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). However, VLMs often struggle in these domains due to the scarcity of diverse text-rich vision-language data. 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. 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. 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. Using CoSyn, we constructed a dataset comprising 400K images and 2.7M rows of vision-language instruction-tuning data. 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. 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.

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). 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. 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). 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).

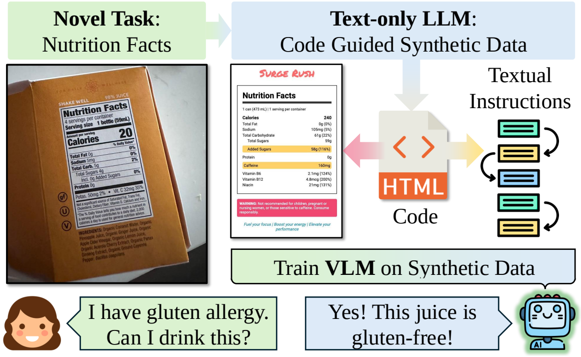

Novel Task:
Nutrition Facts
Text-only LLM:
Code Guided Synthetic Data
Surge Rush
Nutrition Facts
Textual
Instructions
HTML
Code
Train **VLM** on Synthetic Data
I have gluten allergy. Can I drink this?
Yes! This juice is gluten-free!

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.

To address these challenges and inspired by the fact that text-rich images are typically rendered from code, we develop **Co**de Guided **Syn**thetic data generation system (**CoSyn**), a flexible framework for generating diverse synthetic text-rich multimodal data for vision-language instruction tuning. As illustrated in Figure 2, CoSyn can generate multimodal data for various target domains from a short natural language query, such as *book covers*. 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). 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.

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. We comprehensively evaluate the effectiveness of training on CoSyn-generated synthetic data across seven text-rich VQA benchmarks. 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. Notably, training on CoSyn synthetic data enables sample-efficient learning, achieving stronger performance with less data. In addition, CoSyn can synthesize chain-of-thought (CoT) reasoning data (Wei et al., 2022), improving performance on tasks requiring multi-hop reasoning. 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. 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.

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. 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. Open-source VLMs perform poorly on this novel task, even after training on millions of images. However, by training on CoSyn-400K, our model adapts strongly to this novel domain in a zero-shot setting with significantly less training data. 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. This highlights CoSyn's data efficiency and ability to help VLMs adapt to new domains through targeted synthetic data generation.

Finally, beyond the standard VQA task, we use CoSyn to generate synthetic *pointing* training data, which is particularly useful in agentic tasks. 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). Our model trained on synthetic pointing data achieves state-of-the-art performance on the ScreenSpot click prediction benchmark (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.

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. 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). More recent architectures have converged on using MLP layers to project visual features into the language space (Liu et al., 2023). However, these architectures are often imbalanced, with the language backbone substantially larger than the visual encoder. 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). Our work addresses this issue by generating high-quality multimodal data for text-rich images.

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). 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. 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.

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). This approach has been applied to chart and plot VQA typically using a limited small set of chart types and by instantiating handcrafted question

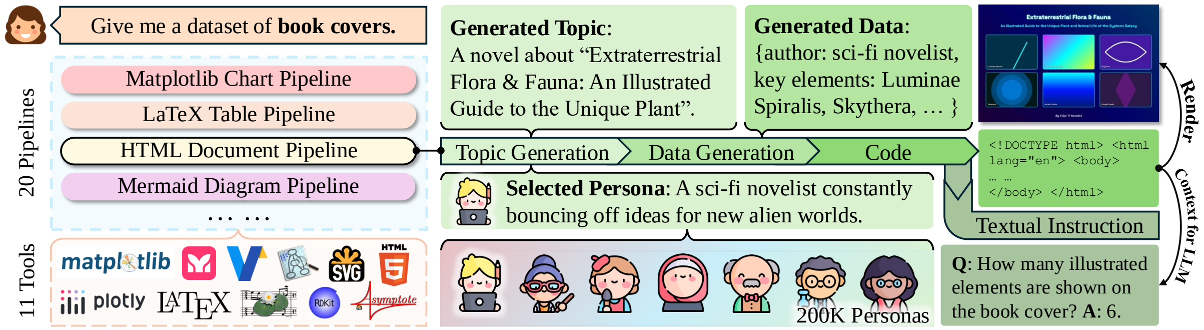

Give me a dataset of book covers.
20 Pipelines
Matplotlib Chart Pipeline
LaTeX Table Pipeline
HTML Document Pipeline
Mermaid Diagram Pipeline
… …
11 Tools
matplotlib
plotly
LaTeX
SVG
HTML
RDKit
Asymptote
Generated Topic: A novel about "Extraterrestrial Flora & Fauna: An Illustrated Guide to the Unique Plant".
Generated Data: {author: sci-fi novelist, key elements: Luminae Spiralis, Skythera, … }
Topic Generation
Data Generation
Code
Selected Persona: A sci-fi novelist constantly bouncing off ideas for new alien worlds.
200K Personas
Extraterrestrial Flora & Fauna
<!DOCTYPE html> <html lang="en"> <body> … … </body> </html>
Textual Instruction
Q: How many illustrated elements are shown on the book cover? A: 6.
Render
Context for LLM

Figure 2: The overview of our **Co**de Guided **Syn**thetic 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.

templates (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 or Q&A pairs from table or text descriptions associated with charts to train VLMs. Other recent approaches, similar to the procedure in this work, explore generating data and code to render synthetic charts (Han et al., 2023; Shinoda et al., 2024; Xia et al., 2024) while using the data and code representation to generate annotations and Q&A. 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. In our work, we expand the scope of our generation beyond charts to encompass a wider range of diverse text-rich images.

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). $\mathcal{D}_q$ is used to train a VLM to improve its ability to understand images related to q. The core idea of our approach is using code C as the intermediate representation to bridge the image and text. 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, $P\left(I|C\right)$. $P_{\text{LM}}\left(T|C\right)$ uses code C (without the image) as context for an LLM to generate the textual instruction-tuning data.

4 CoSyn System

Figure 2 illustrates the workflow of our **Co**de-Guided **Syn**thetic data generation system (**CoSyn**). The system takes a language input, such as "generate a dataset of book covers", and outputs a multimodal dataset. Based on the input query, CoSyn selects one of 20 generation pipelines built on 11 rendering tools. The process starts with topic generation, conditioned on a sampled persona that guides the style and content. Next, the system generates data content and converts it into code, which is then executed to render synthetic images. Finally, using the code as context, we prompt the LLM to generate corresponding textual instructions.

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.

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. For example, Matplotlib, Plotly, and Vega-Lite are used to create different types of charts. LaTeX and HTML are used for documents and tables, while Mermaid and Graphviz generate diagrams. We utilize SVG and Asymptote to create vector graphics and math-related content. For specialized tasks, we rely on Lilypond to generate music sheets and RDKit for chemical structures. We implement customized functions for each tool to execute LLM-generated code and obtain corresponding rendered images. These tools collectively enable CoSyn to produce a wide range of high-quality, text-rich synthetic images.

Pipelines. We design 20 pipelines based on 11 ren-

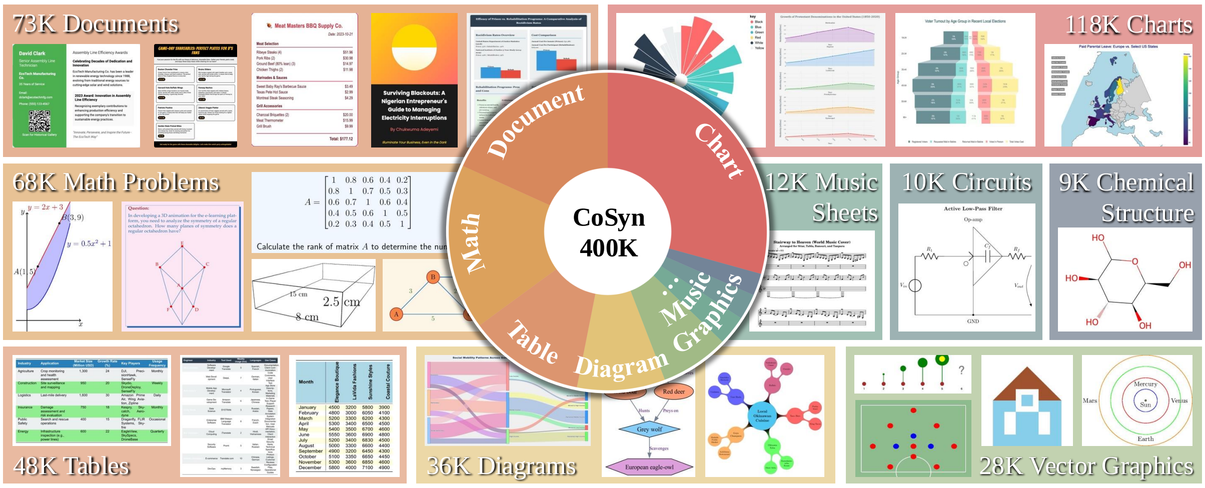


Figure 3: Our CoSyn-400K dataset consists of 9 categories of text-rich images with 2.7M instruction-tuning data. More qualitative examples, along with question-answer annotations, are available in Figure 12 -18 in Appendix C.

dering tools.[1] 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. Each stage is controlled by a prompt customized for image category and rendering tool. Figure 8 shows all prompts of the HTML Document pipeline.

Use personas for diversity. 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. Recent work (Ge et al., 2024) shows that incorporating personas in prompts can improve diversity by enabling models to generate from varied perspectives. CoSyn adopts personas to enhance diversity during the Topic Generation stage. Each persona is a short sentence describing a personality or identity. 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. We use the 200K personas released by Ge et al. (2024).

Implementation details. 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. 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. For the data and code generation stages, we use Claude-3.5-Sonnet, which performs well in coding tasks (Anthropic, 2024b). For instruction-tuning data generation, we select GPT-4o-mini (OpenAI, 2023) for its cost efficiency.

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. Since CoSyn is controlled via language inputs, it can easily generate diverse, fine-grained image types by varying the input queries. For instance, we use over 100 queries to generate document data covering *receipts*, *resumes*, *meal plans*, etc. Some queries used for CoSyn-400K are provided in Appendix A.3. This ensures that our dataset covers a broad range of domains. The following sections validate how our synthetic datasets enhance the ability of VLMs to understand text-rich images.

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. This section introduces the architecture of our model, training strategy, datasets we used, baselines for comparison, and other details on implementation.

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

[1] Some tools are used in multiple pipelines, e.g., HTML is used for generating documents, tables, and charts.

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.

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.

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:

- **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.[2]
- **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.
- **Synthetic Datasets.** As introduced in Sec 4 and also shown in Figure 3, our synthetic datasets include 400K text-rich images from 9 categories.

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.

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).

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.

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 synthetic pointing data in improving VLMs for web agent tasks (Sec 6.3).

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

[2] TableVQA is an eval-only benchmark (no training split), and we do not use the training split from ScreenQA.

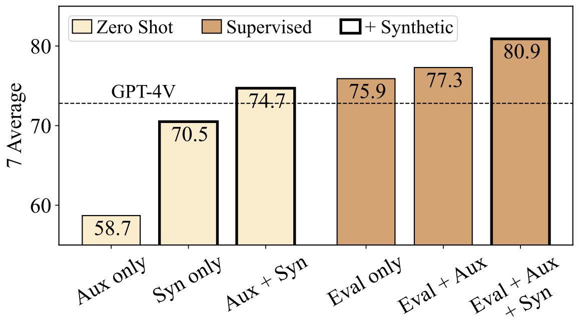


Figure 4: **Ablation on training data selection.** Aux, Syn, and Eval stand for auxiliary, synthetic, and evaluation datasets, respectively. We report the average score on eight benchmarks. The detailed performance breakdown on each benchmark is in Table 7.

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.

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.

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.

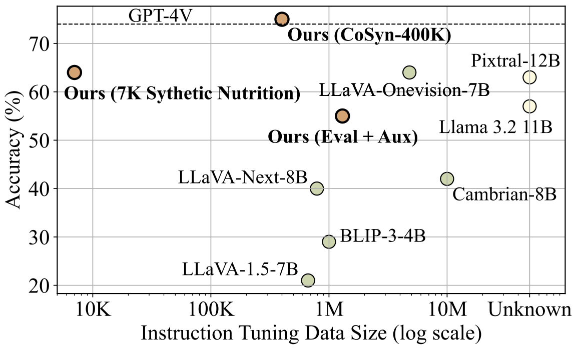


Figure 5: **Zero shot performance on NutritionQA.** The x-axis denotes the number of training examples used for the instruction-tuning stage. The models on the upper left side demonstrate better data efficiency.

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.

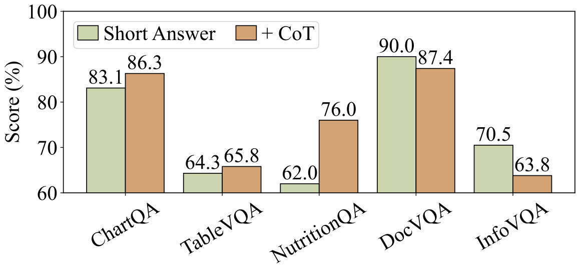


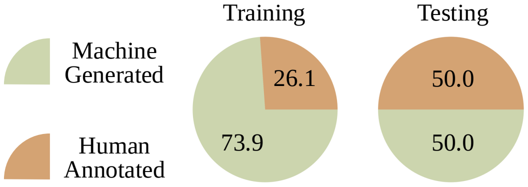
Figure 6: **Ablation of using Chain-of-Thought reasoning.** Short Answer represents prompting model to output the answer as short as possible. + CoT stands for providing Chain-of-Thought reasoning before giving the final answer. Results on all datasets are in Table 6.

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:

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.

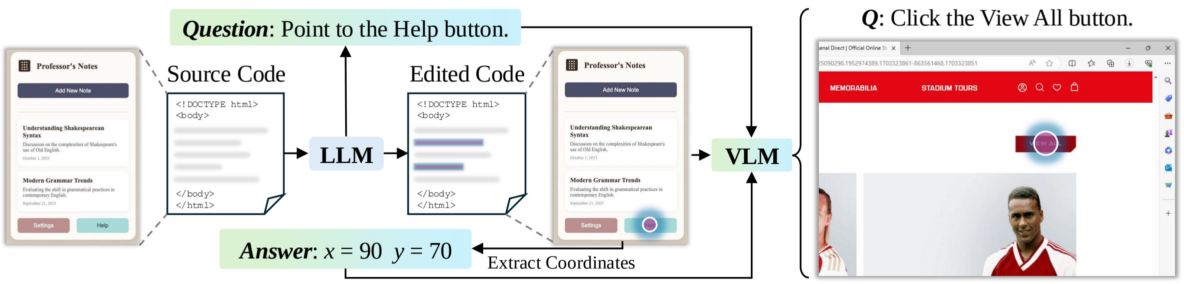


ChartQA	Average	Machine	Human	Δ ↓
PaliGemma-3B	71.4	88.5	54.2	34.3
ChartPali-5B	77.3	**93.7**	60.9	32.8
Ours (w/o Syn)	81.4	92.2	70.4	21.8
Ours (w/ Syn)	**86.3**	93.4	**79.1**	**14.2**

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.



(a) Synthetic Pointing Data Generation **(b) Real Screenshot Pointing**

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.

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.

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.[3] 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.

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.

	Mobile		**Desktop**		**Web**		**Avg**
Model	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.

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.

7 Conclusion

In this work, we introduced CoSyn, a framework for generating synthetic data that significantly enhances VLM performance on text-rich image understanding. Our comprehensive analysis highlights the advantages of synthetic data for domain generalization, data efficiency, and bias mitigation. 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.

[3] The coordinates of points are normalized to (0, 100) to mitigate the influence of image resolution.

Limitation

The effectiveness of synthetic data depends heavily on the quality and diversity of the prompts and rendering pipelines used for data generation. For highly specialized or underrepresented domains, generating sufficiently diverse data remains challenging and may require careful prompt engineering or additional customization of rendering tools. Targeted synthetic data generation may be essential for certain tasks to achieve adequate performance, and ensuring relevance and coverage still requires domain-specific expertise. 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. Finally, our current synthetic data is limited to English and may require further extension for multilingual support.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

B Additional Analysis

We conduct additional analyses below to investigate further why our synthetic data can effectively enhance vision-language models.

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:

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

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

where $\text{sim}(\cdot)$ 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.

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.

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.

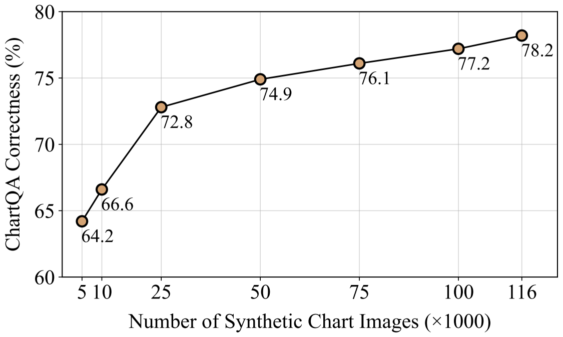


Figure 9: **Scaling the Size of Synthetic Data.** We evaluate the *zero-shot* performance on ChartQA of models fine-tuned on increasing numbers of synthetic images.

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.

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

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.

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.

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.

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.

Document Pointing Task. To further validate the effectiveness of our synthetic pointing data, we introduce DocPointQA, 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.

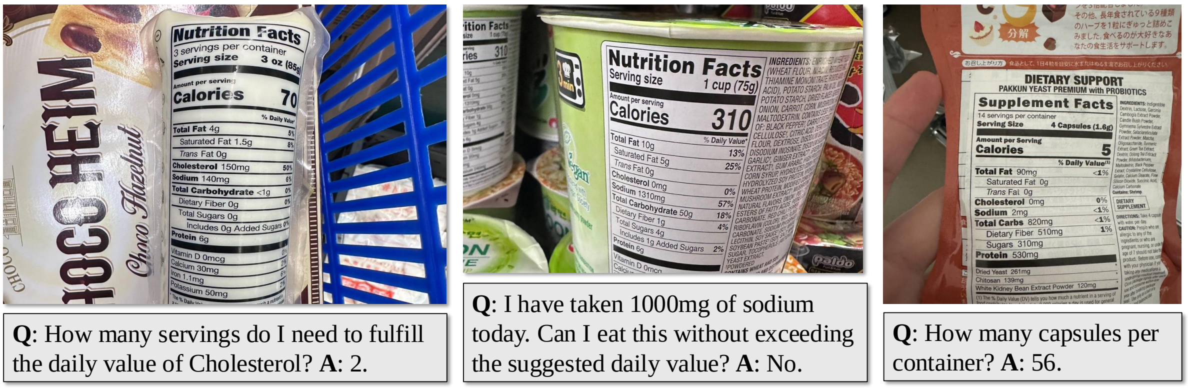
Pointing Data	Precision	Recall	F1	Distance ↓
PixMo-point	49.7	49.3	52.7	17.3
Synthetic (Ours)	63.8	66.1	62.8	9.2
Combined (Ours)	**69.9**	**70.6**	**70.7**	**8.8**

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 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.

Use of AI Assistants. We use AI to fix some typos and grammar. Authors write all contents.

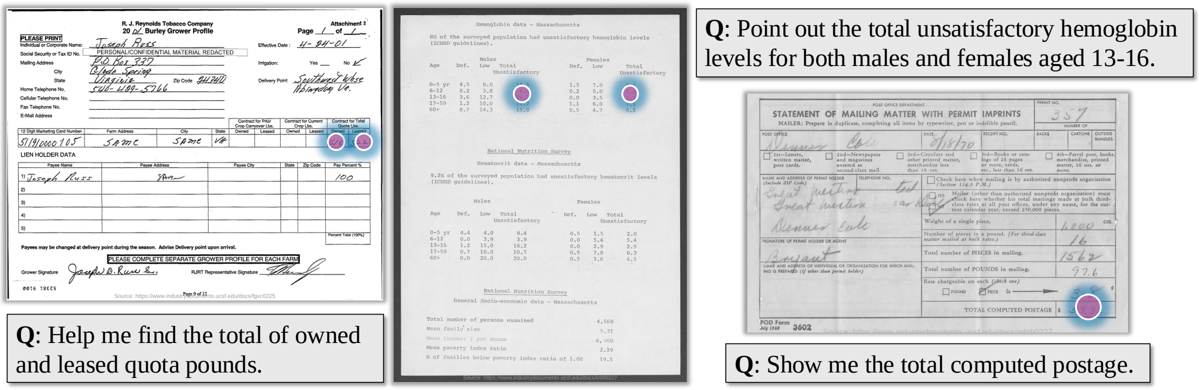


Q: How many servings do I need to fulfill the daily value of Cholesterol? **A**: 2.

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.

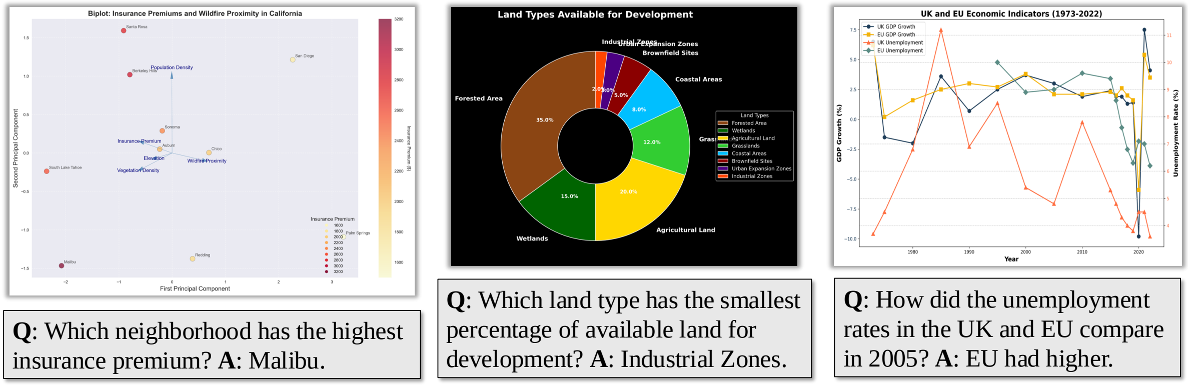


Q: Help me find the total of owned and leased quota pounds.

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.

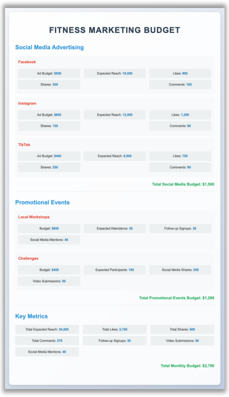


Q: Which neighborhood has the highest insurance premium? **A**: Malibu.

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.



FITNESS MARKETING BUDGET

Social Media Advertising

Facebook

Ad Budget: $500 | Expected Reach: 10,000 | Likes: 800
Shares: 200 | Comments: 100

Instagram

Ad Budget: $600 | Expected Reach: 12,000 | Likes: 1,200
Shares: 150 | Comments: 80

TikTok

Ad Budget: $400 | Expected Reach: 8,000 | Likes: 700
Shares: 250 | Comments: 90

Total Social Media Budget: $1,500

Promotional Events

Local Workshops

Budget: $800 | Expected Attendance: 50 | Follow-up Signups: 30
Social Media Mentions: 40

Challenges

Budget: $400 | Expected Participants: 100 | Social Media Shares: 200
Video Submissions: 50

Total Promotional Events Budget: $1,200

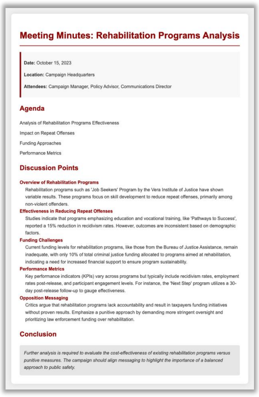
Key Metrics

Total Expected Reach: 30,000 | Total Likes: 2,700 | Total Shares: 600
Total Comments: 270 | Follow-up Signups: 30 | Video Submissions: 50
Social Media Mentions: 40

Total Monthly Budget: $2,700

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



Meeting Minutes: Rehabilitation Programs Analysis

Date: October 15, 2023

Location: Campaign Headquarters

Attendees: Campaign Manager, Policy Advisor, Communications Director

Agenda

Analysis of Rehabilitation Programs Effectiveness

Impact on Repeat Offenses

Funding Approaches

Performance Metrics

Discussion Points

Overview of Rehabilitation Programs

Rehabilitation programs such as 'Job Seekers' Program by the Vera Institute of Justice have shown variable results. These programs focus on skill development to reduce repeat offenses, primarily among non-violent offenders.

Effectiveness in Reducing Repeat Offenses

Studies indicate that programs emphasizing education and vocational training, like 'Pathways to Success', reported a 15% reduction in recidivism rates. However, outcomes are inconsistent based on demographic factors.

Funding Challenges

Current funding levels for rehabilitation programs, like those from the Bureau of Justice Assistance, remain inadequate, with only 10% of total criminal justice funding allocated to programs aimed at rehabilitation, indicating a need for increased financial support to ensure program sustainability.

Performance Metrics

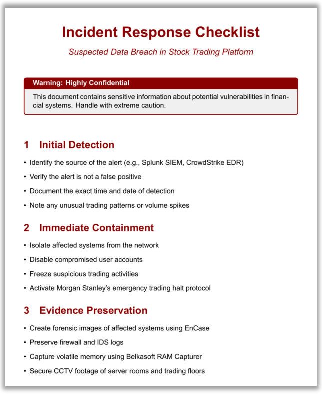
Key performance indicators (KPIs) vary across programs but typically include recidivism rates, employment rates post-release, and participant engagement levels. For instance, the 'Next Step' program utilizes a 30-day post-release follow-up to gauge effectiveness.

Opposition Messaging

Critics argue that rehabilitation programs lack accountability and result in taxpayers funding initiatives without proven results. Emphasize a punitive approach by demanding more stringent oversight and prioritizing law enforcement funding over rehabilitation.

Conclusion

Further analysis is required to evaluate the cost-effectiveness of existing rehabilitation programs versus punitive measures. The campaign should align messaging to highlight the importance of a balanced approach to public safety.



Incident Response Checklist

Suspected Data Breach in Stock Trading Platform

Warning: Highly Confidential

This document contains sensitive information about potential vulnerabilities in financial systems. Handle with extreme caution.

1 Initial Detection

- Identify the source of the alert (e.g., Splunk SIEM, CrowdStrike EDR)
- Verify the alert is not a false positive
- Document the exact time and date of detection
- Note any unusual trading patterns or volume spikes

2 Immediate Containment

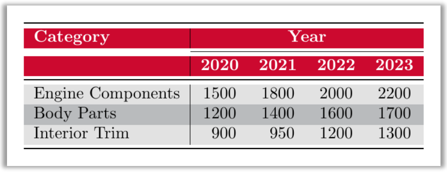
- Isolate affected systems from the network
- Disable compromised user accounts
- Freeze suspicious trading activities
- Activate Morgan Stanley's emergency trading halt protocol

3 Evidence Preservation

- Create forensic images of affected systems using EnCase
- Preserve firewall and IDS logs
- Capture volatile memory using Belkasoft RAM Capturer
- Secure CCTV footage of server rooms and trading floors

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

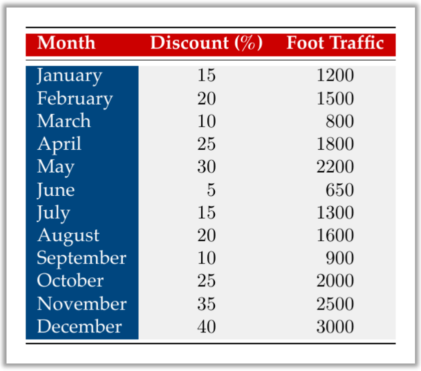
Figure 13: Randomly selected examples from our synthetic **document** data.



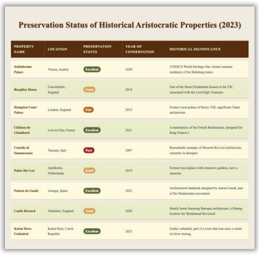
Category	Year			
	2020	2021	2022	2023
Engine Components	1500	1800	2000	2200
Body Parts	1200	1400	1600	1700
Interior Trim	900	950	1200	1300

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

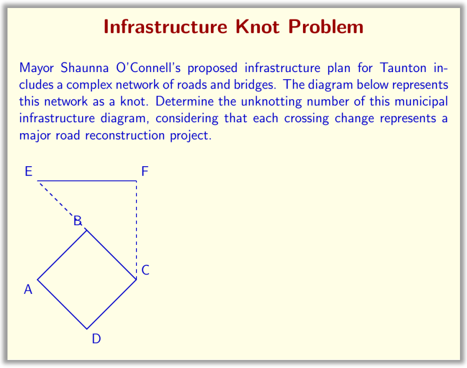


Preservation Status of Historical Aristocratic Properties (2023)

PROPERTY NAME	LOCATION	PRESERVATION STATUS	YEAR OF CONSERVATION	HISTORICAL SIGNIFICANCE
Schönbrunn Palace	Vienna, Austria	Excellent	2020	UNESCO World Heritage Site, former summer residence of the Habsburg rulers.
Burghley House	Lincolnshire, England	Good	2018	One of the finest Elizabethan houses in the UK, associated with the Lord High Treasurer.
Hampton Court Palace	London, England	Fair	2015	Former royal palace of Henry VIII, significant Tudor architecture.
Château de Chambord	Loir-et-Cher, France	Excellent	2021	A masterpiece of the French Renaissance, designed for King Francis I.
Castello di Sammezzano	Tuscany, Italy	Poor	2007	Remarkable example of Moorish Revival architecture, currently in disrepair.
Paleis Het Loo	Apeldoorn, Netherlands	Good	2019	Former royal palace with extensive gardens, now a museum.
Palacio de Gaudí	Astorga, Spain	Excellent	2022	Architectural landmark designed by Antoni Gaudí, part of the Modernisme movement.
Castle Howard	Yorkshire, England	Good	2020	Stately home featuring Baroque architecture, a filming location for 'Brideshead Revisited'.
Kutná Hora Cathedral	Kutná Hora, Czech Republic	Excellent	2023	Gothic cathedral, part of a town that was once a center of silver mining.

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

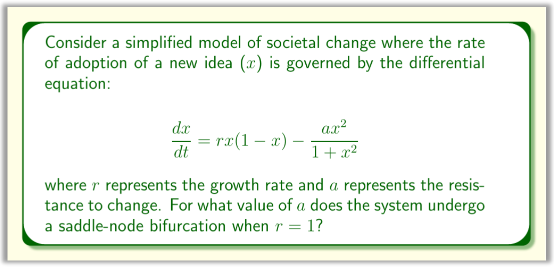
Figure 14: Randomly selected examples from our synthetic **table** data.



Infrastructure Knot Problem

Mayor Shaunna O'Connell's proposed infrastructure plan for Taunton includes a complex network of roads and bridges. The diagram below represents this network as a knot. Determine the unknotting number of this municipal infrastructure diagram, considering that each crossing change represents a major road reconstruction project.

Q: Can you answer this question? **A**: 2



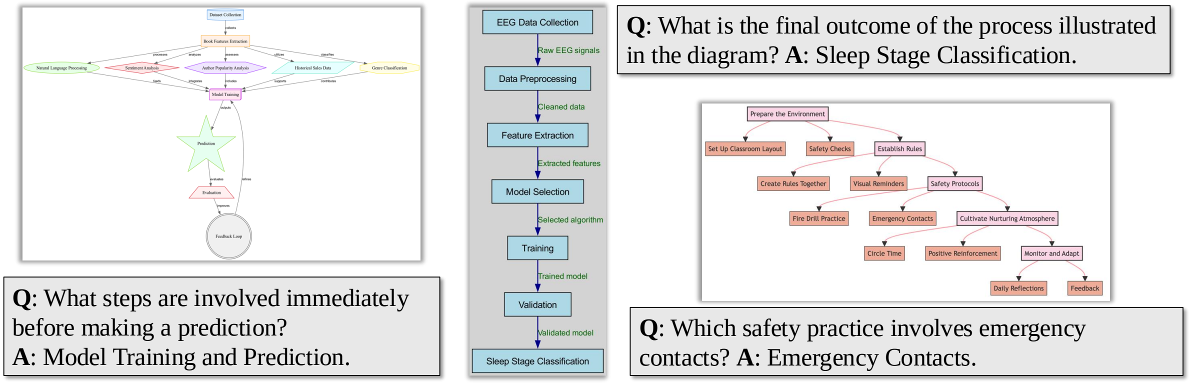
Consider a simplified model of societal change where the rate of adoption of a new idea (x) is governed by the differential equation:

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

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

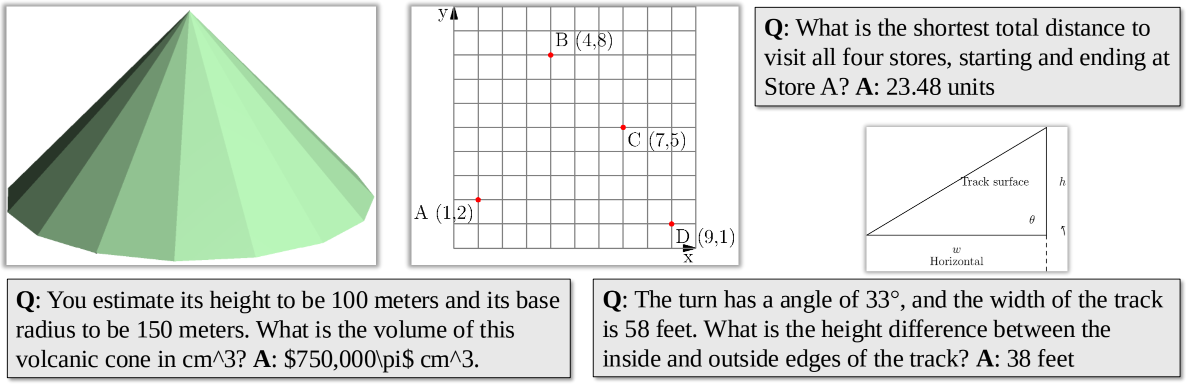


Q: What steps are involved immediately before making a prediction? **A**: Model Training and Prediction.

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.

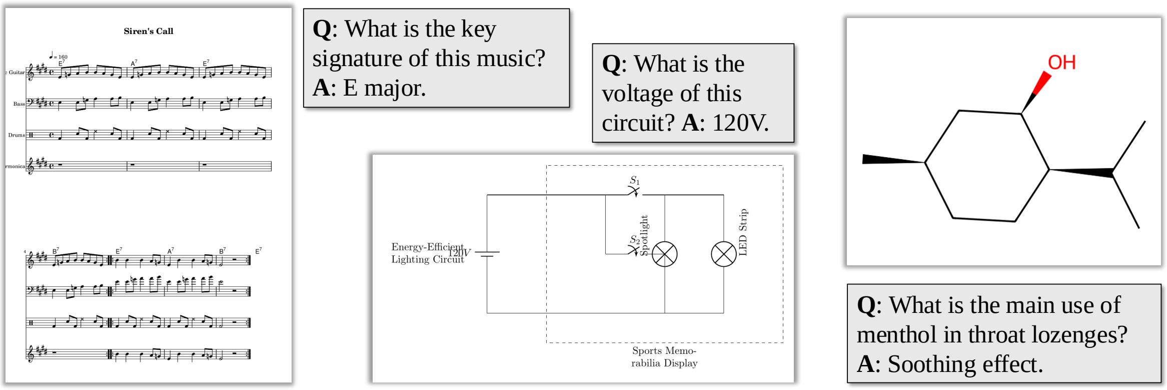


Q: What is the shortest total distance to visit all four stores, starting and ending at Store A? **A**: 23.48 units

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: 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.

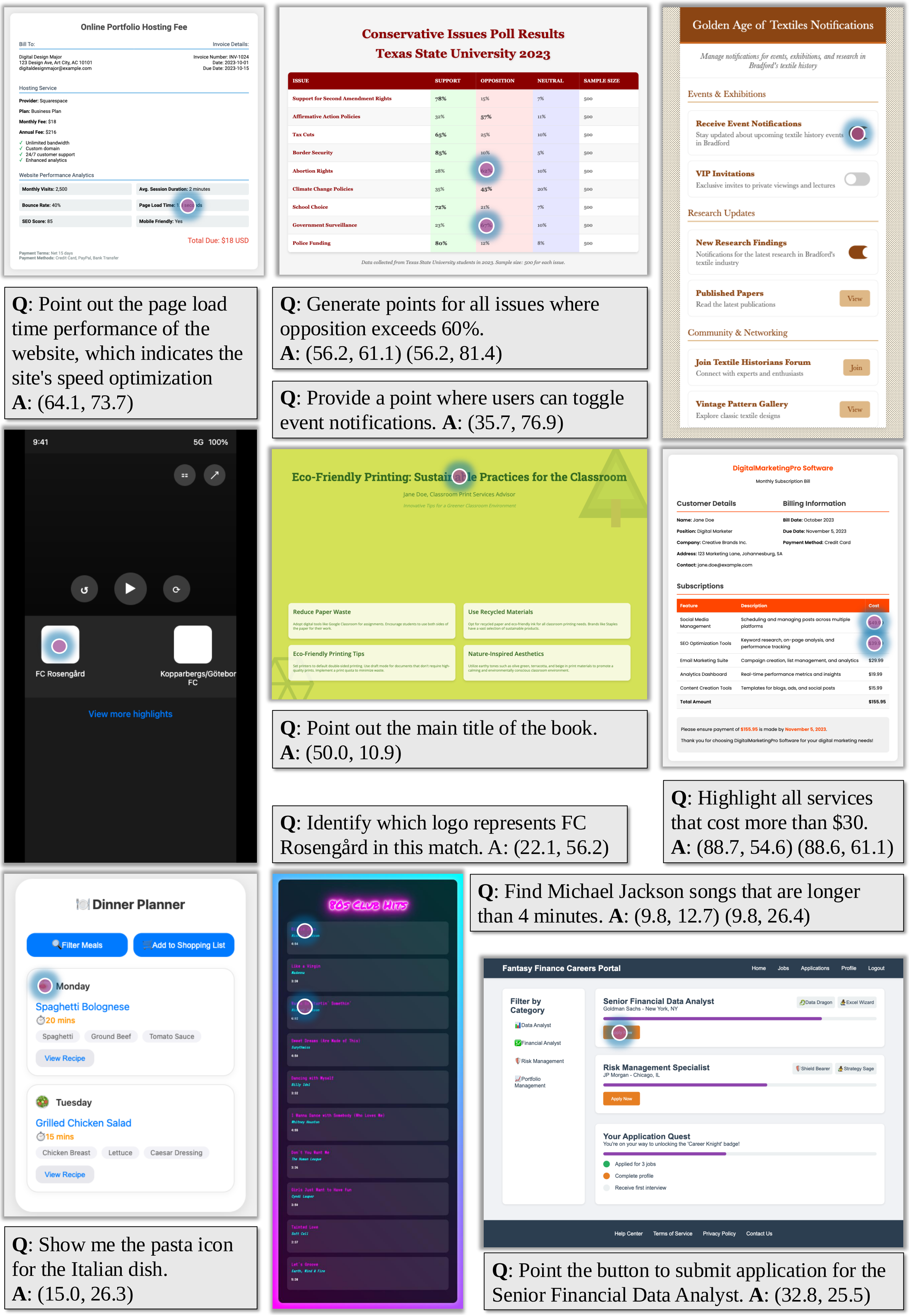


Q: What is the key signature of this music? **A**: E major.

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**.


Online Portfolio Hosting Fee
Conservative Issues Poll Results
Texas State University 2023
Golden Age of Textiles Notifications
Eco-Friendly Printing: Sustainable Practices for the Classroom
FC Rosengård
Kopparbergs/Göteborg FC
View more highlights
DigitalMarketingPro Software
Dinner Planner
80's Club Hits
Fantasy Finance Careers Portal

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

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

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

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

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

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.