转换为 ASP 3. 添加强约束以确保危险性 4. 添加弱约束以确保相似度

我们实施了所有这些步骤，并将其在线提供。[6] 更具体地说，我们按如下方式处理它们。

[6] https://github.com/raki123/MR-CKR

构建 *MR-CKR* 编码。 在这里，我们读取基础本体，并收集 (i) 通常成立的公理，以及 (ii) 有关某些特定场景的知识。然后，我们按照图4所示构建 MR-CKR。这里有两点需要注意。首先，我们还将先前从基础本体中收集的一般公理添加到 c_{base}中。这确保了生成场景的真实性。其次，我们并不会为每个概念都添加用于向概念中添加/删除个体的默认规则，而只针对其中一部分相关概念进行添加。这有助于缩小问题编码的规模，并提高推理性能。

将 *MR-CKR* 编码转换为 *ASP*。TheCKRew 软件[7] 是先前工作 [2] 中已有的工具，可将 MR-CKR 转换为 ASP。原始转换能够处理上下文之间高度复杂的关系，并支持任意默认规则和灵活的本体背景知识。但代价是，生成的 ASP 编码不适用于我们的用途，因为我们需要处理包含许多上下文和概念的大型场景图。

为了解决这个问题，我们针对自己的场景对编码进行了专门化。具体而言，对于给定的概念 C，这些（可撤销）公理

$$\mathrm{D}_{sim}(Named \sqsubseteq ADD_C) \qquad \mathrm{D}_{sim}(Named \sqsubseteq NOADD_C)$$
$$ADD_C \sqcap NOADD_C \sqsubseteq \bot$$

告诉我们，可以猜测任意命名个体 X 是否添加到 C中，要么选择加法，要么选择不加法，前提是本体中没有其他理由阻止这两种选择。由于我们的基础本体是一致的，本体中不可能存在阻止不加法的理由。因此，在我们的场景中，这种二选一确实成立。

这使我们能够使用以下规则对上述（可撤销）公理进行编码：

instd(X,ADD_C,Con,"main") :- instd(X,"Named",Con,"main"), not instd(X, $NOADD_C$,Con,"main"). instd(X,$NOADD_C$,Con,"main") :- instd(X," Named",Con,"main"), not instd(X,ADD_C,Con,"main").

我们对删除和非删除也采用相同的处理方式。

这种针对我们场景的专门翻译策略显著提升了性能。原始编码只能让我们从极小的起始场景生成新场景，而改进后的策略能够在几秒内根据本体推理出现实世界场景。

添加 *ASP*约束 添加强约束和弱约束出人意料地简单。我们可以利用 CKRew 用于翻译的词汇来引用派生知识。因此，我们可以在单独的程序文件中提供所有附加约束，并求解 MR-CKR 的 ASP 编码与附加约束的组合。

[7] https://github.com/dkmfbk/ckrew

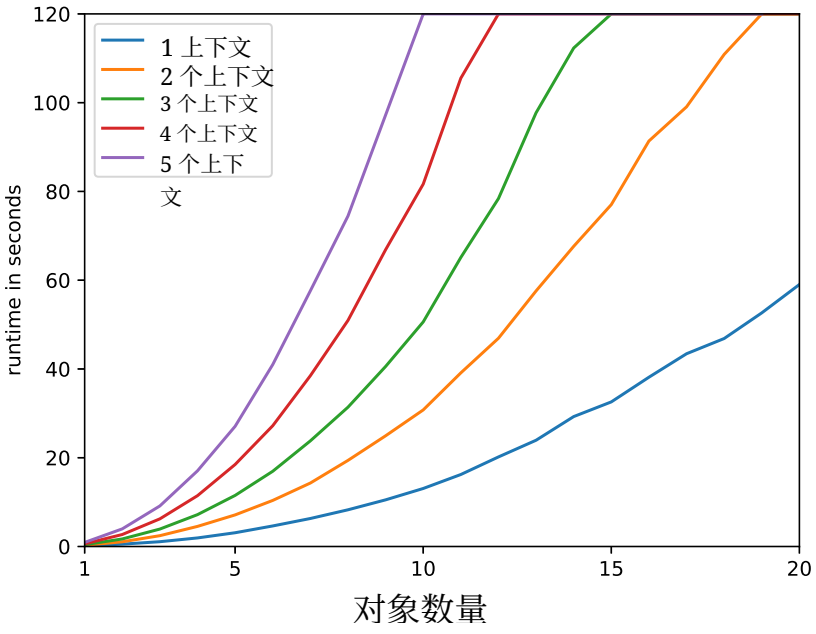

图 5. 使用 MR-CKR 到 ASP 的通用转换后的求解时间。

5.3 可扩展性

我们简要考察在保持较低运行时间的同时，我们能够求解的实例规模可以达到多大。这里，我们一方面考虑将 MR-CKR 转换为 ASP 的原始方案，记为 General；另一方面考虑利用默认规则受限使用这一特点的专门化转换方案，记为 Specialized。我们的目的是表明，在求解现实世界的问题时，使用 Specialized 取代 General 不仅有帮助，甚至是必要的。

其次，我们研究运行时间对（i）场景中对象数量和（ii）上下文数量的依赖关系。我们将（i）设置为 1 至 20，将（ii）设置为 1 至 5，并使用从本体中随机选取的一个示例场景进行测试。

我们只测量求解时间，因为构建 MR-CKR 并将 MR-CKR 转换为 ASP 编码所耗费的时间微不足道（不到 1 秒），无论转换方式以及对象/上下文的数量如何。

在求解阶段，我们使用带有输入选项“-t 3”的非基程序“Clingo” [11]，以指定使用三个线程并行求解该问题。我们设置 120 秒的时间限制，并将未能在此时间内完成的运行的运行时间记为 120 秒。

我们的调查结果见图 5 和图 6。我们发现，即使只使用一个上下文，General 之后的运行时间也会迅速增长。尽管它

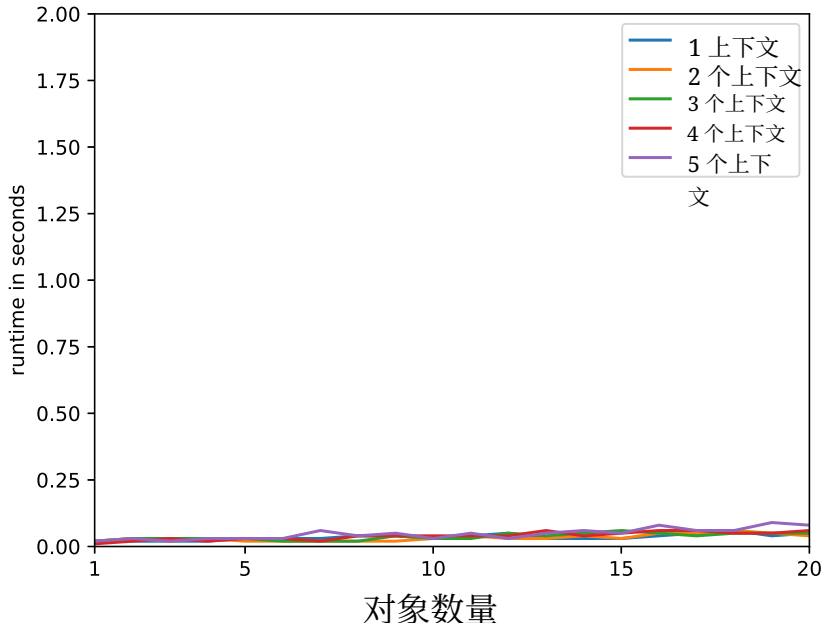

图 6。使用 MR-CKR 到 ASP 的专门化转换后的求解时间。

仍保持在可行范围内：当在场景中使用一个上下文且最多包含 20 个对象时，情况依然如此；但当使用更多上下文时，就不能这样说了。对于五个上下文，当场景中包含十个或更多对象时，求解已经变得缓慢。此外，原始场景包含更多对象（超过 300 个），因此，即使只有一个上下文，这种转换也只能应用于受限的示例。

另一方面，对于 Specialized，我们发现求解时间始终远低于一秒，即使使用全部五个上下文和场景中的 20 个对象也是如此。请注意这里 Y 轴范围的不同，我们调整了范围以便显示运行时间。即使使用场景中最初包含的所有对象（超过 300 个），求解时间仍保持在约 0.67 秒。

我们发现，尽管原始翻译 General 能够处理范围更广的 MR-CKR，但在我们的设置中采用专门翻译 Specialized 更为有效。借助 Specialized，即使使用完整场景（即其中的所有对象）和所有上下文，我们也能在不到一秒的时间内生成新场景。这表明，借助 Specialized，我们还可以针对比原型中所提供的更复杂的语义条件和基础本体生成新输入，从而为未来拓展我们的工作带来有趣的机会。

6 结论

我们提出了一种新框架，可基于现有输入为神经模型生成有趣的新输入，尤其适用于自动驾驶场景数据的场景生成。值得注意的是，我们的框架基于符号推理方法：一方面，这使我们能够融入现实世界知识（以上下文知识的形式），确保生成的输入真实可信；另一方面，还能构建新输入必须满足的语义标准。

我们看到，纳入框架的所有组件都带来了各自的优势：

– **MR-CKR**使我们能够：(i) 轻松纳入本体知识；(ii) 在不同上下文中执行不同的修改。– **代数度量** 使我们能够轻松指定要优化的成本值。– **ASP**作为一种用于转换的声明式编程语言，使我们能够借助标准求解器高效地执行推理/场景生成。

虽然我们在原型中只考虑了一个小型示例，但它成功生成了新的场景描述。此外，由于该方法可以轻松推广到不同类型场景的生成，因此为我们的方法提供了概念验证。

在未来的工作中，一个有趣的方向是扩展这一示例：通过检查神经模型在预测任务中表现不佳的输入，收集相关场景更复杂的语义描述。具体而言，可以使用更复杂的上下文结构来表示场景的不同变体，同时利用输入的性能表现，对更值得生成的案例进行量化。另一个尚待解决的挑战是，利用新场景的符号描述生成图像，将其输入神经模型，并评估使用这些新示例进行训练后，网络性能能够提升多少。

致谢

本研究部分受欧盟委员会资助的项目“Humane AI：通过理解人类、我们的社会和周围世界，打造增强并赋能人类的 AI 系统”（资助编号 #820437）和“AI4EU：欧洲人工智能按需平台与生态系统”（资助编号 #825619），以及奥地利科学基金（FWF）项目 W1255-N23 的支持。谨此致谢。

参考文献

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.（编）：《描述逻辑手册》。剑桥大学出版社（2003） 2. Bozzato, L., Eiter, T., Kiesel, R.：通过基于代数度量的答案集编程，对多关系上下文层级进行推理。《理论与实践逻辑编程》**21**(5), 593–609（2021）。https://doi.org/10.1017/S1471068421000284, https://doi.org/10 1471068421000284 3 1017. /S 。Bozzato, L., Eiter, T., Serafini, L.：通过可辩护例外增强上下文知识库。《人工智能》**257**, 72–126（2018） 4. Bozzato, L., Serafini, L., Eiter, T.：在上下文层级中运用可辩护例外进行推理。载于：KR 2018。第329–338页。AAAI出版社（2018） 5. Buccafurri, F., Leone, N., Rullo, P.：析取数据日志中的强约束和弱约束。载于：Dix, J., Furbach, U., Nerode, A.（编）《逻辑编程与非单调推理：第4届国际会议 LPNMR'97，达格施图尔城堡，德国，1997年7月28日至31日，会议论文集》。《计算机科学讲义》，第1265卷，第2–17页。施普林格（1997）。https://doi.org/10.1007/3-540-63255-7_2, https: //doi.org/
10 1007 3 540 63255 7_2 . / - - - 。Chen, B., Feng, Y., Dai, T., Bai, J., 6
Jiang, Y., Xia, S., Wang, X.：针对图像检索中的深度乘积量化网络生成对抗样本。《IEEE模式分析与机器智能汇刊》**45**(2), 1388–1404（2023）。https://doi.org/10. 1109 3165024 7 2022 3165024, 10 1109 2022 /TPAMI. . https://doi.org/ . /TPAMI. . . Chowdhury, S.N., Wickramarachchi, R., Gad-Elrab, M.H., Stepanova, D., Henson, C.A.：探索利用常识知识实现自动驾驶。载于：ISWC（海报/演示/工业界）（2021） 8. Droste, M., Gastin, P.：加权自动机与加权逻辑。《理论计算机科学》**380**(1-2), 69–86（2007）。https://doi.org/10.1016/j.tcs.2007.02.055, https://doi.org/10 055 9 1016 2007 02 . /j.tcs. . . 。Eiter, T., Kiesel, R.：用于定量流推理的加权 LARS。载于：Giacomo, G.D., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J.（编）《ECAI 2020——第24届欧洲人工智能会议，2020年8月29日至9月8日，西班牙圣地亚哥-德孔波斯特拉，2020年8月29日至9月8日——含第10届人工智能杰出应用会议（PAIS 2020）》。《人工智能与应用前沿》，第325卷，第729–736页。IOS出版社（2020）。https://doi.org/10.3233/FAIA200160, https://doi.org/10 200160 10 3233. / FAIA 。Gao, J., Lanchantin, J., Soffa, M.L., Qi, Y.：黑盒生成对抗文本序列以规避深度学习分类器。载于：2018 IEEE安全与隐私研讨会，SP Workshops 2018，美国加利福尼亚州旧金山，2018年5月24日。第50–56页。IEEE计算机学会（2018）。https://doi.org/10.1109/SPW.2018.00016, https://doi.org/10 00016 11 1109 2018 . / SPW. . . Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.：Clingo $=$ ASP $+$ 控制：初步报告。CoRR $\mathbf{abs/1405.3694}$（2014），http://arxiv.org/abs/1405. 3694 12. Kimmig, A., den Broeck, G.V., Raedt, L.D.：代数模型计数。《应用逻辑杂志》**22**, 46–62（2017）。https://doi.org/10.1016/j.jal.2016.11.031, https:// doi.org/10 031 13 1016 2016 11 . /j.jal. . . 。Rozsa, A., Rudd, E.M., Boult, T.E.：对抗多样性与困难正样本生成。载于：2016 IEEE计算机视觉与模式识别会议研讨会，CVPR Workshops 2016，美国内华达州拉斯维加斯，2016年6月26日至7月1日。

第410–417页。IEEE计算机学会（2016）。https://doi.org/10.1109/CVPRW.2016 58 14 58, 10 1109 2016 . https://doi.org/ . /CVPRW. . . Serafini, L., Homola, M.：面向语义网的情境化知识库。《Web语义》期刊**12**，64–87（2012）15. Wickramarachchi, R., Henson, C., Sheth, A.：用于驾驶场景实体预测的知识融入学习。《大数据前沿》p. 98（2021）16. Yang, B., Zhang, H., Li, Z., Zhang, Y., Xu, K., Wang, J.：使用AdaBelief优化器和裁剪不变性生成对抗样本。《应用智能》**53**(2), 2332–2347（2023）。https://doi.org/10.1007/s10489-022-03469-5, https://doi.org/10. 1007/s10489-022-03469-5