arXiv:2303.02927v3 [cs.AI] 6 Jun 2023

LIDA：一种利用大语言模型自动生成语法无关可视化和信息图的工具

Victor Dibia微软研究院
victordibia@microsoft.com

摘要

支持用户自动创建可视化的系统，必须处理若干子任务——理解数据的语义、列举相关的可视化目标，以及生成可视化规范。在这项工作中，我们将可视化生成视为一个多阶段生成问题，并指出，基于大语言模型（LLM）和图像生成模型（IGM）构建的精心编排的流水线适用于解决这些任务。我们介绍了LIDA，这是一种用于生成语法无关可视**化**和信**息**图**的**新工具。LIDA 由 4 个模块组成：摘要器将数据转换为内容丰富而精简的自然语言摘要；目标探索器 根据数据列举可视化目标；VISGENERATOR生成、优化、执行并筛选可视化代码；以及信息图生成器 利用IGM生成忠实于数据的风格化图形。 LIDA 提供 Python API，以及一套混合 用户 界面 （支持直接操作和多语言自然语言），用于交互式图表、信息图和数据故事生成。代码和演示可通过以下网址获取——
https://microsoft.github.io/lida/

1 引言

可视化通过降低从大型表格数据集中提取见解所需的认知负担，让数据更易于理解。然而，创建可视化是一项复杂的创意任务，涉及多个步骤。首先，用户必须熟悉数据集（内容和语义），并列出一系列可利用数据解决的相关目标或假设。接下来，用户必须为每个目标选择合适的可视化表示（图形标记、变换和布局）。最后，用户必须通过编写代码或使用现有的直接操作界面来实现可视化。这些步骤都需要专业知识，而且可能既繁琐又容易出错，这对可视化经验有限的用户（新手用户）来说尤为如此。现有研究试图通过自动化可视化（AUTOVIZ）创建过程来应对这些挑战，即根据数据集生成可视化（Podo 等人，2023年）。自动化可以分为两种模式：i）全自动——系统自动生成与数据相关的可视化；ii）半自动化——用户指定目标，系统生成能够实现这些目标的可视化。前一种模式对不熟悉数据的用户很有帮助，后一种模式则适用于对数据和可视化任务有一定了解的用户。

因此，成功的 AUTOVIZ 工具必须擅长完成多个子任务：理解数据的语义，列举相关的可视化目标，并生成满足这些目标的语法、设计、任务和感知要求的可视化规范（Podo 等人，2023）。此外，考虑到目标用户群体是新手用户，此类工具必须通过提供自然语言（NL）交互模态来支持用户（Mitra 等人，2022；Narechania 等人，2020；Chen 等人，2022），赋予用户控制系统行为的手段，并提供帮助理解和调试／验证系统行为的分析工具。尽管相关研究已处理 AUTOVIZ 任务的若干方面，但仍存在一些已知局限（Podo 等人，2023），例如：(i) 依赖覆盖范围有限、难以设计且维护繁琐的启发式方法（Wongsuphasawat 等人，2017年）；(ii) 需要用户进行大量交互才能生成可视化（Wongsuphasawat 等人，2017年；Moritz 等人，2018）；(iii) 实现的自动化方法对系统输入和输出的控制有限（Dibia 和 Demiralp，2019）；(iv) 每个子任务都需要特定于语法（或图表类型）的训练数据和模型架构（Dibia 和 Demiralp，2019；Luo 等人，2018）；(v) 未考虑信息图等其他图表表示格式。

与此同时，大型基础模

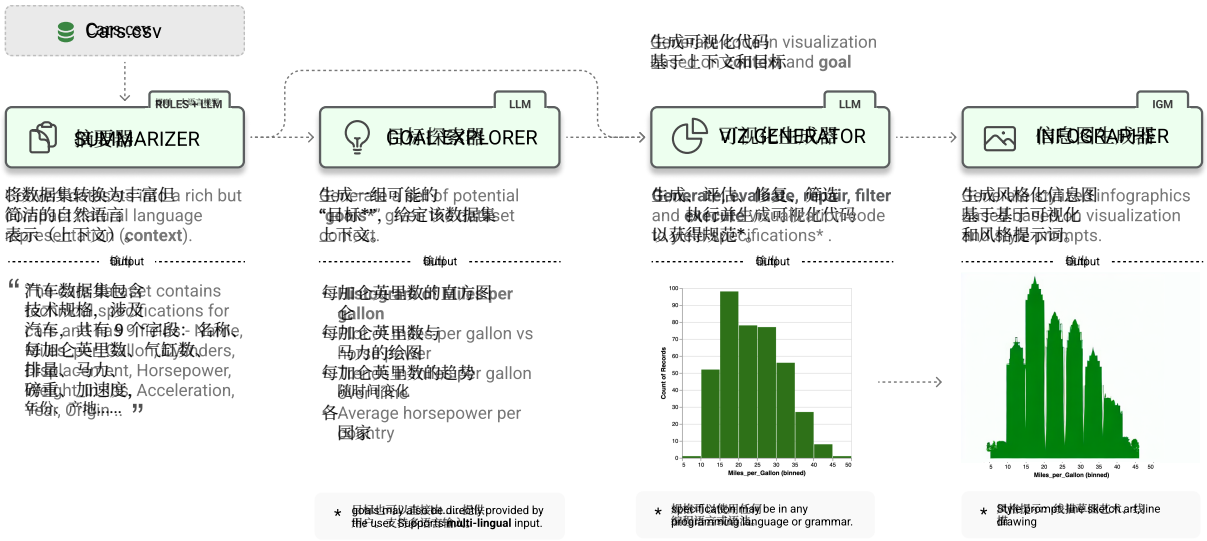

图 1：LIDA 通过 4 个模块生成可视化和信息图——数据摘要、目标探索、可视化生成和信息图生成。图中展示了各模块的示例输出。

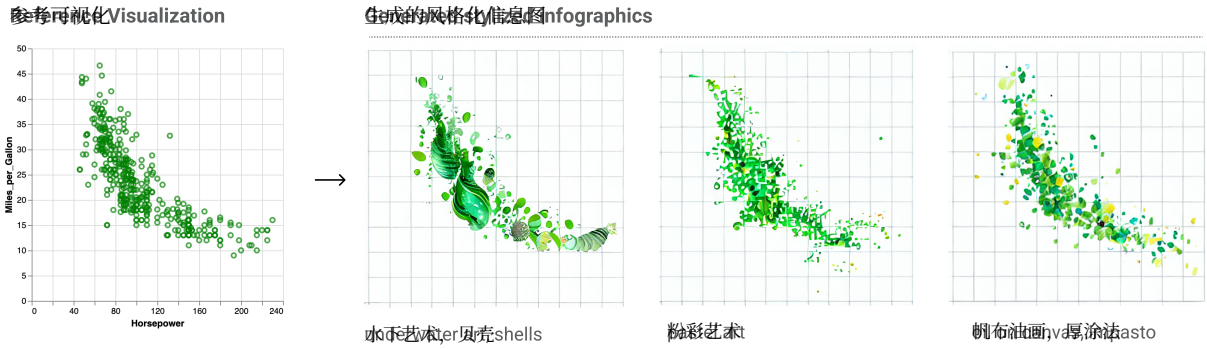

图 2：LIDA 生成的忠实于数据的信息图及其配套风格提示词。

型（Bommasani 等人，2021年）在多种创意任务上展现出最先进水平的表现，例如多语言文本生成、代码生成、图像描述、图像生成和图像编辑。在本研究中，我们认为，可以将这些模型的强大能力整合起来，以解决 AUTOVIZ 任务，同时应对现有方法的局限。本研究的贡献如下：

- 我们提出了一种新颖的多阶段模块化方法（图 1），利用大语言模型[1]自动生成数据可视化和信息图。具体而言，我们：(i) 高效地将数据集表示为自然语言摘要，作为上下文，为大语言模型处理可视化任务提供依据。(ii) 利用大语言模型生成一组可视化目标。重要的是，我们运用提示工程，引导模型生成符合要求的可视化，并遵循最佳实践（见附录C）。(iii) 使用大语言模型，根据生成的目标（或用户提供的目标）生成不依赖语法的可视化规范。(iv) 提供一个混合界面，既支持传统的直接操控控件（例如，手动选择要探索的字段），也提供丰富的**多语言**自然语言界面，以满足不同技能和经验水平用户的需求。(v) 应用文本条件图像生成模型（IGM），生成风格化信息图，使其既能传达信息（总体上忠实于数据），又美观、易记且引人入胜（见第2.3节）。

•我们介绍了用于评估大语言模型赋能的可视化工具的指标，包括用于衡量流水线可靠性的指标（可视化错误率 -可视化错误率），以及用于衡量可视化质量的指标（自评可视化质量 - 自评可视化质量）（见第4节）。

[1]本研究主要使用 OpenAI *gpt-3.5-turbo-x* 系列模型来生成文本和代码。

• 我们通过一个开源库实现了我们的方法——LIDA[2]。LIDA 提供 Python API、Web API 和一个**功能丰富的 Web 界面**，适用于研究和实际应用。

与现有的 AUTOVIZ 方法相比，LIDA 提出了一种**简化**的实现（无需针对子任务的模型）、**通用**的实现（可调整为使用任何编程语言或语法生成可视化）、**灵活**的实现（可对各个模块进行优化）以及**可扩展**的实现（随着底层 LLM 的发展，系统性能将提升）。总体而言，这些贡献为构建复杂工作流提供了基础模块，例如可视化翻译、图表问答（应用于图表可访问性）、自动化数据探索和自动化数据故事。

据我们所知，LIDA 是首个将可视化/信息图生成形式化为多步生成任务的工具，并展示了一条能够处理各种子任务的端到端流水线。

2 相关工作

LIDA 借鉴了将大型基础模型应用于跨文本和图像等模态的创意任务的研究，以及自动生成可视化和信息图方面的进展。

2.1 面向创意任务的基础模型

基于大型 Transformer 的模型（Vaswani 等人，2017年）在海量数据（数 TB 的文本和图像）上训练后取得的进展，带来了范式转变：单个模型在文本、图像、音频和视频等多种数据模态上，均展现出最先进水平的任务表现。这些模型也称为基础模型（Bommasani 等人，2021年），已被证明能有效用于各种人类创造性任务。GPT-3 系列（Brown 等人，2020）、OPT（Zhang 等人，2022年）、PaLM（Chowdhery 等人，2022年）和 LaMDA（Cohen 等人，2022年）等大语言模型能够学习语言的复杂语义，因此可有效完成文本摘要和问答等任务。Codex（Chen 等人，2021年）、AlphaCode（Li 等人，2022年）和 InCoder（Fried 等人，2022年）等代码大语言模型，在一系列代码智能任务上展现出最先进水平的表现。最后，CLIP（Radford 等人，2021年）、DALLE（Rameshetal.，2022，2021年）和潜在扩散（Rombach 等人，2022）已展现出在图像描述、图像编辑和图像生成等图像生成任务上的最先进水平能力。

在这项工作中，我们借鉴了程序辅助语言模型（Gao 等人，2022年）的思路：在这种设置中，大语言模型生成程序作为中间推理步骤，但将求解步骤交由 Python 解释器等运行时完成。我们利用大语言模型的语言建模能力来生成语义上有意义的可视化目标，并利用其代码编写能力来生成可视化代码，再通过编译生成可视化。这些可视化（图像）随后被用作图像生成模型的输入，以生成风格化信息图。

2.2 自动化可视化（AUTOVIZ）

现有的 AUTOVIZ 研究探索了多种方法，例如启发式方法、任务分解或基于学习的方法。基于启发式的方法通过探索数据属性，生成潜在可视化的搜索空间（Wongsuphasawat 等人，2017年）；再依据质量属性对这些可视化进行排序（Luo 等人，2018；Moritz 等人，2018），并呈现给用户。例如，DeepEye（Luo 等人，2018）会枚举所有可能的可视化，并使用二叉决策树分类器将其分类或排序为“好”或“差”；而 Voyager（Wongsuphasawat 等人，2017年）则使用启发式方法枚举可视化空间。不过，启发式方法维护起来可能很繁琐，对可视化空间的覆盖可能不足，也无法利用现有数据集中编码的信息。较新的研究探索了任务分解方法，即将 AUTOVIZ 流程拆分为多个任务，分别通过专用工具解决，再汇总生成可视化（Narechania 等人，2020；Chen 等人，2022；Wang 等，2022b）。例如，NL4DV（Narechania 等人，2020）实现了一个自定义查询引擎，用于解析自然语言查询、识别属性和任务，并生成 Vega-Lite 规范。任务分解方法的局限在于，其性能受制于各步骤的实现效果（例如，NL4DV 中用于消歧自然语言查询的模型存在局限（Narechania 等人，2020））。最后，端到端的基于学习的方法

[2]https://microsoft.github.io/lida/.

旨在自动学习从数据直接映射到生成的可视化的方法。例如，Data2Vis（Dibia 和 Demiralp，2019年）（与本研究最相关的工作）采用序列到序列模型，通过学习从数据集中采样的原始 JSON 数据到 Vega-Lite（Satyanarayan 等人，2017年）规范的映射，隐式地处理AUTOVIZ 子任务。目前学习方法的一些局限是：它们仅适用于单一语法；对每种受支持的语法，都需要定制模型、定制的配对训练数据和训练目标（Dibia 和 Demiralp，2019年；Luo 等人，2018；Chen 等人，2022年）；此外，它们没有提供生成信息图的途径，也没有提供对可视化输出进行细粒度控制的机制，或稳健的错误检测与恢复策略。

LIDA 通过多种方式解决这些局限：(i) 利用大语言模型从大规模语言和代码数据集中学到的模式，并将这些知识应用于各个子任务。(ii) 提供一条不依赖特定语法的统一流水线，可使用多种编程语言和可视化语法生成可视化。(iii) 支持通过自然语言控制生成的可视化。(iv) 利用大语言模型涌现出的能力，例如思维链推理，以提高生成文本/代码的可靠性（Kojima 等人，2022年；Wei 等人，2022年；Shi 等人，2022a）、模型校准（Kadavath 等人，2022年）（对可视化正确性概率的预测），以及在结果排序/筛选中的自一致性（Wang 等人，2022a）。(v) 提供一种机制，用于生成忠实于数据且美观的信息图。(vi) 支持全自动模式：使用大语言模型发现有意义的目标/假设（要可视化的字段、要提出的问题）；也支持半自动模式：由用户提供假设，再据此生成可视化。

通过将可视化/信息图生成视为生成任务，并将核心问题求解交由大语言模型和 IGM 处理，LIDA 简化了此类系统的设计与维护。

2.3 信息图生成

信息图（信息图形）是一种视觉人工制品，旨在运用视觉图像和装饰元素传达复杂的数据驱动叙事（Harrison 等人，2015年）。现有研究表明，信息图美观、引人入胜且更容易令人记住（Tyagi 等人，2021年；Harrison 等人，2015年；Haroz 等人，2015年），而且不会给用户带来额外成本（Haroz 等人，2015年）。这些特性推动了信息图在时尚、广告、商业和一般交流等领域的应用。然而，对于内容创作者而言，制作能够传达数据洞察的信息图可能是一项繁琐的工作，往往需要掌握多种工具和领域的技能。关于信息图生成的研究主要探索了象形图的制作（Haroz 等人，2015年）：用生成的图像替换传统图表中的图形标记，以及学习从现有象形图中提取或迁移风格（Shi 等人，2022b）。在本研究中，我们将这一领域拓展至探索如何使用 DALLE（Ramesh 等人，2022年，2021年）和潜在扩散（Rombach 等人，2022年）等大型图像生成模型，根据自然语言风格描述生成视觉标记以及完整的信息图。这种方法还支持用户生成视觉风格，并可根据用户对调色板、视觉风格、字体等方面的偏好，对可视化进行个性化定制。

3 LIDA 系统

LIDA 由 4 个核心模块组成：摘要-器、目标探索器、可视化生成器和 信息图生成器 (见图 1)。每个模块均在 LIDA GitHub 仓库中实现为 Python 库，并可选配用户界面（见附录A）。

3.1 摘要器

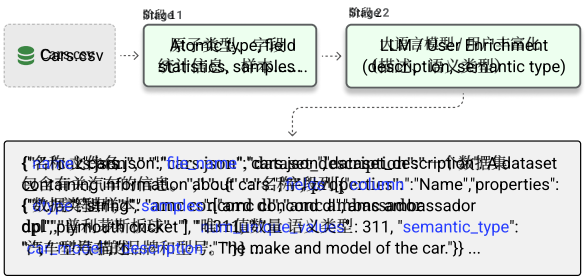

图 3：摘要器根据提取的数据属性（原子类型、字段统计）以及可选的大语言模型增强（预测的字段描述、语义类型）构建自然语言摘要。

大语言模型能够进行零样本预测，只需很少甚至无需示例引导即可解决多种任务。然而，它们可能会产生幻觉，例如生成未以训练数据为依据的文本

或当前任务。一种解决方法是增强（Mialon 等人，2023年）为大语言模型提供基础上下文。因此，摘要器的目标是为给定数据集生成一份**信息密度高**但**简洁的** [3] 摘要，作为对可视化任务有用的基础上下文。有用的上下文被定义为包含分析师理解数据集以及能够在其上执行的任务所需信息的上下文。该摘要分两个阶段实现（见图3）

阶段 1 - 基础摘要生成：我们使用规则提取数据集属性，包括原子类型（例如整数、字符串、布尔值），并借助 pandas 库（McKinney，2010年）提取一般统计信息（最小值、最大值、唯一值数量），以及每列随机抽取的n 个非空样本。

阶段 2 - 摘要增强：基础摘要可以选择由大语言模型或用户通过LIDA 界面进行丰富化，以纳入数据集的语义描述（例如，关于汽车技术规格的数据集）和字段描述（例如每辆汽车的每加仑英里数），以及字段语义类型预测（Zhang 等人，2019年）。

3.2 目标探索器

该模块根据摘要器生成的摘要，生成数据探索目标。我们将目标生成表述为一个多任务生成问题，其中大语言模型必须生成一个问题（假设）、一个可视化来回答该问题，以及理由（见图4）。我们发现，要求大语言模型给出理由，能够生成语义上更有意义的目标。

{ "question": "What is the distribution of Miles_per_Gallon?",
"visualization": "Histogram of Miles_per_Gallon",
"rationale": "This tells us about the fuel efficiency of the cars in the dataset and how it is distributed." }

图 4：LIDA 生成的目标是一个 JSON 数据结构，其中包含一个问题、一项可视化和一条理由。

3.2.1 VISGENERATOR

VISGENERATOR 生成可视化规范，由 3 个子模块组成——代码框架构造器、代码生成器和代码执行器。

代码框架构造器：实现了一个代码框架库，这些代码框架对应于编程

[3]注意：摘要必须简洁，以最大限度地利用大语言模型有限的上下文令牌预算。

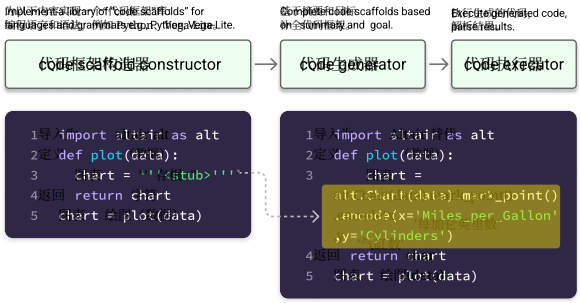

图 5：VISGENERATOR 模块构建可视化代码框架，填充受约束的部分（<存根 >），并执行该代码框架。

语言和可视化语法，例如，Python 脚手架支持 Matplotlib、GGPlot、Plotly、Altair、Seaborn 和 Bokeh 等语法。每个脚手架都是一个可执行程序，它会：i.) 导入相关依赖项；ii.) 定义一个空函数桩，该函数返回可视化规范（见图5a）。

代码生成器：接收一个代码框架、数据集摘要和可视化目标，并构建一个提示词。随后使用一个大语言模型（应用于中间填充模式（Bavarian 等人，2022年））来生成n 个候选可视化代码规范。

代码执行器：对[4] 代码规范进行后处理和执行，并过滤结果。LIDA 实现了多种过滤机制来检测错误，但各自存在延迟方面的权衡：(i) 在高温度下为n 生成大量样本，并丢弃无法编译的候选项。(ii) 在大语言模型中应用自洽性（Wang 等人，2022a），生成多个候选项，并选择共识度最高的方案。(iii) 为所有候选项生成正确性概率（Kadavath 等人，2022年），并选择概率最高的候选项。请注意，后两种方法计算开销较大（需要对大语言模型进行多次前向传播），不适用于实时应用。最终输出是一系列可视化规范（代码）及其对应的栅格图像。

3.2.2 VIZOPS - 对生成的可视化进行操作

鉴于 LIDA 将可视化表示为代码，VISGENERATOR 还实现了用于对这种表示执行操作的子模块。

基于自然语言的可视化优化：提供对话式 API，以便迭代地

[4]建议在沙箱环境中执行。

优化生成的代码（例如，将图表翻译成印地语.....放大 *50%* 等），随后即可执行代码以生成新的可视化。

可视化解释与无障碍访问：生成自然语言解释（有助于调试和理解数据），以及无障碍描述（有助于支持视力障碍用户）。

可视化代码自评估与修复：应用大语言模型对生成的代码进行自我评估，从多个维度进行评估（见第4.1.2节）。

可视化推荐：给定一些上下文（目标或现有可视化），向用户推荐其他可视化（例如用于比较或提供额外视角）。

3.3 信息图生成器

该模块负责根据VISGENERATOR模块的输出生成风格化图形（见图2）。该模块实现了一个以自然语言描述的视觉风格库，并将这些风格直接应用于可视化图像。需要注意的是，用户可以编辑该风格库。这些风格通过扩散模型的文本条件图像到图像生成能力应用于信息图生成（Rombach 等，2022年），具体使用 *Peacasso* 库 API 实现（Dibia，2022年）。随后可选择执行后处理步骤，以改善生成的图像（例如，将坐标轴替换为可视化中的正确数值、移除网格线以及锐化边缘）。

3.4 用户界面

LIDA 实现了一个通过 REST 和 WebSocket API 与核心模块通信的用户界面。该用户界面包含多个视图。

数据上传与摘要：此视图允许用户上传数据集，并通过表格视图查看数据集中的部分行。上传数据会触发对 SUMMARIZER 和 GOAL EXPLORER 模块的调用，并显示数据集的摘要及潜在目标列表。此视图还允许用户选择性地注释和优化生成的摘要，或整理数据集中使用的字段。

可视化视图：用户可以选择用自然语言提供一个可视化目标（例如，“各国家的燃油效率是多少？”），也可以选择一个生成的目标，随后系统会显示生成的可视化。对于每个可视化，系统都会将模型的中间输出（底层数据摘要、可视化规范、代码脚手架）作为解释展示，以帮助用户理解并进行调试（见图9）。此视图还实现了第3.2.2节所述的 VIZOPS 功能（例如，图10展示了用于评估可视化的界面）。请注意，自然语言界面继承了底层大语言模型的多语言能力，因此支持多语言自然语言交互。

总体而言，这些模块相结合，构成了一个能够隐式处理一系列数据可视化操作的系统，例如数据转换、编码、标记选择、样式、布局和注释（Wang 等人，2022b）。

4 评估

4.1 评估指标

我们对 LIDA 的初步评估聚焦于两个高层指标——可视化错误率（VER），用于提供有关 LIDA 流水线可靠性 的信号；以及自评可视化质量（SEVQ），用于评估生成的可视化的质量 。

4.1.1 可视化错误率（VER）

可视化错误率是指生成的可视化中出现代码编译错误的比例。该指标能为 LIDA 流水线的可靠性以及系统变更（例如提示工程或脚手架更新）的影响提供重要洞察。

$$\text{VER} = \frac{E}{T} * 100$$

其中：- E = 表示出现代码编译错误的生成的可视化数量；- T =表示生成的可视化总数。

4.1.2 自评可视化质量（SEVQ）

近期研究表明，GPT-4 等大语言模型编码了广泛的世界知识（OpenAI，2023年），能够评估自身输出的质量（Kadavath 等人，2022年；Lin 等人，2022年），并且在摘要等任务中能够近似人类的判断（Liu 等人，2023年）。我们将 GPT-3.5/GPT-4 应用于可视化任务时的观察结果也显示出类似的表现。具体而言，GPT-4 已学会编码一些可视化最佳实践，并能在多个维度上据此生成对可视化代码的评析。因此，为了评估可视化质量，我们通过使用 GPT-4 评估生成的可视化，计算 SEVQ 指标。具体来说，我们要求 GPT-4

对生成的可视化代码进行评分（给出 1–10 的数值和理由），涵盖 6 个维度：代码准确性、数据转换、目标符合度、可视化类型、数据编码和美学。这些维度以现有的可视化生成/推荐文献为依据，例如Wang 等人（2022b）概述了 6 项可视化任务，包括数据转换、编码、图形标记、样式、布局和注释；而（Moritz 等人，2018）则针对表达力（是否传达了数据中的事实）和有效性（与其他可视化相比，信息是否更容易被感知）这两项标准，归纳了可视化质量方面的约束。各个维度所用提示的更多细节见附录B。

4.2 评估基准设置

我们的初始基准基于从 vega 数据集仓库获取的 57 个数据集[5]。对于每个数据集，LIDA 需要使用多种语法，为每个数据集生成 5 个目标，并为每个目标生成 1 个可视化图表[6]。为确保结果可复现，我们将温度 = 0和样本数量$n = 1$ 设置为大语言模型。生成的评估可视化图表集可在LIDA 项目页面上查看。

4.3 评估与消融研究结果

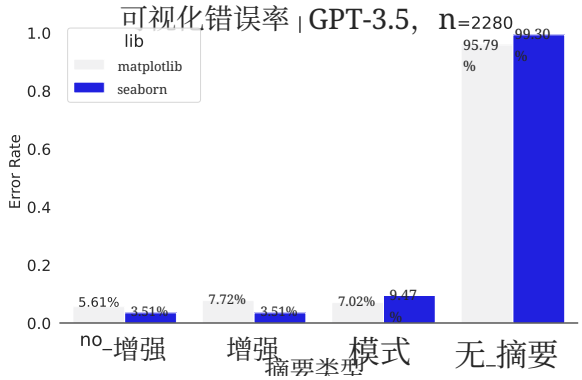

图6：数据摘要策略对可视化错误率（VER）指标影响的消融研究结果。

总体而言，我们发现 LIDA 能够以较低的错误率生成可视化（VER 为 = 3.5%）。我们还进行了消融研究，以了解摘要器在以下几种条件下的影响——（i）无_增强：不进行丰富化的基础摘要（见第3.1节）；（ii）增强：使用大语言模型进行丰富化的摘要；（iii）模式：仅包含字段名称，即以模式作为摘要；（iv）无_摘要：不提供摘要。结果表明，与仅将字段名称（模式）作为摘要相比，加入摘要能够降低错误率。我们还发现，使用大语言模型丰富化基础摘要对可视化错误率的影响较小（不同可视化语法下有所差异）；而像 Seaborn 这样表达力强且表示充分的语法，其可视化错误率较低。这些结果总结于图6中。我们还发现，SEVQ指标有助于识别生成的可视化中的语义质量问题。例如，图10展示了一个示例：用户请求饼图，而LIDA 自我评估模块使用SEVQ指标对该可视化提出批评，并说明条形图为何更有效（见图10），同时还提供了自动修复可视化的选项。

5 结论

在这项工作中，我们将可视化生成表述为一个多阶段文本（和代码）生成问题，可借助大语言模型加以解决。我们推出了 LIDA——一个用于自动生成不受语法限制的可视化和信息图的工具。LIDA 解决了当前自动可视化系统的若干局限：根据数据集自动生成假设/目标、提供可控的可视化生成与优化对话界面、通过同一条流水线支持多种可视化语法，以及生成信息图。与先进系统相比，LIDA 表现出色（参见生成的可视化示例图库）；它简化了系统实现，并利用大语言模型强大的语言建模与代码生成能力，隐式地解决复杂的可视化子任务。最后，我们为大语言模型驱动的可视化工具引入了用于评估可靠性的指标（可视化错误率 VER）和用于评估可视化质量的指标（自评可视化质量 SEVQ）。我们希望，LIDA 中实现的模块能够成为实用的构建模块，助力实现复杂的创意工作流，例如可视化转换、图表问答（可应用于提升图表可访问性）、自动化数据探索和自动化叙事。

[5] https://github.com/vega/vega-datasets [6] 每个步骤中，LIDA 仅尝试一次。理论上，通过递归应用可视化自评估和自修复模块，可以将错误率降至零。

6 局限性

尽管 LIDA 在支持用户创作可视化和信息图方面取得了显著进展，但仍存在若干局限，也为未来研究提供了自然的方向。

低资源语法：LIDA 所引入的问题表述依赖于底层大语言模型具备一定的可视化语法知识，这些知识以训练数据集中的文本和代码形式呈现（例如 Altair、Vega、Vega-Lite、GGPlot、Matplotlib 的示例，以及GitHub、Stackoverflow 等平台中的相关内容）。对于在这些数据集中的体现不够充分的可视化语法（例如 Tableau、PowerBI 等具有图形用户界面而非代码表示形式的工具），如果没有额外的模型微调或转换，LIDA 的性能可能会受到限制。此外，对于超出特定语法表达能力的复杂任务（例如需要进行复杂数据转换的任务），其性能也可能受限。未来研究需要：i）研究任务消歧等策略的影响；ii）分析任务复杂度以及编程语言/语法选择对性能的影响。**部署与延迟**：大语言模型（例如本研究使用的 GPT-3.5）计算成本高昂，需要大量计算资源才能以低延迟进行部署。这些成本可能使其不适用于实际应用。此外，当前设置包含代码执行步骤，该步骤对于验证非常有价值，但会增加部署复杂度（需要沙箱）。因此，可以考虑：i）在精选数据集上对更小但能力强大的大语言模型进行微调（Touvron 等人，2023年），该数据集包含编程语言和可视化语法；ii）设计漏洞缓解方法，例如限制程序作用域，或仅为可视化语法编译器生成输入参数。

解释系统行为：本文讨论的方法简化了可视化创作系统的设计，但也继承了大语言模型相关的可解释性挑战。尽管LIDA 提供模型的中间输出（例如生成的代码和规范）作为解释，并提供对生成代码的事后解释（见第3.2.2节），但仍需进一步研究如何解释系统行为（以及在何种情况下需要解释），并向用户提供可操作的反馈。

系统评估：对大语言模型在创造力任务上的基准测试颇具挑战。尽管当前研究提出了用于评估可靠性（可视化错误率 VER）和可视化质量（SEVQ）的指标（见4节），但仍需针对各种数据集和可视化语法开展更全面的基准测试。此外，还存在以下研究机会：i.) 研究并量化大语言模型在编码 和应用 可视化最佳实践方面的能力；ii.) 开展实证研究，评估模型行为、梳理失败案例并提出缓解措施；iii.) 定性研究 LIDA 等工具在用户创作可视化时对其创造力的影响。

致谢

本稿受益于 HAX 小组（Saleema Amershi、Adam Fourney、Gagan Bansal）、VIDA 小组（Steven Drucker、Dan Marshall）、Bongshing Lee、Rick Barraza 以及 Microsoft Research 的其他成员所提出的意见和参与的讨论。

参考文献

Mohammad Bavarian、Heewoo Jun、Nikolas Tezak、JohnSchulman、Christine McLeavey、Jerry Tworek 和 Mark Chen。2022年。高效训练语言模型以进行中间填充。*arXiv* 预印本 *arXiv:2207.14255*。

Rishi Bommasani、Drew A Hudson、Ehsan Adeli、Russ Altman、Simran Arora、Sydney von Arx、Michael S Bernstein、Jeannette Bohg、Antoine Bosse-lut、Emma Brunskill 等。2021年。论基础模型的机遇与风险。*arXiv* 预印本 *arXiv:2108.07258*。

Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell 等。2020。语言模型是少样本学习器。神经信息处理系统进展，33:1877–1901。

Mark Chen、Jerry Tworek、Heewoo Jun、Qiming Yuan、Henrique Ponde de OliveiraPinto、JaredKaplan、Harri Edwards、Yuri Burda、Nicholas Joseph、Greg Brockman 等。2021年。评估基于代码训练的大语言模型。*arXiv* 预印本 *arXiv:2107.03374*。

Qiaochu Chen、Shankara Pailoor、Celeste Barnaby、AbbyCriswell、ChenglongWang、GregDurrett 和 Isil Dillig。2022年。根据自然语言查询进行可视化的类型导向综合。*ACM* 编程语言学报，6(OOPSLA2):532–559。

Aakanksha Chowdhery、Sharan Narang、Jacob Devlin、Maarten Bosma、Gaurav Mishra、Adam Roberts、Paul Barham、Hyung Won Chung、Charles Sutton、Sebastian Gehrmann 等。2022年。PaLM：利用路径扩展语言建模。*arXiv* 预印本 *arXiv:2204.02311*。

Aaron Daniel Cohen、Adam Roberts、Alejandra Molina、Alena Butryna、Alicia Jin、Apoorv Kulshreshtha、Ben Hutchinson、Ben Zevenbergen、Blaise Hilary Aguera-Arcas、Chung ching Chang、Claire Cui、Cosmo Du、Daniel De Freitas Adiwardana、De- hao Chen、Dmitry (Dima) Lepikhin、Ed H. Chi、Erin Hoffman-John、Heng-Tze Cheng、Hongrae Lee、Igor Krivokon、James Qin、Jamie Hall、Joe Fen- ton、Johnny Soraker、Kathy Meier-Hellstern、Kris- ten Olson、Lora Mois Aroyo、Maarten Paul Bosma、Marc Joseph Pickett、Marcelo Amorim Menegali、Marian Croak、Mark Díaz、Matthew Lamm、Maxim Krikun、Meredith Ringel Morris、Noam Shazeer、Quoc V. Le、Rachel Bernstein、Ravi Rajakumar、Ray Kurzweil、Romal Thoppilan、Steven Zheng、Taylor Bos、Toju Duke、Tulsee Doshi、Vincent Y. Zhao、Vinodkumar Prabhakaran、Will Rusch、YaGuang Li、Yanping Huang、Yanqi Zhou、Yuanzhong Xu 和 Zhifeng Chen。2022年。LaMDA：用于对话应用的语言模型。载于*arXiv*。

Victor Dibia。2022年。面向集成图像生成模型的系统的交互设计：Peacasso 案例研究。

Victor Dibia 和 Çaˇgatay Demiralp。2019年。Data2Vis：使用序列到序列循环神经网络自动生成数据可视化。*IEEE Computer Graphics and Applications*，39(5)：33–46。

Daniel Fried、Armen Aghajanyan、Jessy Lin、Sida Wang、Eric Wallace、Freda Shi、Ruiqi Zhong、Wen-tau Yih、Luke Zettlemoyer 和 Mike Lewis。2022年。InCoder：用于代码填充与合成的生成模型。*arXiv* 预印本 *arXiv:2204.05999*。

Luyu Gao、Aman Madaan、Shuyan Zhou、Uri Alon、Pengfei Liu、Yiming Yang、Jamie Callan 和 Graham Neubig。2022年。PaL：程序辅助语言模型。*arXiv* 预印本 *arXiv:2211.10435*。

Steve Haroz、Robert Kosara 和 Steven L Franconeri。2015年。Isotype可视化：工作记忆、表现与象形图参与度。载于第*33*届 *ACM* 计算系统中的人为因素年会论文集，第1191–1200页。

Lane Harrison、Katharina Reinecke 和 Remco Chang。2015年。信息图美学：为第一印象而设计。载于第*33*届 *ACM* 计算系统中的人为因素年会论文集，第1187–1190页。

Saurav Kadavath、Tom Conerly、AmandaAskell、Tom Henighan、Dawn Drain、Ethan Perez、Nicholas Schiefer、Zac Hatfield Dodds、Nova DasSarma、Eli Tran-Johnson 等。2022年。语言模型（大体上）知道自己知道什么。*arXiv* 预印本 *arXiv:2207.05221*。

Takeshi Kojima、Shixiang Shane Gu、Machel Reid、Yutaka Matsuo 和 Yusuke Iwasawa。2022年。大型语言模型是零样本推理器。*arXiv* 预印本 *arXiv:2205.11916*。

Yujia Li、David Choi、Junyoung Chung、Nate Kushman、Julian Schrittwieser、Rémi Leblond、Tom Eccles、James Keeling、Felix Gimeno、Agustin Dal Lago 等。2022年。使用 AlphaCode 实现竞赛级代码生成。*arXiv* 预印本 *arXiv:2203.07814*。

Stephanie Lin、Jacob Hilton 和 Owain Evans。2022年。教导模型用文字表达其不确定性。*arXiv*预印本 *arXiv:2205.14334*。

Yang Liu、Dan Iter、Yichong Xu、Shuohang Wang、Ruochen Xu 和 Chenguang Zhu。2023年。Gpteval：使用 GPT-4 进行自然语言生成评估，实现更好的人类对齐。*arXiv* 预印本 *arXiv:2303.16634*。

Yuyu Luo、Xuedi Qin、Nan Tang、Guoliang Li 和 Xinran Wang。2018。Deepeye：通过关键词搜索创建优秀的数据可视化。载于*2018*年数据管理国际会议论文集，SIGMOD，第 1733–1736 页。

Wes McKinney。2010。用于 Python 统计计算的数据结构。载于第 *9* 届 *Python* 科学会议论文集，第 51–56 页。

Grégoire Mialon、Roberto Dessì、Maria Lomeli、Christo- foros Nalmpantis、Ram Pasunuru、Roberta Raileanu、Baptiste Rozière、Timo Schick、Jane Dwivedi-Yu、Asli Celikyilmaz 等。2023年。增强型语言模型：一项综述。*arXiv* 预印本 *arXiv:2302.07842*。

Rishab Mitra、Arpit Narechania、Alex Endert 和 John Stasko。2022年。促进可视化自然语言界面中的对话交互。载于*2022 IEEE* 可视化与视觉分析会议（*VIS*），第 6–10 页。IEEE。

Dominik Moritz、Chenglong Wang、Greg L Nelson、Halden Lin、Adam M Smith、Bill Howe 和 Jef- frey Heer。2018。将可视化设计知识形式化为约束：Draco 中可操作且可扩展的模型。*IEEE* 可视化与计算机图形学汇刊，25(1):438–448。

Arpit Narechania、Arjun Srinivasan 和 John Stasko。2020。NL4DV：一种用于根据自然语言查询生成数据可视化分析规范的工具。*IEEETransactions on VisualizationandComputer Graphics*，27(2)：369–379。

OpenAI。2023年。GPT-4 技术报告。

Luca Podo、Bardh Prenkaj 和 Paola Velardi。2023年。用于可视化推荐系统的机器学习：开放挑战与未来方向。*arXiv* 预印本 *arXiv:2302.00569*。

Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark 等。2021年。从自然语言监督中学习可迁移视觉模型。载于国际机器学习会议，第 8748–8763 页。PMLR。

Aditya Ramesh、PrafullaDhariwal、AlexNichol、Casey Chu 和 Mark Chen。2022年。基于 CLIP潜变量的分层文本条件图像生成。*arXiv* 预印本 *arXiv:2204.06125*。

Aditya Ramesh、Mikhail Pavlov、Gabriel Goh、Sc ott Gray、Chelsea Voss、Alec Radford、Mark Chen 和 Ilya Sutskever。2021年。零样本文本到图像生成。载于国际机器学习会议，第 8821–8831 页。PMLR。

Robin Rombach、Andreas Blattmann、Dominik Lorenz、Patrick Esser 和 Björn Ommer。2022年。使用潜在扩散模型进行高分辨率图像合成。2022年 ieee。载于*CVF* 计算机视觉与模式识别会议 *(CVPR)*，第 10674–10685 页。

Arvind Satyanarayan、Dominik Moritz、Kanit Wong-suphasawat 和 Jeffrey Heer。2017年。Vega-lite：一种交互式图形语法。*IEEE*可视化与计算机图形学汇刊（信息可视化论文集）。

Freda Shi、MiracSuzgun、Markus Freitag、XuezhiWang、Suraj Srivats、Soroush Vosoughi、Hyung Won Chung、Yi Tay、Sebastian Ruder、Denny Zhou 等。2022a。语言模型是多语言思维链推理者。*arXiv* 预印本 *arXiv:2210.03057*。

Yang Shi、Pei Liu、Siji Chen、Mengdi Sun 和 Nan Cao。2022b。通过视觉风格迁移支持富有表现力且忠实的图像可视化设计。*IEEETransactions onVisualizationandComputerGraphics*，29(1)：236–246。

Hugo Touvron、Thibaut Lavril、Gautier Izacard、Xavier Martinet、Marie-Anne Lachaux、Timothée Lacroix、BaptisteRozière、Naman Goyal、Eric Hambro、Faisal Azhar、Aurelien Rodriguez、Armand Joulin、Edouard Grave 和 Guillaume Lample。2023年。 Llama：开放且高效的基础语言模型。

Anjul Tyagi、Jian Zhao、Pushkar Patel、Swasti Khurana 和 Klaus Mueller。2021年。以用户为中心的半自动信息图创作与推荐。*arXiv* 预印本 *arXiv:2108.11914*。

Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017年。注意力就是你所需要的一切。神经信息处理系统进展，30。

Xuezhi Wang、Jason Wei、Dale Schuurmans、Quoc Le、Ed Chi 和 Denny Zhou。2022a。自一致性改进语言模型中的思维链推理。*arXiv*预印本 *arXiv:2203.11171*。

Yun Wang、Zhitao Hou、Leixian Shen、Tongshuang Wu、Jiaqi Wang、He Huang、Haidong Zhang 和 Dongmei Zhang。2022b。迈向基于自然语言的可视化创作。*I EEE* 可视化与计算机图形学汇刊，29(1):1222–1232。

Jason Wei、Xuezhi Wang、Dale Schuurmans、Maarten Bosma、Ed Chi、Quoc Le 和 Denny Zhou。2022年。思维链提示能够激发大语言模型的推理能力。*arXiv* 预印本 *arXiv:2201.11903*。

Kanit Wongsuphasawat、Zening Qu、Dominik Moritz、Riley Chang、Felix Ouk、Anushka Anand、Jock Mackinlay、Bill Howe 和 Jeffrey Heer。2017年。Voyager 2：通过部分视图规范增强视觉分析。发表于*ACM CHI*。

Dan Zhang、Yoshihiko Suhara、Jinfeng Li、Madelon Hulsebos、Çağatay Demiralp 和 Wang-Chiew Tan。2019年。Sato：表格中的上下文语义类型检测。*arXiv* 预印本 *arXiv:1911.06311*。

Susan Zhang、Stephen Roller、Naman Goyal、Mikel Artetxe、Moya Chen、Shuohui Chen、Christopher De-wan、MonaDiab、Xian Li、Xi Victoria Lin 等。2022年。OPT：开放式预训练变换器语言模型。*arXiv* 预印本 *arXiv:2205.01068*。

A LIDA 库

LIDA 以 Python 库的形式实现，其中包含针对第3节所述各组件的模块。该库可在 GitHub 上获取[7]，并可通过 pip 安装：*pip install lida*。该库提供 Python API、用于与其他应用程序集成的 Web API，以及命令行界面。它还提供基于 Web 的用户界面，供用户与LIDA 交互（图10、9）。

```python
# pip install lida

from lida.modules import Manager

lida = Manager()
summary = lida.summarize("data/cars.csv")
goals = lida.generate_goals(summary, n= )

vis_specs = manager.generate_viz( summary=summary, goal=goals[i])
charts = manager.execute_viz( code_specs=vis_specs, data=manager.data, summary=summary)
print(charts)
```

图7：LIDA 的使用示例，展示了如何生成摘要、可视化目标和代码规范，并执行代码以生成可视化。

B 自我评估可视化质量（SEVQ）提示词

对于自评可视化质量指标，我们使用 GPT-4 通过为生成的可视化打分来评估其质量

[7] https://github.com/microsoft/lida

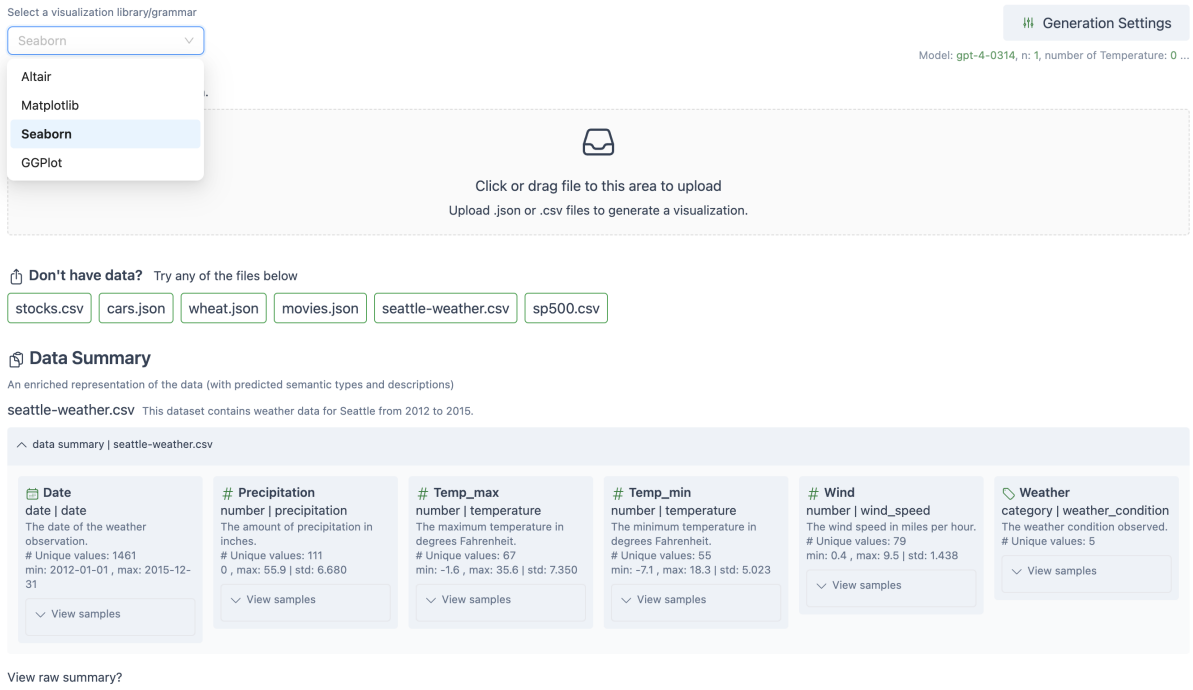

图8：在 LIDA 用户界面的数据上传部分，用户可以选择所需的语法并上传数据集。数据集上传事件会触发目标生成和可视化生成任务。

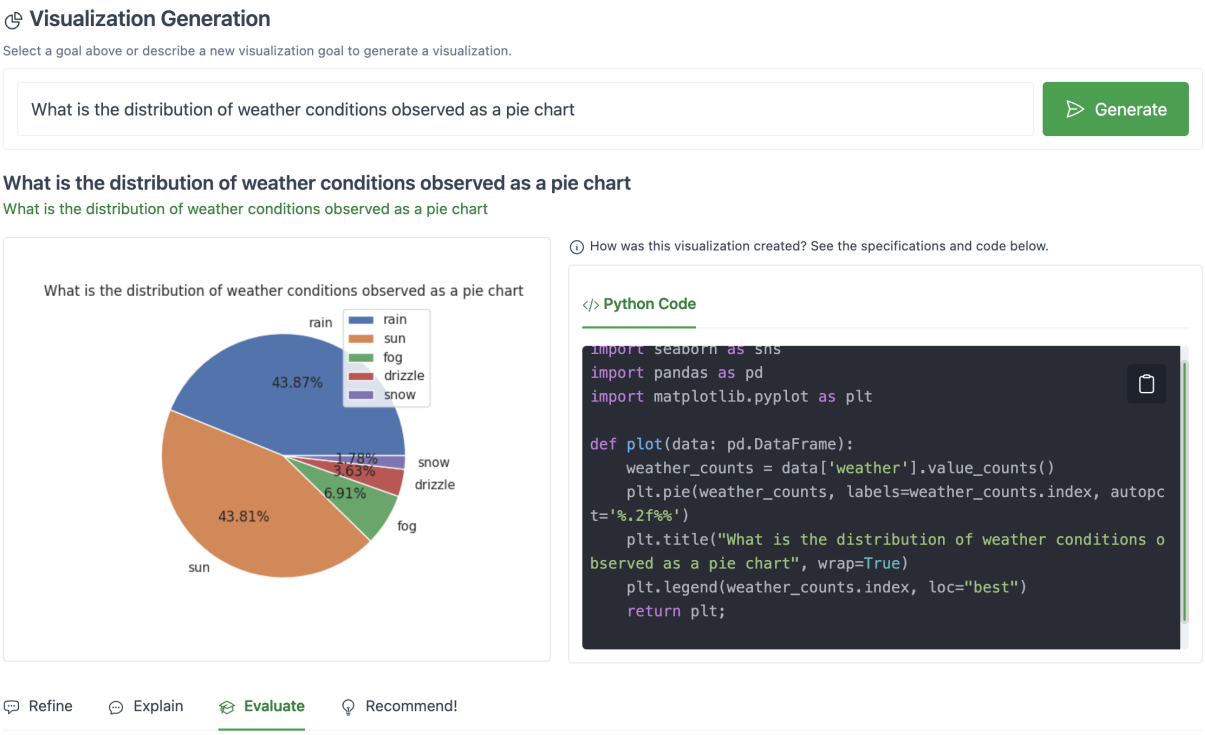

图9：LIDA 用户界面的可视化生成部分支持用户：i.) 用自然语言指定总体目标并生成可视化；ii.) 检查、编辑和执行生成的代码；iii.) 查看生成的可视化；iv.) 对生成的代码执行操作，例如优化、解释、评估并推荐可视化。

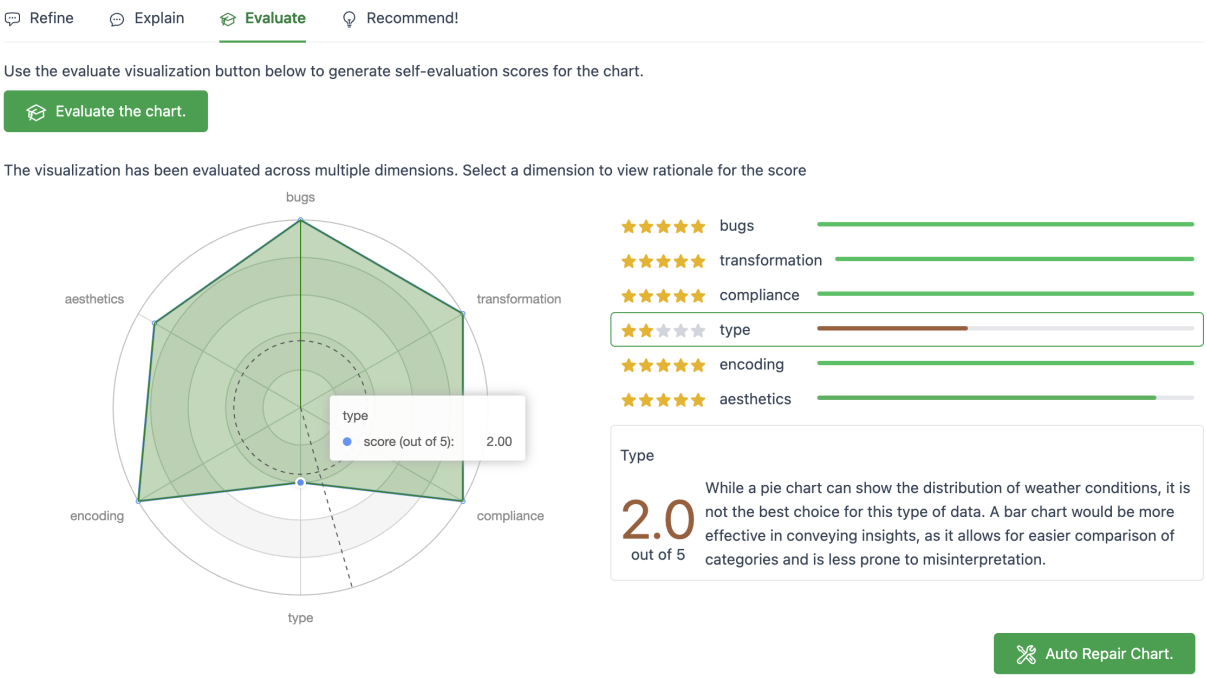

图 10：LIDA 中的自我评估模块用于评估/批判生成的可视化，并针对 6 个维度给出分数及理由。在此例中，可视化包含一个饼图，并推荐使用条形图作为替代方案。

涵盖 6 个任务维度的代码——代码准确性、数据转换、目标符合度、可视化类型、数据编码和美学。这些维度通过提示词的形式交由大语言模型[8]实现，随后由其为每个维度生成 1 到 10 之间的分数。最终的 SEVQ 分数为这 6 项分数的平均值。表 1 列出了各维度所用提示词的概要。

C 设计反思

构建一个利用基础模型（文本和图像）的系统，需要在广阔的设计空间中做出工程决策。本节将简要回顾我们为 LIDA 组件所做的一些设计选择，以及我们考虑过的权衡。

C.1 提示工程

我们探索了多种构建提示词的方法，以最大限度地提高大语言模型解决各个子任务的概率。

- 摘要器：我们发现，提升摘要的丰富程度（定性自然语言描述，包括语义类型）对提高生成目标的质量至关重要，而且可视化代码。在实现方面，我们首先手动编写了数据摘要（见第3.1节），然后通过调用大语言模型来增强摘要，并可选择由用户进一步优化摘要。

- 目标探索器：在提示中提供字段与理由通过符号关联的少样本示例

维度	提示词
代码准确-racy	代码中是否存在错误、逻辑错误，语法错误或拼写错误？这些问题有多严重？有错误吗？应该如何修复？
数据转-换	数据是否经过了适当转换，以适用于该可视化类型？
目标完-符合性	代码在多大程度上满足指定的可视化目标？
可视化type	考虑到最佳实践，所选可视化类型是否适合数据和意图？是否有一种可视化类型能够更有效地传达洞察？
数据编码——ing	数据是否针对该可视化类型进行了恰当编码？
美学	可视化的美学是否恰当地且有效，是否适合该可视化类型和数据？

表1：评估维度及相应提示草图的摘要。

[8]可在项目代码库中找到完整提示词 https://github.com/microsoft/lida。

（例如，绘制字段 X 与 Y 的直方图，以展示 X 和 Y 之间的关系），可以引导模型使用准确的数据集字段名称，并减少幻觉字段的出现。提示工程还提供了嵌入可视化最佳实践的机制，例如避免使用饼图、应用可视化最佳实践、设想你是一位经验丰富的可视化专家和数据分析师。

• VISGENERATOR：将可视化代码生成视为中间填充问题（而非自由形式的补全），可确保模型生成专注于任务的可执行代码。例如，在图5中，模型被要求仅生成代码框架中的 <存根 >部分。我们还注意到，分配给模型的自由度（例如，指定要补全框架的多少内容）会影响其添加复杂程度各异任务的能力。例如，允许模型生成数据预处理代码（并包含 statsmodels 等库）的框架，可使模型处理那些需要在生成可视化之前执行数据转换、抽样和统计分析等步骤的任务。

• 总体而言，我们发现，设置较低的温度（$t = 0$；生成最有可能的可视化），并采用按语法代码脚手架，在生成能够正确编译为可视化规范且忠实完成子任务的代码方面效果最佳。我们还探索了能够处理多项任务的提示词形式，以降低成本（延迟和计算资源）。例如，摘要增强只需一次调用，大语言模型必须生成数据集描述、字段描述和语义类型。

C.2 信息图生成

我们发现，为潜在扩散模型（图像到图像模式）设置较低的 强度参数（$0.25 <$ 强度 < 0.45），并使用简洁的风格提示词，可生成既有风格又忠实于原始可视化整体结构 的图像，从而最大限度地减少扭曲或无关图像。这种受控生成是必要的，以避免信息图中多余图像造成的干扰（Haroz 等人，2015年）。

C.3 自然语言交互

（i）混合界面：提供一种混合界面，允许用户在创建可视化时执行传统的直接操作步骤（例如，选择要使用的字段）；同时配合自然语言界面，让用户既能利用对传统可视化工具已有的心智模型，也能运用自然语言交互方式所具备的 LIDA优势。（ii）自然语言交互模式：除了生成基础可视化之外，我们还支持对生成的可视化代码进行操作（例如优化、解释、评估和推荐）。这一设计借鉴了Mitra 等人 （2022年）的洞察；他们提出了用于可视化创作的多轮对话界面，以消除歧义。

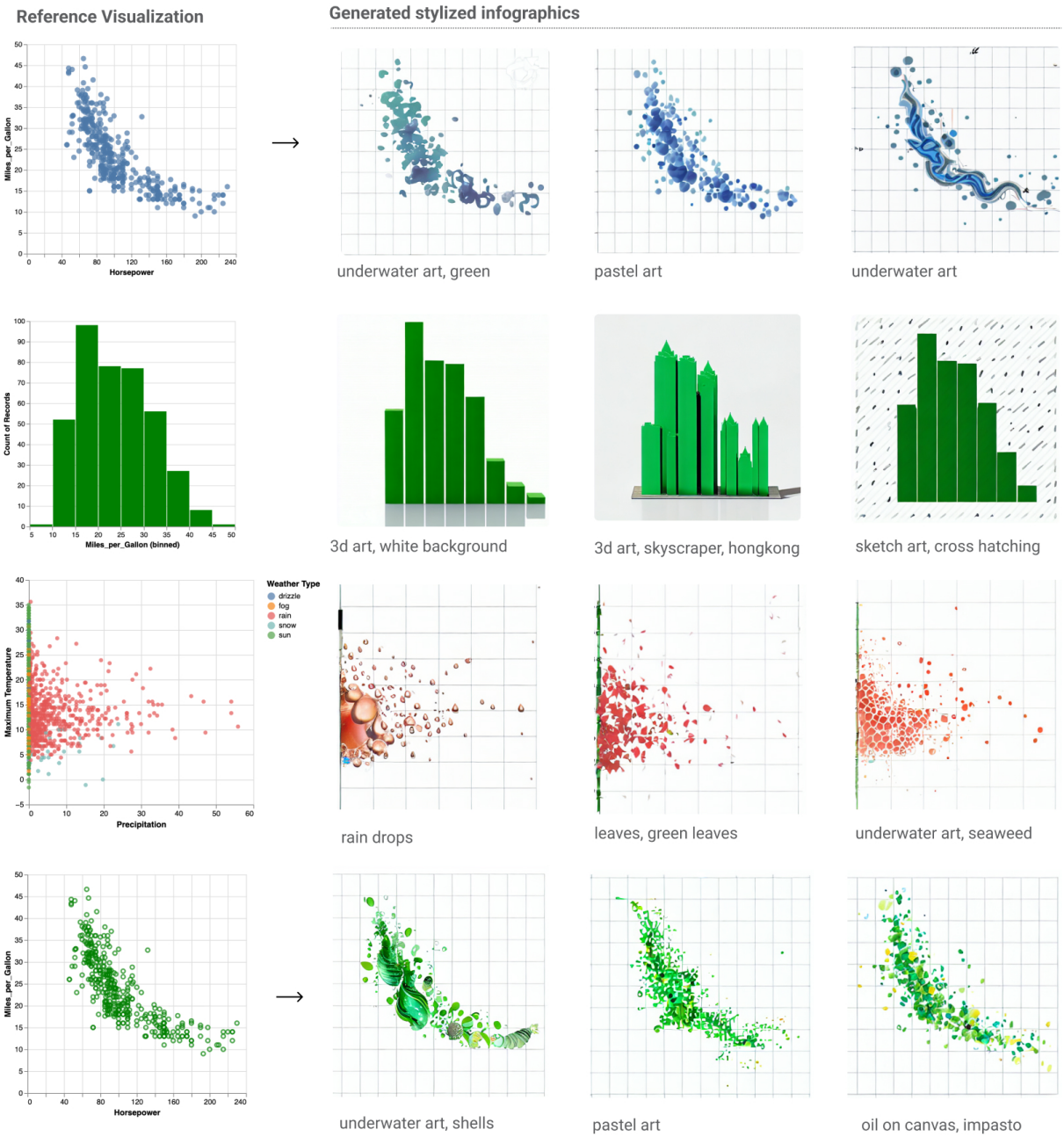

图 11：LIDA 信息图生成模块支持生成忠实于数据的信息图。每张信息图均以生成的可视化和自然语言风格标签为条件，这些标签可用于自定义图表的外观。