Scaling Text-Rich Image Understanding via Code-Guided Synthetic Multimodal Data Generation

Yue Yang*[1], Ajay Patel*[1], Matt Deitke[2], Tanmay Gupta[2], Luca Weihs[2], Andrew Head[1], Mark Yatskar[1], Chris Callison-Burch[1], Ranjay Krishna[2], Aniruddha Kembhavi[2], Christopher Clark[2]
[1]University of Pennsylvania, [2]Allen Institute for Artificial Intelligence
* Equal Contribution {yueyang1, ajayp}@seas.upenn.edu yueyang1996.github.io/cosyn

Abstract

Reasoning about images with rich text, such as charts and documents, is a critical application of vision-language models (VLMs).
推理包含丰富文本的图像（如图表和文档）是视觉-语言模型（VLMs）的一个重要应用。However, VLMs often struggle in these domains due to the scarcity of diverse text-rich vision-language data.
然而，由于缺乏多样化的富含文本的视觉-语言数据，VLMs 在这些领域中常常表现不佳。To address this challenge, we present CoSyn, a framework that leverages the coding capabilities of text-only large language models (LLMs) to automatically create synthetic text-rich multimodal data.
为了解决这一挑战，我们提出了 CoSyn，一个利用纯文本大语言模型（LLMs）的编码能力来自动生成富含文本的多模态合成数据的框架。Given input text describing a target domain (e.g., "nutrition fact labels"), CoSyn prompts an LLM to generate code (Python, HTML, LaTeX, etc.) for rendering synthetic images.
给定描述目标领域的输入文本（例如"营养标签"），CoSyn 会提示 LLM 生成用于渲染合成图像的代码（Python、HTML、LaTeX 等）。With the underlying code as textual representations of the synthetic images, CoSyn can generate high-quality instruction-tuning data, again relying on a text-only LLM.
通过将底层代码作为合成图像的文本表示，CoSyn 可以生成高质量的指令调优数据，再次依赖于纯文本 LLM。Using CoSyn, we constructed a dataset comprising 400K images and 2.7M rows of vision-language instruction-tuning data.
使用 CoSyn，我们构建了一个包含 40 万张图像和 270 万行视觉-语言指令调优数据的数据集。Comprehensive experiments on seven benchmarks demonstrate that models trained on our synthetic data achieve state-of-the-art performance among competitive open-source models, including Llama 3.2, and surpass proprietary models such as GPT-4V and Gemini 1.5 Flash.
在七个基准测试上的综合实验表明，使用我们的合成数据训练的模型在竞争性开源模型（包括 Llama 3.2）中实现了最先进的性能，并超越了 GPT-4V 和 Gemini 1.5 Flash 等专有模型。Furthermore, CoSyn can produce synthetic pointing data, enabling VLMs to ground information within input images, showcasing its potential for developing multimodal agents capable of acting in real-world environments.
此外，CoSyn 可以生成合成的指向数据，使 VLMs 能够在输入图像中定位信息，展示了其在开发能够在现实环境中行动的多模态代理方面的潜力。

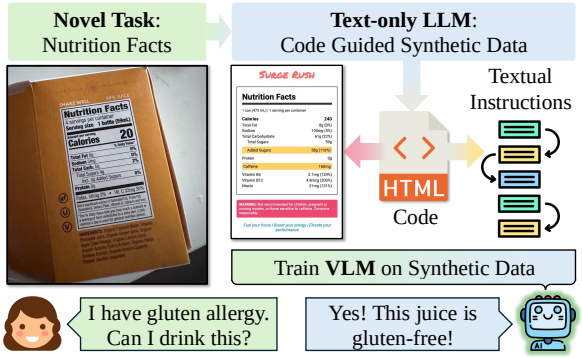


Figure 1: Given a novel task (e.g., answering questions about nutrition facts), our code-guided generation system can produce targeted synthetic data to enhance the performance of VLMs on that specific task.

1 Introduction

Instruction-tuned vision-language models (VLMs) have shown strong performance across a range of multimodal tasks (Radford et al., 2021; OpenAI, 2023; Liu et al., 2023).
指令调优的视觉-语言模型（VLMs）在一系列多模态任务中表现出色 (Radford et al., 2021;

OpenAI, 2023; Liu et al., 2023)。

However, these tasks typically focus on general image understanding over natural images rather than the specialized reasoning required for text-rich images such as charts, documents, diagrams, signs, labels, and screenshots.
然而，这些任务通常侧重于自然图像的一般理解，而不是对富含文本的图像（如图表、文档、图表、标志、标签和截图）所需的专门推理。

Understanding and reasoning over text-rich images is crucial for many applications, including analyzing scientific literature and figures (Asai et al., 2024), improving accessibility for users with visual impairments (Gurari et al., 2018), and enabling agentic workflows in real-world environments (Xie et al., 2024).
理解和推理富含文本的图像对于许多应用至关重要，包括分析科学文献和图表 (Asai et al., 2024)、改善视觉障碍用户的可访问性 (Gurari et al., 2018)，以及在现实环境中实现代理工作流程 (Xie et al., 2024)。

Effectively interpreting these structured visual formats requires both textual comprehension and spatial reasoning, which current models struggle with due to the limited availability of high-quality, realistic, and diverse vision-language datasets (Methani et al., 2020).
有效解释这些结构化视觉格式需要文本理解和空间推理，而当前模型由于高质量、真实且多样化的视觉-语言数据集的有限可用性而难以应对 (Methani et al., 2020)。

To address these challenges and inspired by the fact that text-rich images are typically rendered from code, we develop Code Guided Synthetic data generation system (CoSyn), a flexible framework for generating diverse synthetic text-rich multimodal data for vision-language instruction tuning.
为了解决这些挑战，并受到富含文本的图像通常由代码渲染的事实启发，我们开发了 Code Guided Synthetic 数据生成系统（CoSyn），这是一个灵活的框架，用于生成多样化的合成富含文本的多模态数据，用于视觉-语言指令调优。

As illustrated in Figure 2, CoSyn can generate multimodal data for various target domains from a short natural language query, such as book covers.
如图 2 所示，CoSyn 可以从简短的自然语言查询（如 书籍封面）生成各种目标领域的多模态数据。

CoSyn leverages text-only LLMs, which excel at code generation, to produce both data and code that render diverse text-rich images using 11 supported rendering tools (e.g., Python, HTML, LaTeX).
CoSyn 利用擅长代码生成的纯文本 LLMs，生成数据和代码，使用 11 种支持的渲染工具（如 Python、HTML、LaTeX）渲染多样化的富含文本的图像。

Grounded in the underlying code representation of the images, textual instructions are also generated by the text-only LLM to create vision-language instruction-tuning datasets.
基于图像的底层代码表示，纯文本 LLM 还生成文本指令，以创建视觉-语言指令调优数据集。

Using this framework, we construct the CoSyn-400K, as shown in Figure 3, a large-scale and diverse synthetic vision-language instruction-tuning dataset tailored for text-rich image understanding.
使用此框架，我们构建了 CoSyn-400K，如图 3 所示，这是一个大规模且多样化的合成视觉-语言指令调优数据集，专为富含文本的图像理解而设计。

We comprehensively evaluate the effectiveness of training on CoSyn-generated synthetic data across seven text-rich VQA benchmarks.
我们在七个富含文本的 VQA 基准上全面评估了使用 CoSyn 生成的合成数据进行训练的有效性。

Our model achieves state-of-the-art performance among competitive open-source models and surpasses proprietary models such as GPT-4V and Gemini 1.5.
我们的模型在竞争性开源模型中实现了最先进的性能，并超越了 GPT-4V 和 Gemini 1.5 等专有模型。

Notably, training on CoSyn synthetic data enables sample-efficient learning, achieving stronger performance with less data.
值得注意的是，使用 CoSyn 合成数据进行训练可以实现样本高效学习，用更少的数据实现更强的性能。

In addition, CoSyn can synthesize chain-of-thought (CoT) reasoning data (Wei et al., 2022), improving performance on tasks requiring multi-hop reasoning.
此外，CoSyn 可以合成思维链（CoT）推理数据 (Wei et al., 2022)，提高需要多跳推理的任务的性能。

A fine-grained analysis of question types in ChartQA (Masry et al., 2022) reveals that

training on CoSyn-400K results in stronger generalization to human-written questions.
对 ChartQA (Masry et al., 2022) 中问题类型的细粒度分析表明，使用 CoSyn-400K 进行训练可以更好地泛化到人类编写的问题。

In contrast, models trained solely on existing academic datasets often overfit to biased training data, overperforming on templated or machine-generated questions but struggling with more realistic, human-asked queries.
相比之下，仅使用现有学术数据集训练的模型通常会对有偏的训练数据过拟合，在模板化或机器生成的问题上表现优异，但在更现实的人类提问上表现不佳。

We then identify a key limitation of open-source VLMs that they struggle to generalize to out-of-domain tasks they were not trained on.
然后，我们确定了开源 VLMs 的一个关键局限性，即它们难以泛化到未训练过的领域外任务。

As shown in Figure 1, we introduce NutritionQA, a novel benchmark for understanding photos of nutrition labels, with practical applications like aiding users with visual impairments.
如图 1 所示，我们引入了 NutritionQA，这是一个用于理解营养标签照片的新颖基准，具有实际应用，如帮助视觉障碍用户。

Open-source VLMs perform poorly on this novel task, even after training on millions of images.
开源 VLMs 在这一新颖任务上表现不佳，即使经过数百万张图像的训练。

However, by training on CoSyn-400K, our model adapts strongly to this novel domain in a zero-shot setting with significantly less training data.
然而，通过使用 CoSyn-400K 进行训练，我们的模型在零样本设置中显著适应了这一新颖领域，且训练数据显著减少。

Remarkably, by generating just 7K in-domain synthetic nutrition label examples using CoSyn for fine-tuning, our model surpasses most open VLMs trained on millions of images.
值得注意的是，通过使用 CoSyn 生成仅 7K 的领域内合成营养标签示例进行微调，我们的模型超越了大多数经过数百万张图像训练的开源 VLMs。

This highlights CoSyn' s data efficiency and ability to help VLMs adapt to new domains through targeted synthetic data generation.
这突显了 CoSyn 的数据效率及其通过有针对性的合成数据生成帮助 VLMs 适应新领域的能力。

Finally, beyond the standard VQA task, we use CoSyn to generate synthetic pointing training data, which is particularly useful in agentic tasks.
最后，除了标准的 VQA 任务外，我们还使用 CoSyn 生成合成 指向训练数据，这在代理任务中特别有用。

The pointing data enables VLMs to retrieve coordinates for specific elements in a screenshot given a query like "Point to the Checkout button" (Deitke et al., 2024).
指向数据使 VLMs 能够根据类似 "指向结账按钮"的查询检索截图中特定元素的坐标 (Deitke et al., 2024)。

Our model trained on synthetic pointing data achieves state-of-the-art performance on the ScreenSpot click prediction benchmark (Baechler et al., 2024).
使用合成指向数据训练的模型在 ScreenSpot 点击预测基准上实现了最先进的性能 (Baechler et al., 2024)。

Overall, our work demonstrates that synthetic data is a promising solution for advancing vision-language models in understanding text-rich images and unlocking their potential as multimodal digital assistants for real-world applications.
总体而言，我们的工作表明，合成数据是推进视觉-语言模型理解富含文本图像并解锁其作为多模态数字助手在现实应用中潜力的一个有前景的解决方案。

2 Related Work

Vision Language Models. Tsimpoukelli et al. (2021) first demonstrate that pre-trained, frozen language models can be extended to process visual inputs.
视觉语言模型。Tsimpoukelli et al. (2021) 首次展示了预训练的冻结语言模型可以扩展到处理视觉输入。

Previous works fuse vision and language modalities using different strategies, such as cross-attention mechanisms (Alayrac et al., 2022) and Q-Former (Li et al., 2023).
先前的工作使用不同的策略融合视觉和语言模态，例如交叉注意力机制 (Alayrac et al., 2022) 和 Q-Former (Li et al., 2023)。

More recent architectures have converged on using MLP layers to project visual features into the language space (Liu et al., 2023).
最近的架构趋向于使用 MLP 层将视觉特征投影到语言空间 (Liu et al., 2023)。

However, these architectures are often imbalanced, with the language backbone substantially larger than the visual encoder.
然而，这些架构通常不平衡，语言主干明显大于视觉编码器。

As a result, without high-quality image-text data, models may overly rely on language priors, leading to hallucinations in their responses (Bai et al., 2024).
因此，如果没有高质量的图像-文本数据，模型可能过度依赖语言先验，导致其响应中出现幻觉 (Bai et al., 2024)。

Our work addresses this issue by generating high-quality multimodal data for text-rich images.
我们的工作通过为富含文本的图像生成高质量的多模态数据来解决这个问题。

Text-rich Images Understanding. Chart understanding and text-rich image understanding continue to challenge state-of-the-art models as naturally occurring vision-language data that can support training for understanding text-rich images is still scarce (Kahou et al., 2017; Kafle et al., 2018; Xu et al., 2023; Mukhopadhyay et al., 2024).
富含文本的图像理解。图表理解和富含文本的图像理解继续挑战最先进的模型，因为能够支持训练理解富含文本的图像的自然视觉-语言数据仍然稀缺 (Kahou et al., 2017; Kafle et al., 2018; Xu et al., 2023; Mukhopadhyay et al., 2024)。

In addition to charts and plots, a number of datasets address other kinds of text-rich images such as documents, infographics, diagrams and figures, and screenshots (Siegel et al., 2016; Mathew et al., 2021, 2022; Baechler et al., 2024; Roberts et al., 2024) have been made available.
除了图表和绘图外，许多数据集还涉及其他类型的富含文本的图像，例如文档、信息图表、图表和截图 (Siegel et al., 2016; Mathew et al., 2021, 2022; Baechler et al., 2024; Roberts et al., 2024)。

Many of these benchmarks are limited in size and scope, diversity of visualization types, and question types, making them suitable for evaluation but not for training data that could lead to generalized performance.
许多这些基准测试在规模、范围、可视化类型和问题类型的多样性方面受到限制，使其适合评估但不适合作为可能导致泛化性能的训练数据。

Synthetic Data for VLM. Generating synthetic images with annotations grounded in known source representations has been widely used in domains with limited vision-language data (Johnson-Roberson et al., 2017; Johnson et al., 2017; Cascante-Bonilla et al., 2022; Zhang et al., 2024).
视觉语言模型的合成数据。生成基于已知源表示的带注释的合成图像已广泛应用于视觉-语言数据有限的领域 (Johnson-Roberson et al., 2017; Johnson et al., 2017; Cascante-Bonilla et al., 2022; Zhang et al., 2024)。

This approach has been applied to chart and plot VQA typically using a limited small set of chart types and by instantiating handcrafted question templates (Kahou et al., 2017; Kafle et al., 2018; Methani et al., 2020; Singh and Shekhar, 2020).
这种方法通常应用于图表和绘图 VQA，使用有限的少量图表类型并通过实例化手工制作的问题模板 (Kahou et al., 2017; Kafle et al., 2018; Methani et al., 2020; Singh and Shekhar, 2020)。

Following this, Li and Tajbakhsh (2023) and Carbune et al. (2024a) explore using text-only LLMs to generate annotations from tables or text descriptions associated with charts to train VLMs.
随后，Li and Tajbakhsh (2023) 和 Carbune et al. (2024a) 探索使用纯文本 LLMs 从与图表相关的表格或文本描述生成注释以训练 VLMs。

Other recent approaches, similar to our procedure, explore generating data and code to render synthetic charts (Han et al., 2023; Shinoda et al., 2024; Xia et al., 2024).
其他最近的方法，类似于我们的程序，探索生成数据和代码以渲染合成图表 (Han et al., 2023; Shinoda et al., 2024; Xia et al., 2024)。

Molmo (Deitke et al., 2024) releases a synthetic text-rich image dataset, PixMo-docs, but smaller in scale and diversity than ours.
Molmo (Deitke et al., 2024) 发布了一个合成富含文本的图像数据集 PixMo-docs，但在规模和多样性上比我们的数据集小。

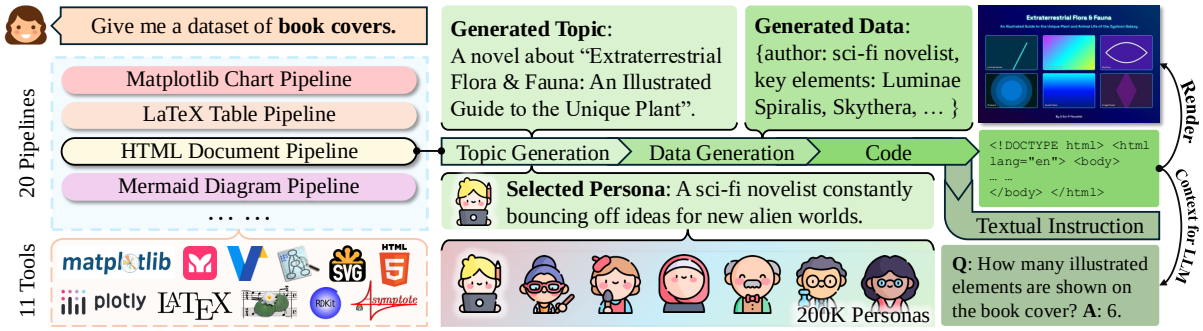


Figure 2: The overview of our Code Guided Synthetic data generation system (CoSyn), which has 20 generation pipelines based on 11 render tools. Given a user query, e.g., "book cover," CoSyn selects the appropriate pipelines and starts with generating diverse topics conditioned on personas, then synthesizes detailed data for code generation. The code renders the image and is also fed as context for an LLM to construct instruction-tuning data.

These works generate synthetic data that is still highly limited in terms of the diversity of topics, figure types, and rendering pipelines, which is important for generalizing to out-of-distribution tasks.
这些工作生成的合成数据在主题、图表类型和渲染管道的多样性方面仍然非常有限，这对于泛化到分布外任务非常重要。

In our work, we expand the scope beyond charts to encompass a wider range of diverse text-rich images.
在我们的工作中，我们将范围扩展到图表之外，以涵盖更广泛的多样化富含文本的图像。

3 Problem Formulation

Given a text query q about an image type, e.g., flow charts, our goal is to create a synthetic multimodal dataset $\mathcal{D}_q = \{(I, T)\}$, where I is the image, and T is the textual instruction-tuning data (e.g., question-answer pairs).
给定一个关于图像类型的文本查询 q，例如 流程图，我们的目标是创建一个合成的多模态数据集 $\mathcal{D}_q = \{(I, T)\}$，其中 I 是图像，T 是文本指令调优数据（例如，问答对）。

$\mathcal{D}_q$ is used to train a VLM to improve its ability to understand images related to q.
$\mathcal{D}_q$ 用于训练视觉语言模型（VLM），以提高其理解与 q 相关的图像的能力。

The core idea of our approach is using code C as the intermediate representation to bridge the image and text.
我们方法的核心思想是使用代码 C 作为中间表示，以连接图像和文本。

The overall generation process can be decomposed as follows:

$$P\left(I, T|q\right) = P_{\text{LM}}\left(C|q\right) \cdot P\left(I|C\right) \cdot P_{\text{LM}}\left(T|C\right)$$

where $P_{\text{LM}}\left(C|q\right)$ represents prompting a language model to generate code C, which is executed to render the image, $P\left(I|C\right)$.
整体生成过程可以分解如下：

$$P\left(I, T|q\right) = P_{\text{LM}}\left(C|q\right) \cdot P\left(I|C\right) \cdot P_{\text{LM}}\left(T|C\right)$$

其中 $P_{\text{LM}}\left(C|q\right)$ 表示提示语言模型生成代码 C，该代码被执行以渲染图像 $P\left(I|C\right)$。

$P_{\text{LM}}\left(T|C\right)$ uses code C (without the image) as context for an LLM to generate the textual instruction-tuning data.
$P_{\text{LM}}\left(T|C\right)$ 使用代码 C（不包含图像）作为上下文，供大型语言模型（LLM）生成文本指令调优数据。

4 CoSyn System

Figure 2 illustrates the workflow of our Code-Guided Synthetic data generation system (CoSyn).
图 2 展示了我们的代码引导合成数据生成系统（CoSyn）的工作流程。

The system takes a language input, such as "generate a dataset of book covers", and outputs a multimodal dataset.
该系统接收语言输入，例如 "生成一组书籍封面的数据集"，并输出一个多模态数据集。

Based on the input query, CoSyn selects one of 20 generation pipelines built on 11 rendering tools.
根据输入查询，CoSyn 从基于 11 种渲染工具构建的 20 个生成管道中选择一个。

The process starts with topic generation, conditioned on a sampled persona that guides the style and content.
该过程从主题生成开始，基于采样的角色来指导风格和内容。

Next, the system generates data content and converts it into code, which is then executed to render synthetic images.
接下来，系统生成数据内容并将其转换为代码，然后执行代码以渲染合成图像。

Finally, using the code as context, we prompt the LLM to generate corresponding textual instructions.
最后，使用代码作为上下文，我们提示大语言模型生成相应的文本指令。

In the following, we provide detailed explanations of the rendering tools supported by CoSyn, the tailored generation pipelines based on these tools, our persona-driven approach to diversify content and styles, and the large-scale dataset of 400K synthetic images generated by CoSyn.
在下文中，我们将详细解释 CoSyn 支持的渲染工具、基于这些工具的定制生成管道、我们通过角色驱动的方法来多样化内容和风格，以及由 CoSyn 生成的 40 万张合成图像的大规模数据集。

Rendering Tools. We integrate various rendering tools to generate diverse types of images, forming the foundation of CoSyn' s ability for text-rich image generation.

渲染工具。我们集成了多种渲染工具来生成不同类型的图像，这构成了 CoSyn 生成富含文本图像能力的基础。

For example, Matplotlib, Plotly, and Vega-Lite are used to create different types of charts.
例如，Matplotlib、Plotly 和 Vega-Lite 用于创建不同类型的图表。

LaTeX and HTML are used for documents and tables, while Mermaid and Graphviz generate diagrams.
LaTeX 和 HTML 用于生成文档和表格，而 Mermaid 和 Graphviz 用于生成图表。

We utilize SVG and Asymptote to create vector graphics and math-related content.
我们使用 SVG 和 Asymptote 来创建矢量图形和数学相关内容。

For specialized tasks, we rely on Lilypond to generate music sheets and RDKit for chemical structures.
对于特定任务，我们依赖 Lilypond 来生成乐谱，并依赖 RDKit 来生成化学结构。

We implement customized functions for each tool to execute LLM-generated code and obtain corresponding rendered images.
我们为每个工具实现了定制功能，以执行大语言模型生成的代码并获取相应的渲染图像。

These tools collectively enable CoSyn to produce a wide range of high-quality, text-rich synthetic images.
这些工具共同使 CoSyn 能够生成各种高质量、富含文本的合成图像。

Pipelines. We design 20 pipelines based on 11 rendering tools.[1]

管道。我们基于 11 种渲染工具设计了 20 个管道。[2]

Each pipeline follows the same procedure: (1) Topic generation to define the theme of this synthetic example, (2) Data generation to populate the detailed contents, (3) Code generation to create executable code that renders the image, and (4) Instruction generation conditioned on code to produce instructions, including questions, answers and explanations for chain-of-thought reasoning.
每个管道遵循相同的流程：(1) 主题生成以定义该合成示例的主题，(2) 数据生成以填充详细内容，(3) 代码生成以创建可执行代码来渲染图像，(4) 指令生成基于代码生成指令，包括问题、答案和用于链式推理的解释。

Each stage is controlled by a prompt customized for image category and rendering tool. Figure 8 shows all prompts of the HTML Document pipeline.
每个阶段由针对图像类别和渲染工具定制的提示控制。图 8 展示了 HTML 文档管道的所有提示。

Use personas to enhance diversity.

使用角色增强多样性。

LLMs often struggle to generate diverse synthetic data using sampling parameters alone (Yu et al., 2023), with biases leading to repetitive outputs across different runs.
大语言模型通常难以仅通过采样参数生成多样化的合成数据 (Yu et al., 2023)，偏差会导致不同运行中输出重复。

Recent work (Ge et al., 2024) shows that incorporating personas in prompts can improve diversity by enabling models to generate from

[1] Some tools are used in multiple pipelines, e.g., HTML is used for generating documents, tables, and charts.
[2] 一些工具在多个管道中使用，例如 HTML 用于生成文档、表格和图表。

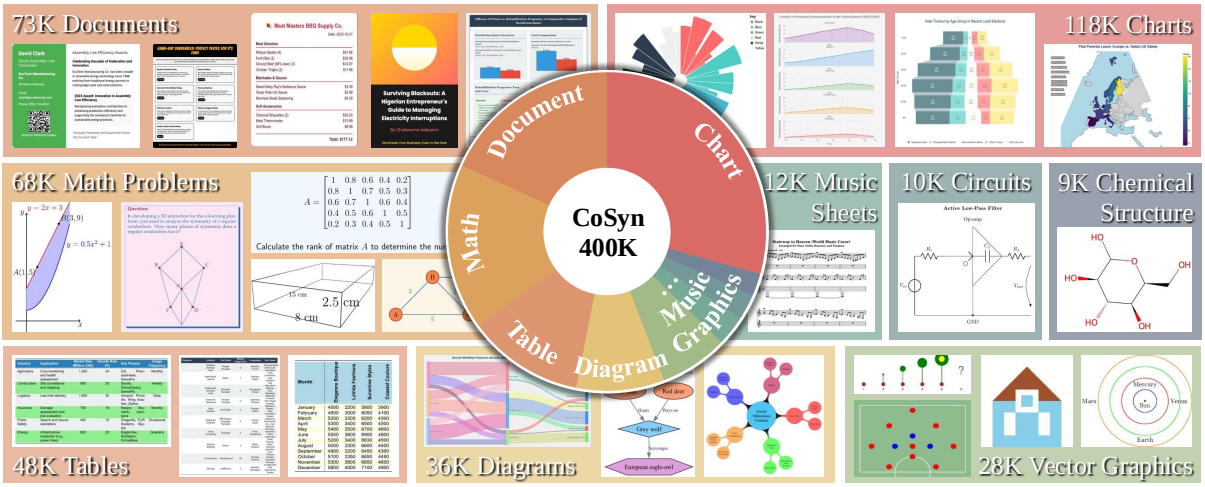


Figure 3: Our CoSyn-400K dataset consists of 9 categories of text-rich images with 2.7M instruction-tuning data. More qualitative examples, along with question-answer annotations, are available in Figure 12 -18 in Appendix C.

varied perspectives.
最近的工作 (Ge et al., 2024) 表明，在提示中加入角色可以通过使模型从不同角度生成内容来提高多样性。

CoSyn adopts personas to enhance diversity during the Topic Generation stage.
CoSyn 在主题生成阶段采用角色来增强多样性。

Each persona is a short sentence describing a personality or identity.
每个角色是一个描述个性或身份的短句。

For example, as shown in the middle of Figure 2, we sample a persona "a sci-fi novelist who likes alien worlds", which results in a topic of "a novel about Extraterrestrial Flora & Fauna" for generating the book cover image.
例如，如图 2 中间所示，我们采样了一个角色"喜欢外星世界的科幻小说家"，这导致生成书籍封面图像的主题为"关于外星植物和动物的小说"。

We use the 200K personas released by Ge et al. (2024).
我们使用了 Ge et al. (2024) 发布的 20 万个角色。

Implementation details.

实现细节。

CoSyn is built on the DataDreamer library (Patel et al., 2024), which supports robust multi-stage synthetic data generation pipelines that are easy to maintain, reproduce, and extend.
CoSyn 基于 DataDreamer 库 (Patel et al., 2024) 构建，该库支持健壮的多阶段合成数据生成管道，易于维护、复制和扩展。

DataDreamer documents the prompts and parameters used at each generation stage and implements several efficient techniques, such as parallel generation and response caching, to optimize performance.
DataDreamer 记录了每个生成阶段使用的提示和参数，并实现了并行生成和响应缓存等高效技术以优化性能。

For the data and code generation stages, we use Claude-3.5-Sonnet, which performs well in coding tasks (Anthropic, 2024b).
在数据和代码生成阶段，我们使用 Claude-3.5-Sonnet，它在编码任务中表现良好 (Anthropic, 2024b)。

For instruction-tuning data generation, we select GPT-4o-mini (OpenAI, 2023) for its cost efficiency.
对于指令调优数据生成，我们选择 GPT-4o-mini (OpenAI, 2023)，因为它具有成本效益。

CoSyn-400K. As shown in Figure 3, we use CoSyn to generate a large-scale synthetic dataset of 400K images across nine categories: charts, documents, math problems, tables, diagrams, vector graphics, music sheets, electrical circuits, and chemical structures.

CoSyn-400K。如图 3 所示，我们使用 CoSyn 生成了一个包含 40 万张图像的大规模合成数据集，涵盖九个类别：图表、文档、数学问题、表格、图表、矢量图形、乐谱、电路图和化学结构。

Since CoSyn is controlled via language inputs, it can easily generate diverse, fine-

grained image types by varying the input queries.
由于 CoSyn 通过语言输入控制，它可以通过改变输入查询轻松生成多样化、细粒度的图像类型。

For instance, we use over 100 queries to generate document data covering receipts, resumes, meal plans, etc.
例如，我们使用了 100 多个查询来生成涵盖收据、简历、饮食计划等的文档数据。

Some queries used for CoSyn-400K are provided in Appendix A.3.
用于 CoSyn-400K 的一些查询在附录 A.3 中提供。

This ensures that our dataset covers a broad range of domains.
这确保了我们的数据集涵盖了广泛的领域。

The following sections validate how our synthetic datasets enhance the ability of VLMs to understand text-rich images.
以下部分验证了我们的合成数据集如何增强视觉语言模型理解富含文本图像的能力。

5 Experimental Setup

Our experiments aim to verify the value of our synthetic data in the supervised fine-tuning stage of training vision-language models.
我们的实验旨在验证我们的合成数据在视觉-语言模型训练的监督微调阶段的价值。
This section introduces the architecture of our model, training strategy, datasets we used, baselines for comparison, and other details on implementation.
本节介绍了我们的模型架构、训练策略、使用的数据集、比较的基线以及实现的其他细节。

Model Architecture. We follow the same image preprocessing and architecture as Molmo (Deitke et al., 2024), which uses the MLP layer to connect the vision encoder and a pretrained LLM. We choose OpenAI' s CLIP (ViT-L/14 336px) (Radford et al., 2021) as the vision backbone and Mistral-7B (Jiang et al., 2023) as the language model.
模型架构。我们遵循与 Molmo (Deitke et al., 2024) 相同的图像预处理和架构，使用 MLP 层连接视觉编码器和预训练的大型语言模型 (LLM) 。我们选择 OpenAI 的 CLIP (ViT-L/14 336px) (Radford et al., 2021) 作为视觉骨干，Mistral-7B (Jiang et al., 2023) 作为语言模型。

Training Process. We adopt the same training strategy as Molmo (Deitke et al., 2024), which consists of two stages: (1) Pre-training on dense captions from PixMo-Cap and (2) Supervised fine-tuning on three categories of datasets below:
训练过程。我们采用与 Molmo (Deitke et al., 2024) 相同的训练策略，包括两个阶段：(1) 在 PixMo-Cap 的密集标注上进行 预训练，以及 (2) 在以下三类数据集上进行 监督微调：

- Evaluation Datasets. We evaluate our model on seven text-rich benchmarks, including ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), InfographicVQA (Mathew et al., 2022), TableVQA-Bench (Kim et al., 2024), AI2 Diagrams (Kembhavi et al., 2016), TextVQA (Singh et al., 2019), and ScreenQA (Baechler et al., 2024). We adopt their official metrics for calculating performance. In total, we have 138K training images from the evaluation datasets.[3]
  评估数据集。我们在七个富含文本的基准上评估我们的模型，包括 ChartQA (Masry et al., 2022)、DocVQA (Mathew et al., 2021)、InfographicVQA (Mathew et al., 2022)、TableVQA-Bench (Kim et al., 2024)、AI2 Diagrams (Kembhavi et al., 2016)、TextVQA (Singh et al., 2019) 和 ScreenQA (Baechler et al., 2024)。我们采用它们的官方指标来计算性能。评估数据集共有 138K 张训练图像。[4]
- Auxiliary Datasets. We select additional academic datasets for fine-tuning: VQAv2 (Goyal et al., 2017), GQA (Hudson and Manning, 2019), OK-VQA (Marino et al., 2019), OCR-VQA (Mishra et al., 2019), A-OKVQA (Schwenk et al., 2022), ScienceQA (Lu et al., 2022), TabMWP (Lu et al., 2023), ST-VQA (Biten et al., 2019), TallyQA (Acharya et al., 2019), DVQA (Kafle et al., 2018), FigureQA (Kahou et al., 2017), and PlotQA (Methani et al., 2020). The auxiliary datasets contain around 1M training images.
  辅助数据集。我们选择了额外的学术数据集进行微调：VQAv2 (Goyal et al., 2017)、GQA (Hudson and Manning, 2019)、

[3]TableVQA is an eval-only benchmark (no training split), and we do not use the training split from ScreenQA.
[4]TableVQA 是一个仅用于评估的基准（没有训练集），我们也没有使用 ScreenQA 的训练集。

Model	ChartQA	DocVQA	InfoVQA	TableVQA	AI2D	TextVQA	ScreenQA	Average
GPT-4V	78.1	87.2	75.1	60.5	89.4	78.0	41.6	72.8
Gemini 1.5 Flash	85.4	89.9	75.3	72.6	91.7	78.7	40.1	76.2
Claude-3 Opus	80.8	89.3	55.6	70.0	88.1	67.5	39.8	70.2
PaliGemma-3B†	71.4	84.8	47.8	46.4	73.3	76.5	32.2	61.8
BLIP-3-4B†	60.0	61.4	31.5	24.3	74.2	71.0	26.2	49.8
Cambrian-7B†	73.3	77.8	41.6	40.6	73.0	71.7	44.4	64.2
LLaVA-1.5-7B†*	17.8	28.1	25.8	33.1	55.5	58.2	17.6	33.7
LLaVA-Next-8B†	69.5	78.2	43.8	43.9	71.6	65.3	34.2	58.1
LLaVA-OneVision-7B†	80.0	87.5	68.8	64.6	81.4	78.3	46.3	72.4
Pixtral-12B	81.8	90.7	50.8	67.0	79.0	75.7	39.4	69.2
Llama 3.2 11B	83.4	88.4	63.6	51.1	91.9	73.1	87.7	77.0
Ours (7B)†	86.3	90.0	70.5	65.8	91.9	82.0	80.1	80.9
Ours (7B-zero-shot)†*	80.8	82.9	59.8	64.9	83.9	72.7	78.1	74.7

Table 1: Results on 7 text-rich benchmarks. The result of the best-performing open-source model is bold, and the second-best is underlined. Models with † stand for open data and code for multimodal training. Models with * are zero-shot models, which means the models are not trained on instances from any of the evaluation datasets.

OK-VQA (Marino et al., 2019)、OCR-VQA (Mishra et al., 2019)、A-OKVQA (Schwenk et al., 2022)、ScienceQA (Lu et al., 2022)、TabMWP (Lu et al., 2023)、ST-VQA (Biten et al., 2019)、TallyQA (Acharya et al., 2019)、DVQA (Kafle et al., 2018)、FigureQA (Kahou et al., 2017) 和 PlotQA (Methani et al., 2020)。辅助数据集包含约 1M 张训练图像。

- Synthetic Datasets. As introduced in Sec 4 and also shown in Figure 3, our synthetic datasets include 400K text-rich images from 9 categories.
  合成数据集。如第 4 节所述，并在图 3 中展示，我们的合成数据集包含来自 9 个类别的 400K 张富含文本的图像。

Our best-performing model uses all three categories of datasets above. We also trained a zero-shot model using only auxiliary and synthetic data without any examples from the evaluation datasets, which still exhibits competitive benchmark performance, as shown in the last row of Table 1.
我们表现最好的模型使用了上述所有三类数据集。我们还训练了一个仅使用辅助和合成数据的零样本模型，没有使用任何评估数据集的样本，该模型仍然表现出具有竞争力的基准性能，如表 1 的最后一行所示。

Baselines. We compare recent open-source VLMs with a similar scale (7B), including PaliGemma-3B (Beyer et al., 2024), BLIP-3-4B (Xue et al., 2024), Cambrian-7B (Tong et al., 2024), LLaVA-1.5-7B (Liu et al., 2023), LLaVA-Next-8B (Liu et al., 2024), LLaVA OneVision-7B (Li et al., 2024), Pixtral-12B (Agrawal et al., 2024), Llama 3.2 V (Meta, 2024). We also include proprietary models: GPT-4V (OpenAI, 2023), Gemini-1.5-Flash (Team, 2024), and Claude-3 Opus (Anthropic, 2024a).
基线。我们比较了最近开源的规模相近（7B）的视觉-语言模型（VLMs），包括 PaliGemma-3B (Beyer et al., 2024)、BLIP-3-4B (Xue et al., 2024)、Cambrian-7B (Tong et al., 2024)、LLaVA-1.5-7B (Liu et al., 2023)、LLaVA-Next-8B (Liu et al., 2024)、LLaVA OneVision-7B (Li et al., 2024)、Pixtral-12B (Agrawal et al., 2024)、Llama 3.2 V (Meta, 2024)。我们还包括了专有模型：GPT-4V (OpenAI, 2023)、Gemini-1.5-Flash (Team, 2024) 和 Claude-3 Opus (Anthropic, 2024a)。

Implementation Details. We train our model on TPU v3-128 with a batch size of 32. Our best-performing model is trained for 60K steps, taking about 30 hours. The checkpoints with the highest validation performance are retained for testing.
实现细节。我们在 TPU v3-128 上训练我们的模型，批量大小为 32。我们表现最好的模型训练了 60K 步，耗时约 30 小时。验证性能最高的检查点被保留用于测试。

6 Results

This section covers (1) the competitive performance of the model trained on our synthetic data (Sec 6.1), (2) the comprehensive analyses to highlight the benefits of synthetic data (Sec 6.2), and (3) the effectiveness of syn-

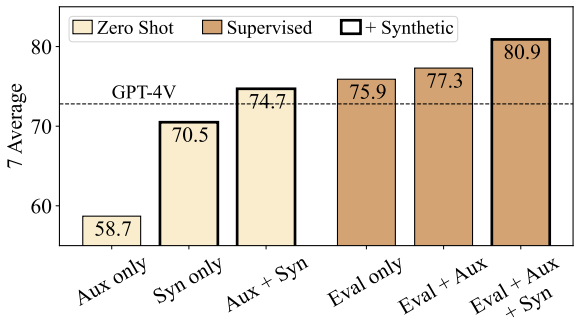


Figure 4: Ablation on training data selection. Aux, Syn, and Eval stand for auxiliary, synthetic, and evaluation datasets, respectively. We report the average score on eight benchmarks. The detailed performance breakdown on each benchmark is in Table 7.

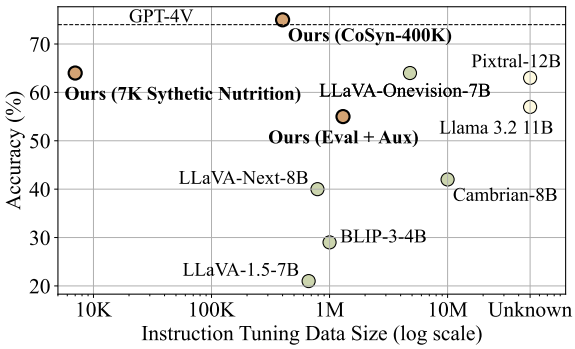


Figure 5: Zero shot performance on NutritionQA. The x-axis denotes the number of training examples used for the instruction-tuning stage. The models on the upper left side demonstrate better data efficiency.

thetic pointing data in improving VLMs for web agent tasks (Sec 6.3).
本节涵盖：(1) 在我们合成数据上训练的模型的竞争性能（第 6.1 节），(2) 强调合成数据优势的综合分析（第 6.2 节），以及 (3) 合成指向数据在改进 VLMs 用于网络代理任务中的有效性（第 6.3 节）。

6.1 Main Results

Table 1 compares our model' s performance with both open and closed models across seven text-rich benchmarks. On average, our 7B model achieves the highest performance, surpassing the second-best model (Llama 3.2 11B) by 3.9%. Notably, our model ranks first in four out of the seven datasets and second in the remaining three. More surprisingly, our zero-shot model (the last row in Table 1) outperforms most open and closed models without exposure to any training instances from the evaluation datasets. In contrast, these competing models often rely on benchmark training data and are thus not true zero-shot models. This result demonstrates that the capabilities learned from our synthetic data can transfer effectively to downstream tasks.
表 1 比较了我们的模型在七个文本丰富的基准测试中与开源和闭源模型的性能。平均而言，我们的 7B 模型表现最佳，超过了第二好的模型（Llama 3.2 11B）3.9

6.2 Analysis

In the following experiments, we quantify the contribution of synthetic data to the benchmark performance by ablating the combinations of fine-tuning datasets. Then, we demonstrate that our CoSyn system can efficiently assist VLMs in generalizing to novel tasks. Finally, we show that synthetic data can help mitigate the overfitting of biases.
在以下实验中，我们通过消融微调数据集的组合来量化合成数据对基准性能的贡献。然后，我们展示了我们的 CoSyn 系统可以有效地帮助 VLMs 泛化到新任务。最后，我们展示了合成数据可以帮助减轻偏差的过拟合。

Synthetic data boosts the performance. Table 4 presents an ablation study on the choices of supervised fine-tuning data. In the zero-shot settings, when the model is trained on auxiliary datasets (over 1M training images not directly from the evaluation tasks), it fails to generalize effectively to the evaluation tasks, with a substantial performance gap of 14.1% below GPT-4V. However, using only 400K synthetic samples achieves a performance comparable to GPT-4V. Our best zero-shot model surpasses GPT-4V when jointly training synthetic and auxiliary data. Under the supervised settings, training with in-domain data alone yields strong performance. However, adding 1M auxiliary samples provides a modest improvement of 1.4%, while incorporating synthetic data results in a more significant 3.6% boost. These results demonstrate the effectiveness of synthetic data in enhancing VLMs' performance on text-rich tasks.
合成数据提升性能。表 4 展示了关于监督微调数据选择的消融研究。在零样本设置中，当模型在辅助数据集（超过 100 万张训练图像，不直接来自评估任务）上训练时，它无法有效地泛化到评估任务，性能差距高达 14.1

Zero-shot Generalization on a Novel Task.

Vision-language models typically rely on in-domain data to perform well on specific tasks. When encountering a novel task, such as answering questions about nutrition labels in Figure 1, models without seeing similar examples during training may struggle with this novel task. However, our CoSyn system enables controllable data generation. Given the task name as input, CoSyn can generate task-specific data to fine-tune the model.

To validate this, we annotated a small evaluation dataset called NutritionQA, which includes 100 examples of questions about photos of nutrition labels. Some questions require multi-hop reasoning, as Figure 10 illustrates. We evaluated GPT-4V and several open-source VLMs on this dataset and report the performance in Figure 5. The x-axis in Figure 5 represents the amount of data used during the instruction fine-tuning stage.

Despite being trained on millions of images, we observe that open-source VLMs are not data-efficient and perform poorly on this novel task compared to GPT-4V. Although many open-source VLMs claim to achieve GPT-4V-level performance, they fall short when tested on new tasks in the wild. Without synthetic data, our model (Eval + Aux) achieves results similar to those of open models. However, when trained on 400K synthetic samples, our model matches GPT-4V' s performance.

More impressively, we used CoSyn to generate 7K synthetic nutrition label samples and fine-tuned the model using only this 7K data. The resulting model outperforms most open-source VLMs on the NutritionQA task. These results demonstrate that code-guided synthetic data is an effective and efficient method for adapting VLMs to new domains.

在新任务上的零样本泛化。视觉语言模型通常依赖于领域内数据以在特定任务上表现良好。当遇到新任务时，例如回答关于营养标签的问题（如图 1 所示），在训练期间没有看到类似示例的模型可能会在这一新任务上表现不佳。然而，我们的 CoSyn 系统支持可控的数据生成。给定任务名称作为输入，CoSyn 可以生成任务特定的数据来微调模型。

为了验证这一点，我们标注了一个名为 NutritionQA 的小型评估数据集，其中包含 100 个关于营养标签照片的问题示例。一些问题需要多跳推理，如图 10 所示。我们评估了 GPT-4V 和几个开源 VLMs 在该数据集上的表现，并在图 5 中报告了性能。图 5 中的 x 轴表示在指令微调阶段使用的数据量。

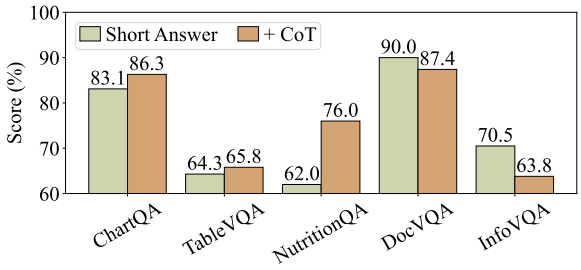


Figure 6: Ablation of using Chain-of-Thought reasoning. Short Answer represents prompting model to output the answer as short as possible. + CoT stands for providing Chain-of-Thought reasoning before giving the final answer. Results on all datasets are in Table 6.

尽管在数百万张图像上进行了训练，我们观察到开源 VLMs 在数据效率上表现不佳，并且在这一新任务上表现不如 GPT-4V。尽管许多开源 VLMs 声称达到了 GPT-4V 级别的性能，但在实际测试新任务时却表现不佳。在没有合成数据的情况下，我们的模型（Eval + Aux）取得了与开源模型相似的结果。然而，当在 40 万合成样本上训练时，我们的模型与 GPT-4V 的性能相当。

更令人印象深刻的是，我们使用 CoSyn 生成了 7000 个合成营养标签样本，并仅使用这 7000 个数据微调了模型。生成的模型在 NutritionQA 任务上优于大多数开源 VLMs。这些结果表明，代码引导的合成数据是一种有效且高效的方法，可以使 VLMs 适应新领域。

Synthetic Data for Chain-of-Thought Reasoning. Existing text-rich datasets, such as ChartQA (Masry et al., 2022), are typically annotated with short answers. However, questions like "Compute the mean of the data in the plot" require step-by-step mathematical reasoning to arrive at the correct answer. Models trained only with short-answer supervision may fail to learn proper plot comprehension, but instead overfitting to annotation biases in these datasets. On the contrary, our CoSyn-400K includes explanation text alongside the short answer. Each instruction-tuning example consists of a (question, explanation, short answer) triplet, enabling models to learn chain-of-thought (CoT) reasoning. During fine-tuning, we design two prompt templates for our synthetic data:

用于链式思维推理的合成数据。现有的文本丰富数据集，例如 ChartQA (Masry et al., 2022)，

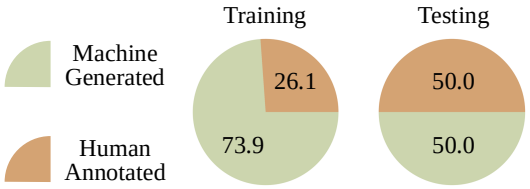
通常标注有简短答案。然而，像“计算图中数据的平均值”这样的问题需要逐步的数学推理才能得出正确答案。仅通过简短答案监督训练的模型可能无法学习正确的图表理解，而是过度拟合这些数据集中的标注偏差。相反，我们的 CoSyn-400K 包含了与简短答案一起的解释文本。每个指令微调示例由 (问题, 解释, 简短答案) 三元组组成，使模型能够学习链式思维 (CoT) 推理。在微调期间，我们为合成数据设计了两种提示模板：

> CoT Prompt: <Question> Provide reasoning steps and then give the short answer. <Explanation> Answer: <Answer>

> Short Answer Prompt: <Question> Answer with as few words as possible. <Answer>

Those prompts allow VLMs to switch between the two answering styles and perform CoT reasoning when necessary. Figure 6 shows that incorporating CoT reasoning improves performance on ChartQA, TableVQA, and NutritionQA, as these datasets contain examples requiring multi-hop reasoning. However, we observe that adding CoT reasoning reduces performance on DocVQA and InfoVQA. We find this decline is caused by answer biases in these benchmarks. Specifically, the ground-truth answers favor short responses, often penalizing more detailed and verbal responses. For instance, in DocVQA, the ground-truth for an example is “T-Th”, whereas the model responds with “Tuesday to Thursday”. Although the response is correct, the strict string-matching metric assigns it a zero score. This highlights key limitations of current multimodal benchmarks, including answering biases and rigid evaluation metrics that fail to capture the full extent of a model’s capabilities.

这些提示允许 VLMs 在两种回答风格之间切换，并在必要时执行链式思维推理。图 6 显示，加入链式思维推理可以提高 ChartQA、TableVQA 和 NutritionQA 的性能，因为这些数据集包含需要多跳推理的示例。然而，我们观察到，在 DocVQA 和 InfoVQA 上加入链式思维推理会降低性能。我们发现这种下降是由这些基准测试中的答案偏差引起的。具体来说，真实答案倾向于简短回答，通常会惩罚更详细和口头化的回答。例如，在 DocVQA 中，一个示例的真实答案是 “T-Th”，而模型回答为 “Tuesday to Thursday”。尽管回答是正确的，但严格的字符串匹配指标将其评分为零。这突显了当前多模态基准测试的关键局限性，包括回答偏差和僵化的评估指标，无法全面捕捉模型的能力。



ChartQA	Average	Machine	Human	Δ ↓
PaliGemma-3B	71.4	88.5	54.2	34.3
ChartPali-5B	77.3	93.7	60.9	32.8
Ours (w/o Syn)	81.4	92.2	70.4	21.8
Ours (w/ Syn)	86.3	93.4	79.1	14.2

Table 2: Results on human and machine-generated questions of ChartQA. The pie charts above display the percentage distribution of two question types in training and testing. Δ (↓ lower is better) denotes the performance gap between human and machine questions.

Synthetic Data for Mitigating Biases. Our previous experiments reveal answering biases in multimodal benchmarks, which VLMs trained solely on these datasets often inherit. To further validate this issue, we analyze ChartQA and observe a distribution shift in question types. As shown in the pie charts above Table 2, some ChartQA questions are human-annotated, while others are generated by the language model T5 (Raffel et al., 2020), which is heavily influenced by prompt phrasing and limited to a fixed set of question templates. During training, most questions (73.9%) in ChartQA are machine-generated, while the test set contains an even distribution of human-annotated and machine-generated questions. Models trained exclusively on ChartQA tend to overfit to T5-generated questions. Table 2 illustrates this issue: PaliGemma (Beyer et al., 2024) and ChartPali (Carbune et al., 2024b) achieve high accuracy on machine-generated questions but experience a significant performance drop of over 30% on human-annotated questions.

Similarly, without synthetic data, our model shows a noticeable 21.8% gap between the two question types. However, incorporating synthetic data during training reduces this gap to 14.2%, improving the model’s ability to

answer human-asked questions. This suggests that synthetic data can mitigate overfitting on benchmarks and enhance VLMs' usability in real-world applications.

用于减轻偏差的合成数据。我们之前的实验揭示了多模态基准测试中的回答偏差，仅在这些数据集上训练的 VLMs 通常会继承这些偏差。为了进一步验证这一问题，我们分析了 ChartQA 并观察到问题类型的分布变化。如表 2 上方的饼图所示，一些 ChartQA 问题是由人工标注的，而另一些是由语言模型 T5 (Raffel et al., 2020) 生成的，后者受提示措辞的严重影响，并且仅限于一组固定的问题模板。在训练期间，ChartQA 中的大多数问题（73.9%）是机器生成的，而测试集包含均匀分布的人工标注和机器生成的问题。仅在 ChartQA 上训练的模型往往会过度拟合 T5 生成的问题。表 2 说明了这一问题：PaliGemma (Beyer et al., 2024) 和 ChartPali (Carbune et al., 2024b) 在机器生成的问题上取得了高准确率，但在人工标注的问题上表现显著下降了超过 30%。

类似地，在没有合成数据的情况下，我们的模型在两种问题类型之间显示出明显的 21.8% 差距。然而，在训练中加入合成数据将这一差距缩小到 14.2%，提高了模型回答人工提问的能力。这表明合成数据可以减轻基准测试中的过拟合，并增强 VLMs 在实际应用中的可用性。

6.3 Synthetic Pointing Data

Pointing enables vision-language models to answer questions by providing specific points on images. This functionality allows models to ground their responses in visual content and interact with environments, which is crucial for developing digital agents. We find that we can synthesize pointing data using our code-guided generation system.

合成指向数据。指向功能使视觉语言模型能够通过在图像上提供特定点来回答问题。这一功能使模型能够将其响应基于视觉内容并与环境交互，这对于开发数字代理至关重要。我们发现可以使用我们的代码引导生成系统合成指向数据。

Method. Since we have access to the source code for all generated images, we can prompt an LLM to modify the code to draw points on the images explicitly. As illustrated in Figure 7, we feed the image' s source code as context to the LLM, which generates a pointing question and edits the code to draw points with a predefined color. By extracting the pixel values of these points, we can obtain their exact (x, y) coordinates.[5] We then use this pointing data to train VLMs, enabling them to answer questions by providing point coordinates. In total, we generate pointing data for 65K synthetic images. Figure 19 shows some qualitative examples from our synthetic pointing dataset.

方法。由于我们可以访问所有生成图像的源代码，我们可以提示 LLM 修改代码以在图像上显式绘制点。如图 7 所示，我们将图像的源代码作为上下文输入 LLM，LLM 生成指向问题并编辑代码以使用预定义颜色绘制点。通过提取这些点的像素值，我们可以获得其精确的 (x, y) 坐标。然后，我们使用这些指向数据来训练 VLMs，使其能够通过提供点坐标来回答问题。我们总共为 65K 合成图像生成了指向数据。图 19 展示了我们合成指向数据集中的一些定性示例。

Setup. We evaluate pointing ability on ScreenSpot (Cheng et al., 2024), where the task requires models to provide the correct click location based on a given instruction. ScreenSpot contains screenshots from mobile phones, desktops, and web pages. To assess the effectiveness of our synthetic pointing data, we compare it to the model trained on PixMo-point (Deitke et al., 2024), which consists of 155K human-annotated images. Our best-performing model uses both PixMo-point and synthetic pointing data. Additionally, we compare against existing methods like CogAgent (Hong et al., 2024), SeeClick (Cheng et al., 2024), and UGround (Gou et al., 2024), which is trained on 1.3M screenshots.

设置。我们在 ScreenSpot (Cheng et al., 2024) 上评估指向能力，该任务要求模型根据给定指令提供正确的点击位置。ScreenSpot 包含来自手机、桌面和网页的截图。为了评估我们合成指向数据的有效性，我们将其与在 PixMo-point (Deitke et al., 2024) 上训练的模型进行比较，后者由 155K 人工标注的图像组成。我们表现最好的模型同时使用了 PixMo-point 和合成指向数据。此外，我们还与现有方法如 CogAgent (Hong et al., 2024)、SeeClick (Cheng et al., 2024) 和 UGround (Gou et al., 2024) 进行了比较，后者在 130 万张截图上进行了训练。

[5] The coordinates of points are normalized to (0, 100) to mitigate the influence of image resolution.
点的坐标被归一化为 (0, 100) 以减轻图像分辨率的影响。

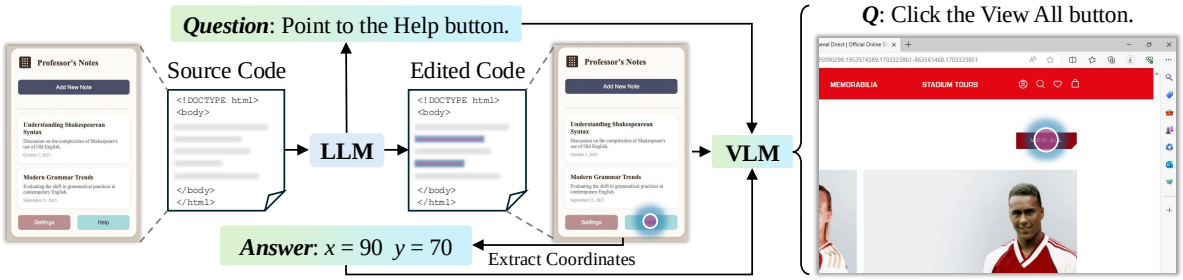


Figure 7: The overview of enabling VLMs to point through synthetic data. (a) We synthesize pointing data by prompting an LLM to generate pointing questions and edit the code to draw the answer points explicitly. (b) We demonstrate that the VLM trained on synthetic pointing data can be generalized to real agentic tasks.

Results. Table 3 compares the click accuracy of our models with previous methods. Using 65K synthetic pointing samples, our model achieves performance comparable to the one trained on 155K human-annotated samples. When combining synthetic and human data, our model achieves state-of-the-art performance on ScreenSpot, surpassing the recent UGround (Gou et al., 2024), which was trained on 1.3M screenshots. These results demonstrate that synthetic pointing data is a data-efficient approach for improving VLM performance on agentic tasks involving click prediction.

结果。表 3 比较了我们的模型与先前方法的点击准确率。使用 65K 合成指向样本，我们的模型取得了与在 155K 人工标注样本上训练的模型相当的性能。当结合合成和人工数据时，我们的模型在 ScreenSpot 上取得了最先进的性能，超过了最近在 130 万张截图上训练的 UGround (Gou et al., 2024)。这些结果表明，合成指向数据是一种数据高效的方法，可以提高 VLMs 在涉及点击预测的代理任务中的性能。

Model	Mobile		Desktop		Web		Avg
	Text	Icon	Text	Icon	Text	Icon	
GPT-4o	20.2	24.9	21.1	23.6	12.2	7.8	18.3
CogAgent	67.0	24.0	74.2	20.0	70.4	28.6	47.4
SeeClick	78.0	52.0	72.2	30.0	55.7	32.5	53.4
UGround	82.8	60.3	82.5	63.6	80.4	70.4	73.3
Synthetic	90.8	53.3	78.4	58.6	80.0	47.1	68.0
Human	84.2	59.0	88.1	52.9	76.5	50.5	68.5
Combined	89.0	65.1	87.6	65.7	83.0	58.7	74.9

Table 3: Click accuracy on ScreenSpot. We report our models trained on different pointing data. Human stands for using the human-annotated data from PixMo-point (Deitke et al., 2024). Combined means combining human-annotated data with our synthetic pointing data.

7 Conclusion

In this work, we introduced CoSyn, a framework for generating synthetic data that significantly enhances VLM performance on text-rich image understanding.

在这项工作中，我们提出了 CoSyn，一个生成合成数据的框架，显著提升了 VLM 在富含文本图像理解任务中的性能。Our comprehensive analysis highlights the advantages of synthetic data for domain generalization, data efficiency, and bias mitigation.

我们的综合分析强调了合成数据在领域泛化、数据效率和偏差缓解方面的优势。Our work demonstrates that the coding capabilities of text-only LLMs can effectively assist multimodal learning and unleash the potential of vision-language models for real-world applications.

我们的工作表明，纯文本 LLMs 的编码能力可以有效地辅助多模态学习，并释放视觉-语言模型在现实应用中的潜力。

Limitation

The effectiveness of synthetic data depends heavily on the quality and diversity of the prompts and rendering pipelines used for data generation.

合成数据的有效性在很大程度上取决于用于数据生成的提示和渲染管道的质量和多样性。For highly specialized or underrepresented domains, generating sufficiently diverse data remains challenging and may require careful prompt engineering or additional customization of rendering tools.

对于高度专业化或代表性不足的领域，生成足够多样化的数据仍然具有挑战性，可能需要仔细的提示工程或渲染工具的额外定制。Targeted synthetic data generation may be essential for certain tasks to achieve adequate performance, and ensuring relevance and coverage still requires domain-specific expertise.
针对特定任务的合成数据生成对于实现足够的性能可能是必要的，而确保相关性和覆盖范围仍然需要领域特定的专业知识。Synthetic data also may not fully capture the complexity of real-world data in some scenarios. Therefore, improving the diversity and realism of synthetic data to better support models in highly variable or evolving domains is a reasonable avenue for future research.
在某些情况下，合成数据可能无法完全捕捉现实世界数据的复杂性。因此，提高合成数据的多样性和真实性，以更好地支持在高度变化或不断发展的领域中的模型，是未来研究的一个合理方向。Finally, our current synthetic data is limited to English and may require further extension for multilingual support.
最后，我们当前的合成数据仅限于英语，可能需要进一步扩展以支持多语言。

Ethical Statement

To the best of our knowledge, this work presents no significant ethical concerns. We note, however, that the use of synthetic data can propagate biases present in the generation model used. Conversely, synthetic data can also help mitigate biases and expand coverage, as demonstrated in this work, by greatly expanding the domains present in vision-language instruction-tuning training data to yield stronger generalized performance.
据我们所知，这项工作没有提出重大的伦理问题。然而，我们注意到，使用合成数据可能会传播生成模型中存在的偏见。相反，合成数据也可以帮助缓解偏见并扩大覆盖范围，正如本工作所展示的那样，通过大大扩展视觉-语言指令调优训练数据中的领域，从而产生更强的泛化性能。

Acknowledgement

This work was done during Yue Yang's internship at the PRIOR team of Ai2. This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200005, and the Defense Advanced Research Projects Agency's (DARPA) SciFy program (Agreement No. HR00112520300), and gifts from the UPenn ASSET center and Ai2. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes, notwithstanding any copyright annotation therein.
这项工作是在 Yue Yang 在 Ai2 的 PRIOR 团队实习期间完成的。这项研究部分得到了国家情报总监办公室（ODNI）、高级情报研究计划活动（IARPA）通过 HIATUS 计划合同 #2022-22072200005 的支持，以及国防高级研究计划局（DARPA）的 SciFy 计划（协议号 HR00112520300）的支持，并得到了宾夕法尼亚大学 ASSET 中心和 Ai2 的捐赠。本文中包含的观点和结论是作者的观点和结论，不应被解释为必然代表 ODNI、IARPA、DARPA 或美国政府的官方政策，无论是明示的还是暗示的。美国政府被授权为政府目的复制和分发重印本，尽管其中包含任何版权注释。

References

Manoj Acharya, Kushal Kafle, and Christopher Kanan. 2019. TallyQA: Answering complex counting questions. In AAAI.

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Devendra Chaplot, Jessica Chudnovsky, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, et al. 2024. Pixtral 12b. arXiv preprint arXiv:2410.07073.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. In NeurIPS.

Anthropic. 2024a. The claude 3 model family: Opus, sonnet, haiku.

Anthropic. 2024b. Introducing the next generation of claude.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, et al. 2024. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. arXiv preprint arXiv:2411.14199.

Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. 2024. Screenai: A vision-language model for ui and infographics understanding. Preprint, arXiv:2402.04615.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. Hallucination of multimodal large language models: A survey. arXiv preprint arXiv:2404.18930.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. 2024. PaliGemma: A versatile 3B VLM for transfer. arXiv preprint arXiv:2407.07726.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. 2019. Scene text visual question answering. In ICCV.

Victor Carbune, Hassan Mansoor, Fangyu Liu, Rahul Aralikatte, Gilles Baechler, Jindong Chen, and Abhanshu Sharma. 2024a. Chart-based reasoning: Transferring capabilities from llms to vlms. arXiv preprint arXiv:2403.12596.

Victor Carbune, Hassan Mansoor, Fangyu Liu, Rahul Aralikatte, Gilles Baechler, Jindong Chen, and Abhanshu Sharma. 2024b. Chart-based reasoning: Transferring capabilities from llms to vlms. arXiv preprint arXiv:2403.12596.

Paola Cascante-Bonilla, Hui Wu, Letao Wang, Rogerio S Feris, and Vicente Ordonez. 2022. Simvqa: Exploring simulated environments for visual question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5056–5066.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. Preprint, arXiv:2401.10935.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. 2024. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. arXiv preprint arXiv:2409.17146.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. Preprint, arXiv:2406.20094.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2024. Navigating the digital world as humans do: Universal visual grounding for gui agents. arXiv preprint arXiv:2410.05243.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In Conference on Computer Vision and Pattern Recognition (CVPR).

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3608–3617.

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. Chartllama: A multimodal llm for chart understanding and generation. arXiv preprint arXiv:2311.16483.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. 2024. Cogagent: A visual language model for gui agents. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14281–14290.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6700–6709.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. arXiv preprint arXiv:2310.06825.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2901–2910.

Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. 2017. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In 2017 IEEE International Conference on Robotics and Automation (ICRA), page 746–753. IEEE Press.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. DVQA: Understanding data visualizations via question answering. In CVPR.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. FigureQA: An annotated figure dataset for visual reasoning. arXiv preprint arXiv:1710.07300.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In ECCV.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. 2024. Tablevqa-bench: A visual question answering benchmark on multiple table domains. arXiv preprint arXiv:2404.19205.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024. LLaVA-OneVision: Easy visual task transfer. arXiv preprint arXiv:2408.03326.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In International conference on machine learning, pages 19730–19742. PMLR.

Shengzhi Li and Nima Tajbakhsh. 2023. Scigraphqa: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. arXiv preprint arXiv:2308.03349.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In NeurIPS.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In NeurIPS.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In ICLR.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. OK-VQA: A visual question answering benchmark requiring external knowledge. In CVPR.

Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In ACL.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. InfographicVQA. In WACV.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. DocVQA: A dataset for VQA on document images. In WACV.

Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models.

Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. PlotQA: Reasoning over scientific plots. In WACV.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In ICDAR.

Srija Mukhopadhyay, Adnan Qidwai, Aparna Garimella, Pritika Ramu, Vivek Gupta, and Dan Roth. 2024. Unraveling the truth: Do VLMs really understand charts? a deep dive into consistency and robustness. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 16696–16717, Miami, Florida, USA. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. arXiv preprint arXiv:2303.08774.

Ajay Patel, Colin Raffel, and Chris Callison-Burch. 2024. DataDreamer: A tool for synthetic data generation and reproducible LLM workflows. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3781–3799, Bangkok, Thailand. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In ICML.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Jonathan Roberts, Kai Han, Neil Houlsby, and Samuel Albanie. 2024. Scifibench: Benchmarking large multimodal models for scientific figure interpretation. arXiv preprint arXiv:2405.08807.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-OKVQA: A benchmark for visual question answering using world knowledge. In ECCV.

Risa Shinoda, Kuniaki Saito, Shohei Tanaka, Tosho Hirasawa, and Yoshitaka Ushiku. 2024. Sbs figures: Pre-training figure qa from stage-by-stage synthesized images. arXiv preprint arXiv:2412.17606.

Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Divvala, and Ali Farhadi. 2016. Figureseer: Parsing result-figures in research papers. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14, pages 664–680. Springer.

Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA models that can read. In CVPR.

Hrituraj Singh and Sumit Shekhar. 2020. Stl-cqa: Structure-based transformers with localization and encoding for chart question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 3275–3284.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. arXiv preprint arXiv:2409.12183.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In NeurIPS.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. Advances in Neural Information Processing Systems, 34:200–212.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824–24837.

Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Min Dou, Botian Shi, Junchi Yan, et al. 2024. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. arXiv preprint arXiv:2402.12185.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks

in real computer environments. Preprint, arXiv:2404.07972.

Zhengzhuo Xu, Sinan Du, Yiyan Qi, Chengjin Xu, Chun Yuan, and Jian Guo. 2023. Chartbench: A benchmark for complex visual reasoning in charts. arXiv preprint arXiv:2312.15915.

Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, et al. 2024. xGen-MM (BLIP-3): A family of open large multimodal models. arXiv preprint arXiv:2408.08872.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. Preprint, arXiv:2306.15895.

Jieyu Zhang, Le Xue, Linxin Song, Jun Wang, Weikai Huang, Manli Shu, An Yan, Zixian Ma, Juan Carlos Niebles, Caiming Xiong, et al. 2024. Provision: Programmatically scaling vision-centric instruction data for multimodal language models. arXiv preprint arXiv:2412.07012.

A Implementation Details

A.1 Prompts

We provide the prompt templates in Figure 8 for the HTMLDocumentPipeline as an example to illustrate the prompts used across our code-guided synthetic data generation pipelines.
我们提供了图 8 中的 HTMLDocumentPipeline 提示模板，作为示例来说明我们在代码引导的合成数据生成管道中使用的提示。

A.2 Rendering Tools and Pipelines

We design 20 generation pipelines built on 11 rendering tools to support the creation of nine categories of text-rich images:(1) Charts: Matplotlib VegaLite, Plotly, LaTeX, HTML; (2) Documents: LaTeX, HTML; (3) Tables: LaTeX, Matplotlib, Plotly, HTML; (4) Diagrams: Graphviz, LaTeX, Mermaid; (5) Math Problems: LaTeX; (6) Vector Graphics: SVG, Asymptote; (7) Music Sheets: LilyPond; (8) Electrical Circuits: LaTeX; (9) Chemical Structures: Rdkit. In addition, we implement a separate pipeline for generating pointing data using HTML as the rendering tool.
我们设计了基于 11 种渲染工具的 20 个生成管道，以支持创建九类富含文本的图像：(1) **图表**：Matplotlib VegaLite、Plotly、LaTeX、HTML；(2) **文档**：LaTeX、HTML；(3) **表格**：LaTeX、Matplotlib、Plotly、HTML；(4) **图表**：Graphviz、LaTeX、Mermaid；(5) **数学问题**：LaTeX；(6) **矢量图形**：SVG、Asymptote；(7) **乐谱**：LilyPond；(8) **电路图**：LaTeX；(9) **化学结构**：Rdkit。此外，我们还实现了一个单独的管道，使用 HTML 作为渲染工具生成指向数据。

A.3 Queries to Construct CoSyn-400K

Since CoSyn accepts textual queries to control data generation, we use a diverse set of queries for each type of text-rich image to ensure broad domain coverage. Below are some examples of the queries used to generate CoSyn-400K:

- Charts: Bar, Line, Pie, Diverge bar, Bubble, Scatter, Histogram, Area, Box plot, Heatmap, Error bar, Radar chart, Rose chart, Stem plot, Stairs plot, Violin chart, 2D contour, Distplots, Log plot, Ternary plots/contour, Candlestick charts, Time series, etc. (51 queries in total)
- Documents: Letter, Form, Report, Receipt, Invoice, Restaurant menu, Newsletter, Schedule, Manual, Brochure, Transaction document, Agenda, Memo, Financial report, Telephone records, Note, Budget, Meeting minutes, Bill, Catalog, Email, Fax, Policy document, Resume, Infographics, Process infographic, Statistical infographic, etc. (107 queries in total)
- Math Problems: Algebra, Counting, Probability, Geometry, Number theory, Precalculus, Prealgebra, Intermediate Algebra, Statistics, Functions, Complex numbers, Logarithms, Inequalities, Linear equations, Exponents, Series, College Algebra, Calculus, Advanced calculus, Linear algebra, Solid geometry, Analytic geometry, Polynomial arithmetic, etc. (110 queries in total)
- Tables: Financial table, Simple table, Pivot table, Comparison table, Timeline table, Decision table, Truth table, Lookup table, Periodic table, Statistical table, Timetable, Hierarchical table, Matrix table, Contingency table, Logarithmic table, Correlation table, etc. (35 queries in total)
- Diagrams: Flow chart, Directed graph, Undirected graph, Decision tree, Mind map, Gantt charts, Finite state machine, Quadrant chart, Chord diagrams, Network diagrams, Sankey diagram, Entity relationship diagram, Sequence diagrams, Bottom-up flow chart, Timeline, State diagram, Concept map, Family tree, Programming flowchart, etc. (34 queries in total)
- Vector Graphics: Visual intelligence test, Spatial intelligence test, Geometry, Solid geometry, Analytic geometry, Polynomial graphs, Trigonometry, Polar coordinates, Coordinate system, Topology, Graph theory, Plane geometry, Functions, Calculus, Vectors, Angles, Perimeter and area problems, etc. (36 queries in total)
- Sheet Music: Classical, Pop, Rock, Jazz, Blues, Hip Hop, Rap, Electronic, Country, Folk, Rhythm and blues, Soul, Reggae, Metal, Punk, Theme, Dance, etc. (34 queries in total)
- Electrical Circuits: Series, Parallel, Hybrid, Household appliances, Industrial appliances, Mobile device, Low-power appliances, High-power appliances, etc. (30 queries in total)

- Chemical Structures: Drug, Organic, Inorganic, Protein, Acids, Bases, Gases, Liquids, Solids, Oxidizers, Flammable liquids, Toxic chemicals, Hazardous chemicals, Aromatic compounds, Aliphatic compounds, Polymers, Metals, Alloys, Electrolytes, etc. (100 queries in total)

A.4 Academic Datasets

During the supervised fine-tuning stage, we include academic datasets in addition to our synthetic datasets. Below, we provide details on the size of these datasets and the evaluation metrics used.
在有监督的微调阶段，除了我们的合成数据集外，我们还包含了学术数据集。下面，我们提供了这些数据集的规模和使用的评估指标的详细信息。

Dataset Size. The number in parentheses indicates the number of training images for each dataset: ChartQA (28.3K), DocVQA (39.5K), InfographicVQA (23.9K), AI2 Diagrams (11.4K), TextVQA (34.6K), VQAv2 (82.8K), GQA (72.1K), OK-VQA (9.0K), OCR-VQA (166.0K), A-OKVQA (17.1K), ScienceQA (6.2K), TabMWP (23.1K), ST-VQA (18.9K), TallyQA (133.0K), DVQA (200.0K), FigureQA (100.0K), PlotQA (160.0K). We downsample some very large synthetic datasets, such as DVQA, FigureQA, and PlotQA, to balance the dataset size. In total, we use approximately 1.1M images from academic datasets.
括号中的数字表示每个数据集的训练图像数量：ChartQA (28.3K)、DocVQA (39.5K)、InfographicVQA (23.9K)、AI2 Diagrams (11.4K)、TextVQA (34.6K)、VQAv2 (82.8K)、GQA (72.1K)、OK-VQA (9.0K)、OCR-VQA (166.0K)、A-OKVQA (17.1K)、ScienceQA (6.2K)、TabMWP (23.1K)、ST-VQA (18.9K)、TallyQA (133.0K)、DVQA (200.0K)、FigureQA (100.0K)、PlotQA (160.0K)。我们对一些非常大的合成数据集（如 DVQA、FigureQA 和 PlotQA）进行了下采样，以平衡数据集的大小。总共，我们使用了大约 1.1M 张来自学术数据集的图像。

Evaluation Metrics. We adopt their official evaluation metrics for the seven text-rich datasets. For ChartQA, we use relaxed correctness, which allows a 5% difference for float number answers. For DocQA and InfoQA, we report Average Normalized Levenshtein Similarity (ANLS). For TableVQA, we report the average performance across the four subsets (VTabFact, VWTQ, VWTQ-Syn, FinTabNetQA) using the metrics provided in this repo. We report the multiple choice accuracy for AI2D, VQA score (Goyal et al., 2017) for TextVQA, and SQuAD F1 score (Rajpurkar et al., 2018) for ScreenQA.
我们采用了七个富含文本数据集的官方评估指标。对于 ChartQA，我们使用宽松的正确性，允许浮点数答案有 5% 的差异。对于 DocQA 和 InfoQA，我们报告平均归一化 Levenshtein 相似度 (ANLS)。对于 TableVQA，我们使用此 repo 中提供的指标报告四个子集（VTabFact、VWTQ、VWTQ-Syn、FinTabNetQA）的平均性能。我们报告 AI2D 的多项选择准确率、TextVQA 的 VQA 分数 (Goyal et al., 2017) 以及 ScreenQA 的 SQuAD F1 分数 (Rajpurkar et al., 2018)。

A.5 Training Details

Image Preprocessing. We adopt the same image preprocessing as Molmo (Deitke et al., 2024), where each input image is cropped into multiple overlapping crops before being encoded by CLIP. During training, we limit the maximum number of crops to 12, but we increase it to 25 at testing time to accommodate the high resolution of text-rich images. This strategy boosts the inference performance without increasing training costs.
我们采用了与 Molmo (Deitke et al., 2024) 相同的图像预处理方法，其中每个输入图像在被 CLIP 编码之前被裁剪成多个重叠的裁剪区域。在训练期间，我们将最大裁剪数量限制为 12，但在测试时将其增加到 25，以适应富含文本图像的高分辨率。这种策略在不增加训练成本的情况下提高了推理性能。

Hyper Parameters. We set the maximum sequence length for training is 2304 tokens. We use the same learning rate of 1e-6 for the MLP connector, LLM, and visual encoder, with batch size 32. The best-performing model is trained for 60K steps with 200 warm-up steps and a cosine scheduler with an end factor of 0.1. All experiments are run on a single TPU v3-128.
我们将训练的最大序列长度设置为 2304 个 token。我们为 MLP 连接器、LLM 和视觉编码器使用相同的学习率 1e-6，批量大小为 32。最佳模型训练了 60K 步，包含 200 步预热步骤，

并使用余弦调度器 end 因子为 0.1。所有实验均在单个 TPU v3-128 上运行。

B Additional Analysis

We conduct additional analyses below to investigate further why our synthetic data can effectively enhance vision-language models.
我们进行了以下额外分析，以进一步研究为什么我们的合成数据能够有效增强视觉-语言模型。

Our synthetic data is more diverse. To quantify the diversity of images and text in our synthetic dataset $\mathcal{D} = \{(I, T)\}$, we propose the following two metrics to compute the diversity:

$$\text{Diversity}(\mathcal{D})_{\text{Image}} = \frac{1}{|\mathcal{D}|^2 - |\mathcal{D}|} \sum_{I_i \in \mathcal{D}} \sum_{I_j \in \mathcal{D}}^{i \neq j} (1 - \text{sim}(I_i, I_j)) \quad (1)$$

$$\text{Diversity}(\mathcal{D})_{\text{Text}} = \frac{1}{|\mathcal{D}|^2 - |\mathcal{D}|} \sum_{T_i \in \mathcal{D}} \sum_{T_j \in \mathcal{D}}^{i \neq j} (1 - \text{sim}(T_i, T_j)) \quad (2)$$

where sim(·) is the cosine similarity function. Both metrics compute the average pairwise cosine distance between the features of every instance in the dataset. For image diversity, we extract features using CLIP, while for text diversity, we use Sentence-BERT (Reimers, 2019) to obtain embeddings of question-answer pairs. Table 4 shows that our synthetic charts are significantly more diverse than those in existing datasets, such as FigureQA and ChartQA, in both image and text diversity.
我们的合成数据更具多样性。为了量化我们合成数据集 $\mathcal{D} = \{(I, T)\}$ 中图像和文本的多样性，我们提出了以下两个指标来计算多样性：

$$\text{Diversity}(\mathcal{D})_{\text{Image}} = \frac{1}{|\mathcal{D}|^2 - |\mathcal{D}|} \sum_{I_i \in \mathcal{D}} \sum_{I_j \in \mathcal{D}}^{i \neq j} (1 - \text{sim}(I_i, I_j)) \quad (3)$$

$$\text{Diversity}(\mathcal{D})_{\text{Text}} = \frac{1}{|\mathcal{D}|^2 - |\mathcal{D}|} \sum_{T_i \in \mathcal{D}} \sum_{T_j \in \mathcal{D}}^{i \neq j} (1 - \text{sim}(T_i, T_j)) \quad (4)$$

其中 sim(·) 是余弦相似度函数。这两个指标计算了数据集中每个实例特征之间的平均成对余弦距离。对于图像多样性，我们使用 CLIP 提取特征，而对于文本多样性，我们使用 Sentence-BERT (Reimers, 2019) 来获取问答对的嵌入。表 4 显示，我们的合成图表在图像和文本多样性方面显著优于现有数据集，如 FigureQA 和 ChartQA。

Dataset	Image Diversity	Text Diversity
FigureQA	0.268	0.567
DVQA	0.307	0.752
PlotQA	0.420	0.743
ChartQA	0.340	0.742
Ours (Charts)	0.596	0.823

Table 4: Compare image and text diversity across different chart datasets. We randomly sample 10K instances from each dataset to compute the results.

Diversity correlates with model performance. We observe that data diversity significantly affects model performance on downstream tasks. To investigate this, we compare synthetic chart data generated using only a single tool (Matplotlib) with charts generated by all five tools available in our CoSyn system. As shown in Table 5, using multiple tools results in higher image diversity and notably improved performance on ChartQA. This experiment underscores the importance of data diversity for enhancing the generalizability of models.
多样性与模型性能相关。我们观察到数据多样性显著影响模型在下游任务中的表现。为了研究这一点，我们比较了仅使用单一工具（Matplotlib）生成的合成图表数据与使用我们 CoSyn 系统中所有五种工具生成的图表数据。如表 5 所示，使用多种工具生成的图表数据具有更高的图像多样性，并且在 ChartQA 上的表现显著提升。这一实验强调了数据多样性对于增强模型泛化能力的重要性。

n. of Tools	Diversity	ChartQA		
		Average	Machine	Human
Single	0.572	73.9	66.5	81.5
Multiple	0.607	75.2	68.6	82.0

Table 5: Single vs. Multiple Rendering Tools for Data Generation. Each row uses the same number of 45K synthetic images. Single only uses Matplotlib, while Multiple involves four other rendering tools: HTML, LaTex, Plotly, and VegaLite.

Scaling the size of synthetic data. In addition to diversity, the scale of synthetic data also impacts model performance. As shown in Figure 9, increasing the number of synthetic chart images leads to improved performance on ChartQA. This demonstrates that scaling up synthetic data can further enhance VLMs on downstream tasks. Due to resource constraints, our final dataset consists of 400K images, which cost us about $8,000. Future work could explore scaling up the dataset size to

push the boundaries of synthetic data's potential.

扩展合成数据的规模。除了多样性，合成数据的规模也会影响模型性能。如图 9 所示，增加合成图表图像的数量可以提高 ChartQA 上的表现。这表明扩展合成数据可以进一步增强视觉-语言模型在下游任务中的表现。由于资源限制，我们的最终数据集包含 40 万张图像，花费了约 8000 美元。未来的工作可以探索扩展数据集规模，以推动合成数据潜力的边界。

LLM for Data Generation	ChartQA		
	Average	Machine	Human
GPT-4o	72.4	65.8	78.9
Claude-3.5-sonnet	77.2	71.0	83.8

Table 8: Compare the LLMs used for synthetic data generation. For both LLMs, we create 100K synthetic charts for fine-tuning the VLMs. We report the zero-shot evaluation results on ChartQA.

Compare LLMs for synthetic data generation. In the default setting, CoSyn uses Claude-3.5-sonnet as the underlying LLM for code generation. To highlight the importance of strong coding capabilities, we compare it with data generated by GPT-4o. As shown in Table 8, synthetic data generated by Claude-3.5-sonnet yields significantly better results than GPT-4o. Our qualitative observation reveals that GPT-4o has a higher failure rate in code generation, particularly for less common coding languages or libraries. This result emphasizes that a strong LLM is essential for the successful synthetic data generation for VLMs.

比较用于合成数据生成的 LLMs。在默认设置中，CoSyn 使用 Claude-3.5-sonnet 作为底层 LLM 进行代码生成。为了强调强大编码能力的重要性，我们将其与 GPT-4o 生成的数据进行了比较。如表 8 所示，Claude-3.5-sonnet 生成的合成数据显著优于 GPT-4o 生成的数据。我们的定性观察表明，GPT-4o 在代码生成中的失败率较高，尤其是在不太常见的编程语言或库中。这一结果强调了强大的 LLM 对于成功生成视觉-语言模型的合成数据至关重要。

Quantify the contributions of synthetic data. Table 7 presents the performance across benchmarks using different combinations of supervised fine-tuning data. A clear trend shows that synthetic data significantly contributes in both zero-shot and supervised settings. Adding our synthetic data consistently boosts performance on each benchmark.

量化合成数据的贡献。表 7 展示了使用不同监督微调数据组合在各个基准测试中的表现。一个明显的趋势是，合成数据在零样本和监督设置中都显著提升了性能。添加我们的合成数据持续提升了每个基准测试的表现。

The impact of Chain-of-thought reasoning. We compare the performance of CoT and short-answer prompts in Table 6. CoT reasoning improves performance on ChartQA, TableVQA, and NutritionQA, where questions require multi-hop and mathematical reasoning that aligns with the findings in language tasks (Sprague et al., 2024). However, short-answer prompts yield better results for the other five datasets due to their annotation biases favoring concise responses. CoT responses tend to be more verbose, which may not match the ground-truth answers exactly, resulting in a performance drop.

链式思维推理的影响。我们在表 6 中比较了链式思维推理（CoT）和简短回答提示的表现。CoT 推理在 ChartQA、TableVQA 和 NutritionQA 上提升了性能，这些任务中的问题需要多跳和数学推理，这与语言任务中的发现一致 (Sprague et al., 2024)。然而，简短回答提示在其他五个数据集上表现更好，因为这些数据集的注释偏向于简洁的回答。CoT 回答往往更为冗长，可能无法完全匹配真实答案，从而导致性能下降。

Document Pointing Task. To further validate the effectiveness of our synthetic pointing data, we introduce DocPointQA[6], a new pointing task with 300 question-point pairs annotated from the DocVQA validation set (Figure 11). We compare models trained on human-annotated PixMo-point data (155K examples), our synthetic pointing data (65K examples), and their combination. Since DocPointQA requires multiple-point answers, we report precision, recall, F1 score, and L2 distance (lower is better) after mapping predicted points to ground truth, following the same setup as Molmo (Deitke et al., 2024). As shown in Table 9, the model trained on our synthetic data outperforms the one trained on PixMo-point. Performance improves even further when both datasets are combined, demonstrating the effectiveness of synthetic data in enhancing the pointing capabilities of vision-language models.

[6] https://huggingface.co/datasets/yyupenn/DocPointQA

Prompt Type	ChartQA	DocVQA	InfoVQA	TableVQA	AI2D	TextVQA	ScreenQA	NutritionQA
CoT	86.3	87.4	63.8	65.8	86.0	70.9	79.0	76.0
Short Answer	83.1	90.0	70.5	64.3	91.9	82.0	80.1	62.0

Table 6: Alation of using chain-of-thought (CoT) in prompts. CoT means letting the model provide reasoning steps before giving the final answer. Short Answer prompts the model to answer with as few words as possible.

FT Data	ChartQA	DocVQA	InfoVQA	TableVQA†	AI2D	TextVQA	ScreenQA†	Average
Aux only*	60.7	56.2	39.7	43.1	81.7	68.5	61.3	58.7
Syn only*	79.4	80.5	60.1	64.4	68.6	63.6	76.6	70.5
Aux + Syn*	80.8	82.9	59.8	64.9	83.9	72.7	78.1	74.7
Eval only	77.4	87.4	63.8	51.8	91.3	81.1	78.1	75.9
Eval + Aux	81.4	87.9	68.2	53.6	91.6	81.8	77.0	77.3
Eval + Aux + Syn	86.3	90.0	70.5	65.8	91.9	82.0	80.1	80.9

Table 7: Alation of the data selection for supervised fine-tuning. Aux, Syn, and Eval stand for auxiliary, synthetic, and evaluation datasets, respectively. The rows with * represent zero-shot models (without using any training examples from any of the evaluation datasets). The datasets with † are test-only datasets (no training splits), which means all numbers on these datasets are zero-shot performance.

文档指向任务。为了进一步验证我们合成指向数据的有效性，我们引入了 DocPointQA[7]，这是一个新的指向任务，包含从 DocVQA 验证集中标注的 300 个问题-点对（图 11）。我们比较了在人类标注的 PixMo-point 数据（15.5 万个样本）、我们的合成指向数据（6.5 万个样本）以及两者组合上训练的模型。由于 DocPointQA 需要多点回答，我们在将预测点映射到真实答案后，报告了精度、召回率、F1 分数和 L2 距离（越低越好），遵循与 Molmo (Deitke et al., 2024) 相同的设置。如表 9 所示，使用我们的合成数据训练的模型优于使用 PixMo-point 数据训练的模型。当两个数据集结合时，性能进一步提升，证明了合成数据在增强视觉-语言模型指向能力方面的有效性。

Pointing Data	Precision	Recall	F1	Distance ↓
PixMo-point	49.7	49.3	52.7	17.3
Synthetic (Ours)	63.8	66.1	62.8	9.2
Combined (Ours)	69.9	70.6	70.7	8.8

Table 9: Zero-shot Pointing on DocPointQA. We compare the models trained on different pointing data. Combined stands for combining PixMo-point (human-annotated) (Deitke et al., 2024) with our synthetic data.

C Qualitative Examples

Figure 10 and 11 show the examples from our annotated NutritionQA[8] and DocPointQA. Figures 12 - 18 list examples from the 9 categories of synthetic text-rich images. Figure 19 illustrates examples from the synthetic pointing dataset.
图 10 和 11 展示了我们标注的 NutritionQA 和 DocPointQA 的示例。图 12 - 18 列出了 9 类合成文本丰富图像的示例。图 19 展示了合成指向数据集的示例。

Use of AI Assistants. We use AI to fix some typos and grammar. Authors write all contents.
使用 AI 助手。我们使用 AI 来修正一些拼写和语法错误。所有内容均由作者撰写。

[7] https://huggingface.co/datasets/yyupenn/DocPointQA
[8] https://huggingface.co/datasets/yyupenn/NutritionQA

Topic Generation: You are an expert in document generation and have a broad knowledge of different topics.
My persona is: "PERSONA" I want you to generate NUM_TOPICS topics for FIGURE_TYPE that I will be interested in or I may see during my daily life given my persona.
Here are the requirements:
1. Each topic is a high-level summary of the contents in FIGURE_TYPE with some design details, e.g., "the utility bill for the month of January 2022 with a detailed breakdown of charges".
2. The topics should be diverse to help me generate varied documents. Each topic should be unique and not overlap with others.
3. The topics are conditioned on the document type. Please ensure the topics you provided can be best visualized in "FIGURE_TYPE".
4. All topics must be in English, even if the persona is non-English.
5. List NUM_TOPICS topics for "PERSONA" and separate them with a | character, e.g., topic1 | topic2 | ..... | topicN.
Do not include any additional text at the beginning or end of your response.

Data Generation: You are an expert in content creation and have broad knowledge about various topics.
My persona is: "PERSONA" I need some materials about "TOPIC", which can be used to generate a FIGURE_TYPE.
Here are the requirements:
1. The materials should be related to the topic and customized according to my persona. Its structure must be suitable for the FIGURE_TYPE.
2. The materials should be realistic, and the contents should be named using real-world entities. Do not use placeholder names like xxA, xxB, etc. Do not use template data like [Name], [Date], etc.
3. The materials should be diverse and contain information from different aspects of the topic to ensure the document is informative.
4. Do not provide too many materials. Just provide key pieces of information that are essential for a **one-page document.**
5. All materials must be in English, even if the persona is non-English.
Please provide the materials in JSON format without additional text at the beginning or end.

Code Generation: You are an expert web designer and are good at writing HTML to create documents.
My persona is: "PERSONA" I have some materials about TOPIC which can be used to generate a FIGURE_TYPE.
Here are the materials (JSON format):
<data> DATA </data>
Please use HTML and CSS to generate a FIGURE_TYPE using the data provided.
Here are the requirements:
1. **Style Requirements**: Feel free to use any CSS framework, libraries, JavaScript plugins, or other tools to create the document.
(1) Try to be creative and make the web page style, fonts, colors, borders and visual layout unique with CSS. Taking persona, topic, and document type into consideration when designing the document.
(2) Select the appropriate design scale (e.g., margins, page size, layout, etc) to ensure the information in the document is clear and easy to understand, with no text overlapping, etc.
(3) **Do not make the page too long or too sparse.** All contents should be in **one page**. This is very important.
2. **Code Requirements**:
(1) You need to hardcode the provided data into the HTML script to generate the document. Be careful with the syntax and formatting of the HTML.
(2) Put everything in one HTML file. Do not use external CSS or JavaScript files.
3. **Output Requirements**: Put "'html at the beginning and "' at the end of the script to separate the code from the text.
Please don't answer with any additional text in the script, your whole response should be the HTML code which can be directly executed.

Instruction Generation: You are an expert in data analysis and good at asking questions about documents. My persona is: "persona" I want you to generate some question-answer pairs of a FIGURE_TYPE about TOPIC, which I would ask. Instead of showing the document, I provide the data and the code that generates the document.
<data> DATA </data> <code> CODE </code>
Please come up with a list of *reasonable questions* that people will ask when they see the rendered document. Here are the requirements:
1. **Question Types**: All questions should be short-answer questions that are answerable based on the visual information in the document. All questions can be answered with a single word, phrase, or number. (as short as possible)
(1) **Information Retrieval questions** ask for specific information in the document, such as numbers, names, dates, titles, etc. The questions should cover different aspects (areas) of the document. This is the most common type of question.
(2) **Reasoning questions** require reasoning over multiple information in the document. These questions should be more challenging and require a deeper understanding of the document.
(3) **Document Type-specific questions** are questions that are specific and unique to this document type FIGURE_TYPE. These questions should be tailored to the content and structure of the document.
2. **Response Format**: Use | to separate the question, explanation, and concise answer for each example.
(1) Follow this format: question | explanation | concise answer, e.g., what is the total revenue? | The total revenue is the sum of all revenue sources in the document, which is $2000 + $3000 + $5000 = $10000. | $10000
(2) Separate the question-answer pairs by double newlines.
question1 | explanation1 | answer1
question2 | explanation2 | answer2...
(3) Do not provide too many questions, 5-10 questions are enough. Focus on the diversity and quality of the questions. Try to cover different aspects of the document.
(4) The concise answer should be as short as possible and directly answer the question. The answer should be faithful and exactly the same as what you would expect to see in the document, don't rephrase it. All words in the answer should be processed in natural language, no coding terms/characters.
Please follow the format strictly and do not include any additional text at the beginning or end of your response.

Figure 8: Prompt templates used for HTML Document Pipeline, including all four stages of generation: topic, data, code, and instruction.

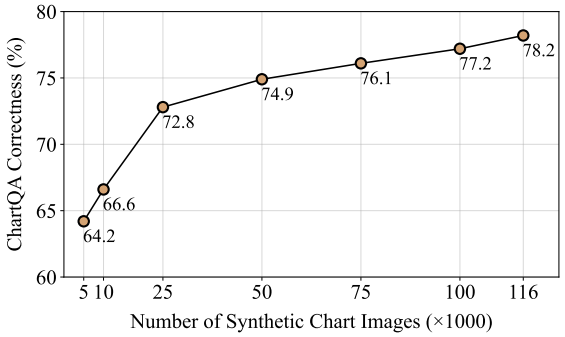
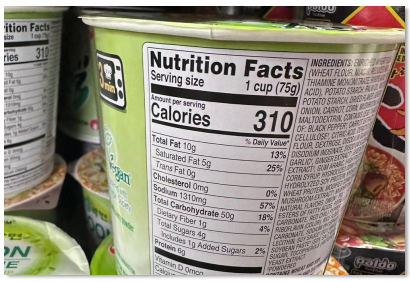


Figure 9: Scaling the Size of Synthetic Data. We evaluate the zero-shot performance on ChartQA of models fine-tuned on increasing numbers of synthetic images.

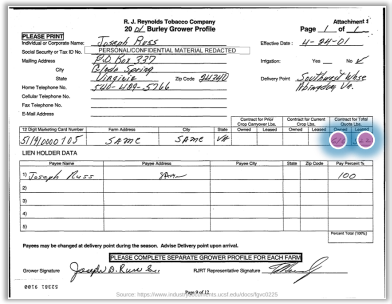
Q: How many servings do I need to fulfill the daily value of Cholesterol? **A**: 2.



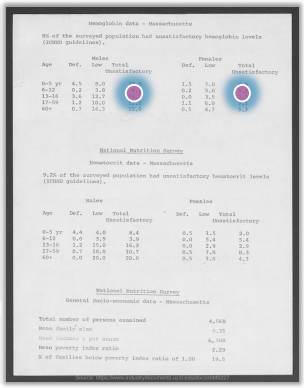
Q: I have taken 1000mg of sodium today. Can I eat this without exceeding the suggested daily value? **A**: No.

Q: How many capsules per container? **A**: 56.

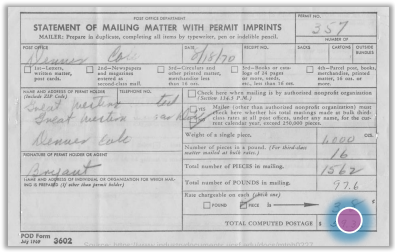
Figure 10: Examples from our newly collected NutritionQA dataset.



Q: Help me find the total of owned and leased quota pounds.

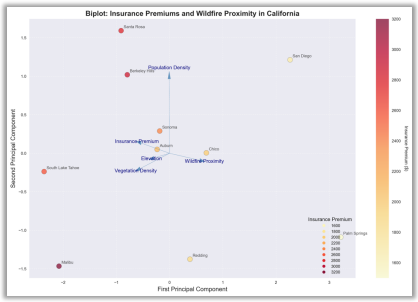


Q: Point out the total unsatisfactory hemoglobin levels for both males and females aged 13-16.

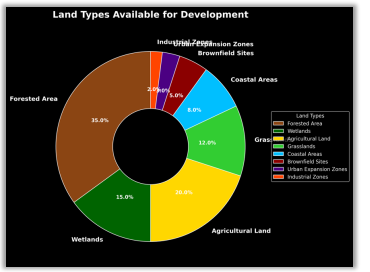


Q: Show me the total computed postage.

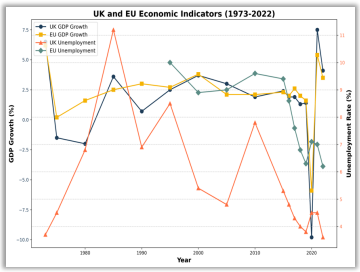
Figure 11: Examples from our newly collected DocPointQA dataset.



Q: Which neighborhood has the highest insurance premium? **A**: Malibu.

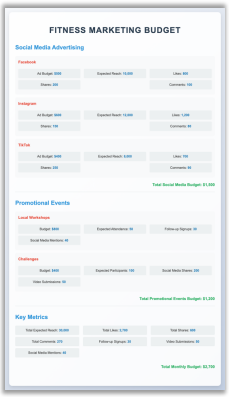


Q: Which land type has the smallest percentage of available land for development? **A**: Industrial Zones.

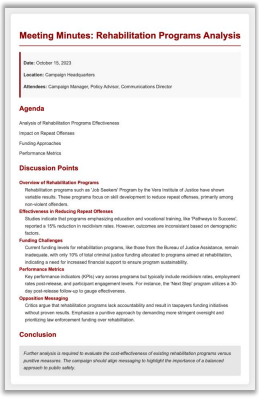


Q: How did the unemployment rates in the UK and EU compare in 2005? **A**: EU had higher.

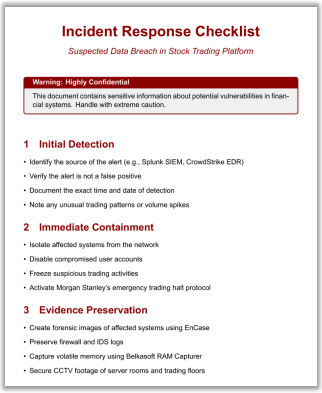
Figure 12: Randomly selected examples from our synthetic chart data.



Q: What is the total monthly budget for the marketing initiatives?
A: $2,700.



Q: What is one key indicator used in rehabilitation programs?
A: Recidivism rates



Q: What is the first step in the checklist?
A: Initial Detection.

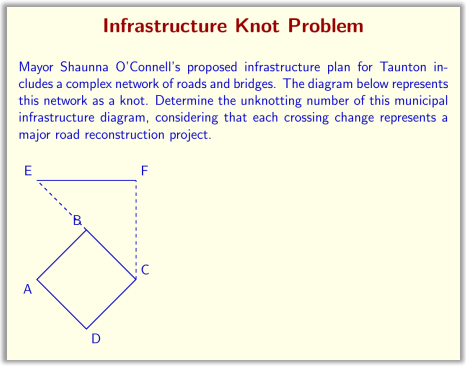
Category	Year			
	2020	2021	2022	2023
Engine Components	1500	1800	2000	2200
Body Parts	1200	1400	1600	1700
Interior Trim	900	950	1200	1300

Q: Which year had the lowest sales for Interior Trim? **A**: 2020

Q: Which month had the highest customer foot traffic? **A**: December.

Month	Discount (%)	Foot Traffic
January	15	1200
February	20	1500
March	10	800
April	25	1800
May	30	2200
June	5	650
July	15	1300
August	20	1600
September	10	900
October	25	2000
November	35	2500
December	40	3000

Q: What is the status of of Château de Chambord?
A: Excellent.



Q: Can you answer this question?
A: 2

Consider a simplified model of societal change where the rate of adoption of a new idea (x) is governed by the differential equation:

$$\frac{dx}{dt} = rx(1-x) - \frac{ax^2}{1+x^2}$$

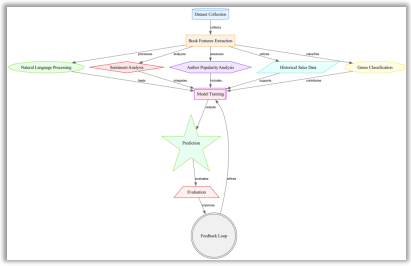
where r represents the growth rate and a represents the resistance to change. For what value of a does the system undergo a saddle-node bifurcation when $r = 1$?

Q: Give your solution to this math problem.
A: $a = 1$

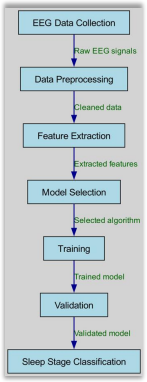
Figure 13: Randomly selected examples from our synthetic document data.

Figure 14: Randomly selected examples from our synthetic table data.

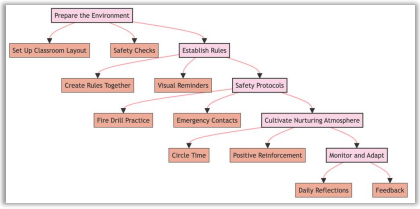
Figure 15: Randomly selected examples from our synthetic math data.



Q: What steps are involved immediately before making a prediction?
A: Model Training and Prediction.

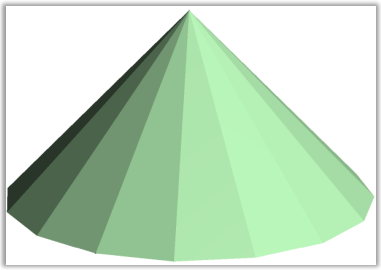


Q: What is the final outcome of the process illustrated in the diagram? **A**: Sleep Stage Classification.

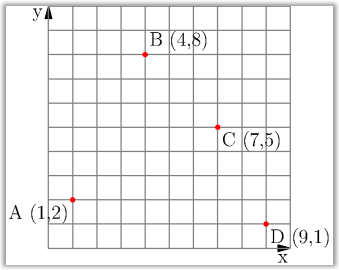


Q: Which safety practice involves emergency contacts? **A**: Emergency Contacts.

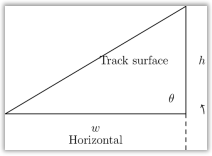
Figure 16: Randomly selected examples from our synthetic diagram data.



Q: You estimate its height to be 100 meters and its base radius to be 150 meters. What is the volume of this volcanic cone in cm^3? **A**: \$750,000\pi\$ cm^3.

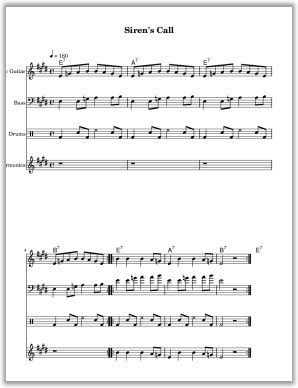


Q: What is the shortest total distance to visit all four stores, starting and ending at Store A? **A**: 23.48 units



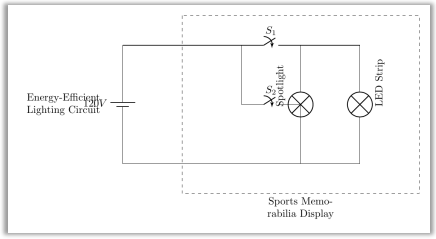
Q: The turn has a angle of 33°, and the width of the track is 58 feet. What is the height difference between the inside and outside edges of the track? **A**: 38 feet

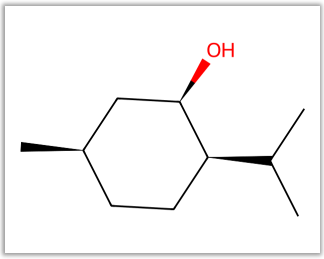
Figure 17: Randomly selected examples from our synthetic vector graphic data.



Q: What is the key signature of this music? **A**: E major.

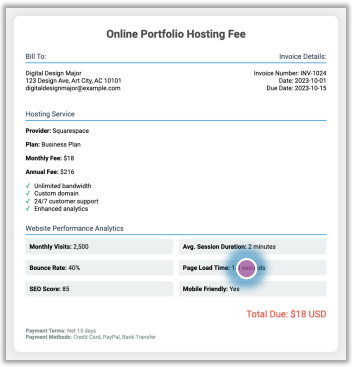
Q: What is the voltage of this circuit? **A**: 120V.



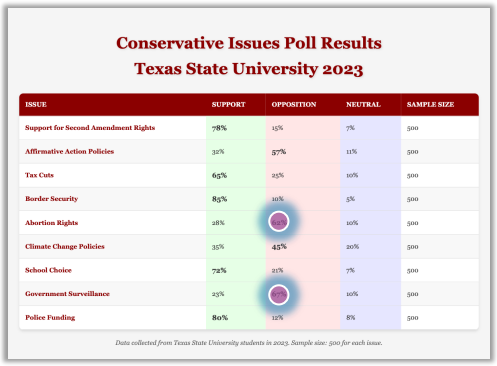


Q: What is the main use of menthol in throat lozenges? **A**: Soothing effect.

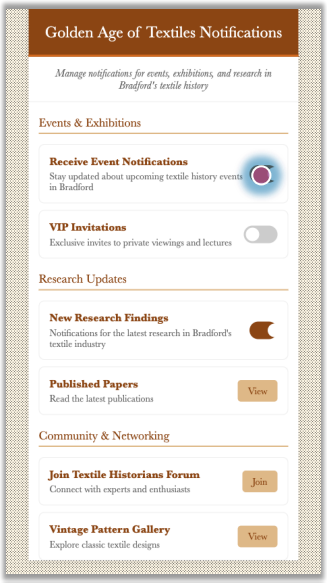
Figure 18: Randomly selected examples from our synthetic sheet music, circuits and chemical structures.



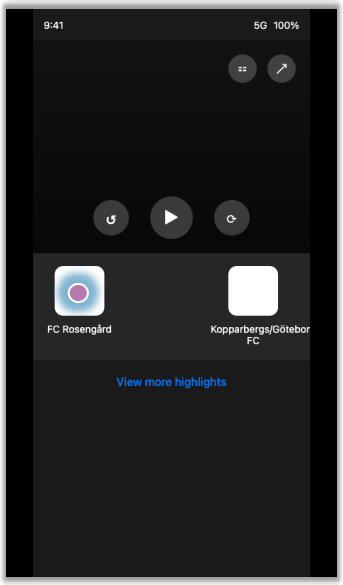
Q: Point out the page load time performance of the website, which indicates the site's speed optimization
A: (64.1, 73.7)



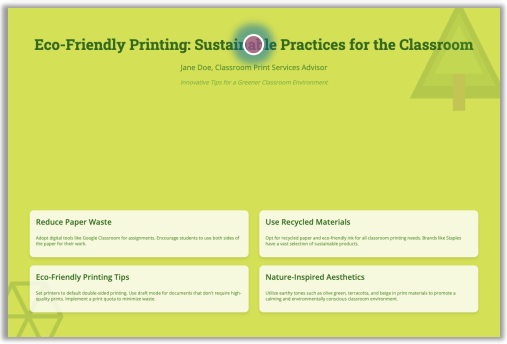
Q: Generate points for all issues where opposition exceeds 60%.
A: (56.2, 61.1) (56.2, 81.4)



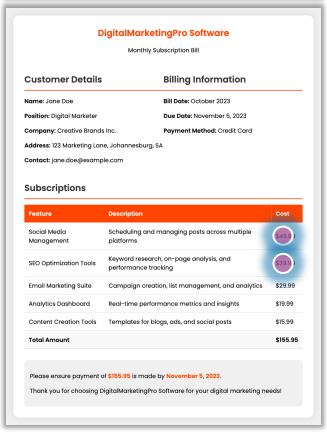
Q: Provide a point where users can toggle event notifications. **A**: (35.7, 76.9)



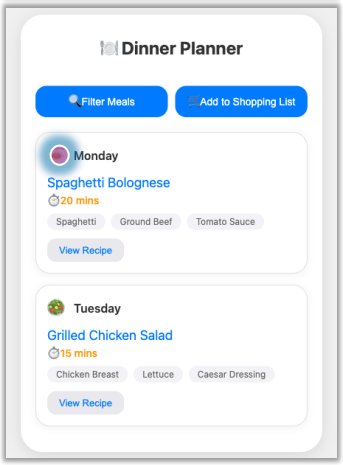
Q: Identify which logo represents FC Rosengård in this match. A: (22.1, 56.2)



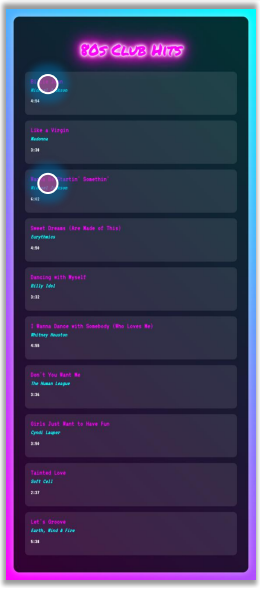
Q: Point out the main title of the book.
A: (50.0, 10.9)



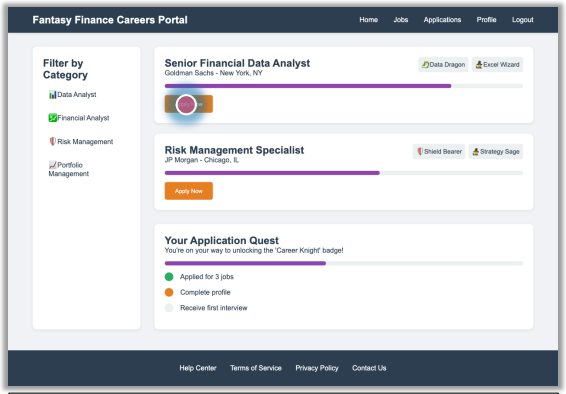
Q: Highlight all services that cost more than $30.
A: (88.7, 54.6) (88.6, 61.1)



Q: Show me the pasta icon for the Italian dish.
A: (15.0, 26.3)



Q: Find Michael Jackson songs that are longer than 4 minutes. **A**: (9.8, 12.7) (9.8, 26.4)



Q: Point the button to submit application for the Senior Financial Data Analyst. **A**: (32.8, 25.5)

Figure 19: Randomly selected examples from our synthetic pointing data.