Scaling Evidence-based Instructional Design Expertise through Large Language Models

Gautam Yadav[1]

[1]Carnegie Mellon University, 5000 Forbes Ave Pittsburgh PA 15213, United States

Abstract

This paper presents a comprehensive exploration of leveraging Large Language Models (LLMs), specifically GPT-4, in the field of instructional design. With a focus on scaling evidence-based instructional design expertise, our research aims to bridge the gap between theoretical educational studies and practical implementation. We discuss the benefits and limitations of AI-driven content generation, emphasizing the necessity of human oversight in ensuring the quality of educational materials. This work is elucidated through two detailed case studies where we applied GPT-4 in creating complex higher-order assessments and active learning components for different courses. From our experiences, we provide best practices for effectively using LLMs in instructional design tasks, such as utilizing templates, fine-tuning, handling unexpected output, implementing LLM chains, citing references, evaluating output, creating rubrics, grading, and generating distractors. We also share our vision of a future recommendation system, where a customized GPT-4 extracts instructional design principles from educational studies and creates personalized, evidence-supported strategies for users' unique educational contexts. Our research contributes to understanding and optimally harnessing the potential of AI-driven language models in enhancing educational outcomes.

Keywords

Large Language Models, Instructional Design, GPT-4, Evidence-Based Education, Personalized Learning

1. Introduction

The incorporation of large language models, such as GPT-4 [1], in learning engineering offers a range of benefits, including the generation of personalized content, augmentation of existing learning materials, and support in evaluation processes. Despite its potential, GPT-4's reliability can be inconsistent, particularly in complex subject areas, leading to potential inaccuracies and biases. To ensure high-quality learning experiences, a balanced approach combining AI-generated content with human oversight is essential.

Our primary aim is to bridge the gap between empirical educational research and its practical implementation, focusing on utilizing Large Language Models (LLMs) to streamline evidence-based instructional design. This goal is underscored by two comprehensive case studies that illustrate the potential of our approach.

In addition to presenting these in-depth examinations, we also explore future trajectories and limitations inherent in this area of research. By drawing these outlines, we aspire to foster a

AIEDLLM1: Empowering Education with LLMs - the Next-Gen Interface and Content Generation, July 7, 2023, Tokyo, Japan

gyadav@andrew.cmu.edu (G. Yadav)
0000-0002-9247-1920 (G. Yadav)

CEUR Workshop Proceedings (CEUR-WS.org)

deeper understanding of the judicious application of AI-driven language models such as GPT-4 in education. This understanding, in turn, can empower educators to optimize the use of these potent tools in their instructional endeavors.

2. Prior Work

The integration of AI-driven language models, like GPT-4, in education, presents numerous advantages, such as the capacity to produce tailored content, enhance existing learning materials, and offer support in evaluation processes. However, while GPT-4 can generate content that appears confident and precise, its reliability may be inconsistent, particularly in complicated subject areas. This can potentially result in incorrect or substandard content. Additionally, biases in AI, originating from the training data and human decision-making, could influence the generated content, resulting in inaccuracies. Expert knowledge in specific fields is crucial in validating and maintaining the quality of AI-generated educational content. Thus, although GPT-4 provides several benefits, a balanced approach that combines AI-generated content with human supervision remains vital to ensure high-quality learning experiences.

Previous research involving large language models has explored their application in educational settings, such as the use of models like GPT-4 for generating questions or providing hints/explanations to students [2, 3, 4]. However, the current literature, to the best of our knowledge, does not extend beyond the creation of single-step open-ended or selected-response questions. Various research studies highlight the effectiveness of active learning strategies such as Predict-Explain-Observe-Explain (PEOE) [5], faded worked examples [6], and self-explanation [7], given certain boundary conditions. Despite their proven efficiency, these assessments are not universally employed due to the time and expertise required to construct them and the challenge of making evidence-based decisions on the optimal strategy to use.

In our work, we explored the application of GPT-4, an AI-powered language model, in the development and assessment of educational content. This exploration has revealed valuable insights into the potential benefits and challenges associated with using GPT-4 to automate the creation of higher-order assessments. By sharing our findings, we aim to provide a well-rounded perspective on our suggestions for optimally harnessing AI technology in educational settings.

3. Case Studies

This section details two case studies drawn from my experience as a Learning Engineer at Carnegie Mellon University, where I focused on enhancing courses by incorporating active learning components for students.

3.1. Case Study 1: E-learning Design & Principles

The first case involved a course called E-learning Design & Principles. The instructor's objective was to address the 30 instructional principles outlined in [8], basing their approach on the following learning objective:

Learning Objective: Deliver nuanced and evidence-based guidance regarding the effectiveness of selected instructional principles, considering boundary conditions in a given context.

The selected assessment strategy was a scenario-based predict-observe-explain (POE) method for each instructional principle. However, creating a single case study, specifically for the Worked Example principle (refer Appendix), demanded several days and multiple iterations. Once an example was finalized, we employed one-shot prompting to scale it for other instructional principles. Here is a sample prompt for the spatial contiguity principle:

1. What are the boundary conditions of using spatial contiguity principle? cite references for these boundary conditions where authors have done a study to reach this conclusion based on data and evidence
2. Create assessments in form of predict-explain-observe-explain scenarios for spatial contiguity principle out of [feed previous prompt output here as references]
   I want to generate assessments in the form of predict-explain-observe-explain scenarios for explaining boundary conditions of when spatial contiguity principle is applicable based on EVIDENCE IN RESEARCH.
   For every multiple-choice question and short answer, we want to generate feedback.
   Can you start by giving a detailed study description followed by PEOE exercises for each of the references generated above and summary in the end of how these features interact with each other to make decision?
   Let me write you an example of PEOE scenarios for boundary conditions using cited research studies in Worked Example principle then you try writing it for spatial contiguity principle: [feed Worked Example principle case study here as an example from as described in Appnedix]

After many iterations, we finalized these prompts, for example, one of the iteration involved emphasizing evidence from research to prevent the generation of hypothetical scenarios over constructing scenarios from studies in the cited papers. As shown in Figure 1, the outputs for each principle still required iterative cycles with the subject matter expert (instructor). Still, the time required for subsequent principles was cut by over 70% as we automated the process of finding relevant references and creating an initial draft.

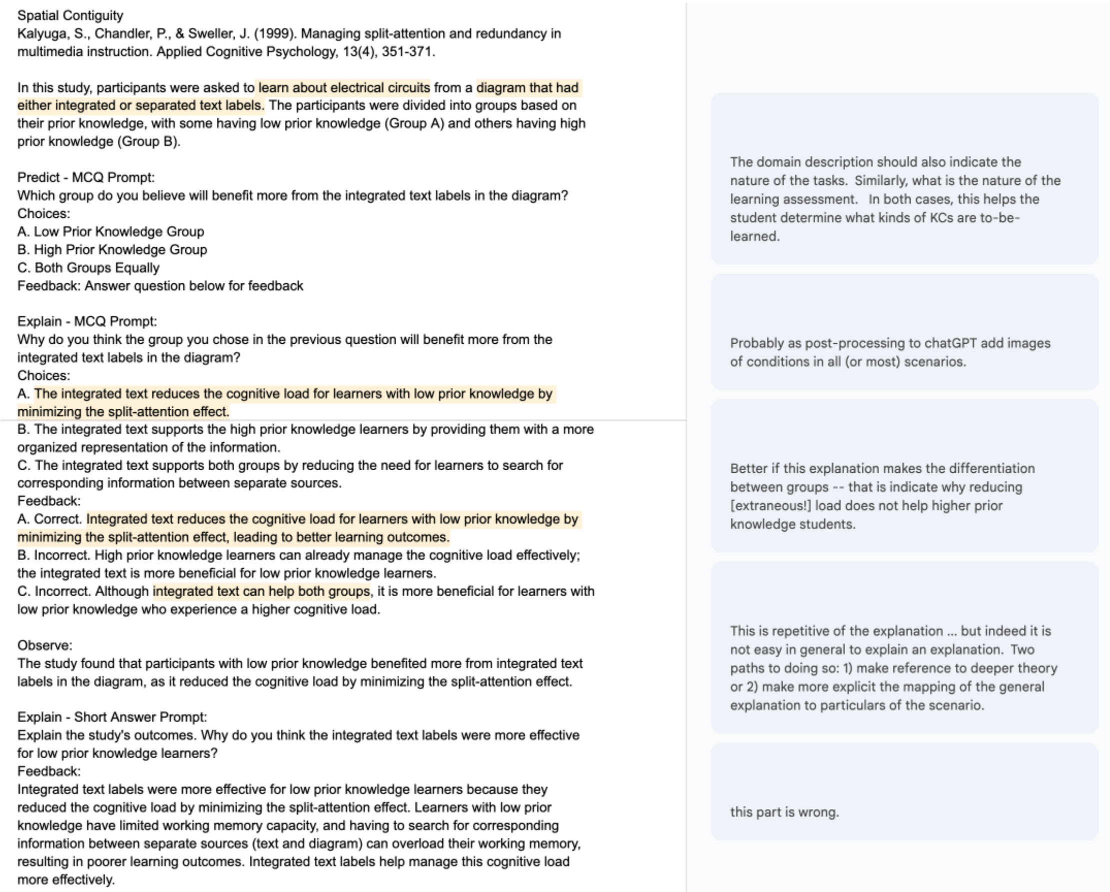
3.2. Case Study 2: Learning Analytics and Educational Data Science

The second case study pertains to a new course titled "Learning Analytics and Educational Data Science," slated for Fall 2023. There were no pre-existing online components, and the instructor wanted to develop programming 'learn-by-doing' assignments using Jupyter Notebook.

Learning Objective: Implement a predictive model using Python

Our chosen assessment strategy was the use of faded worked examples with feedback. We attempted to leverage data visualization problems developed using a combination of worked examples and problem-solving practice activities with feedback for another CMU course using GPT-4. However, these wprled examples were inappropriate in this context, as students could simply copy-paste solutions so we only focused on crafting problem-solving activities.

We iteratively crafted a series of prompts, designed to yield the most effective output through trial and error:


Spatial Contiguity
Kalyuga, S., Chandler, P., & Sweller, J. (1999). Managing split-attention and redundancy in multimedia instruction. Applied Cognitive Psychology, 13(4), 351-371.

In this study, participants were asked to learn about electrical circuits from a diagram that had either integrated or separated text labels. The participants were divided into groups based on their prior knowledge, with some having low prior knowledge (Group A) and others having high prior knowledge (Group B).

Predict - MCQ Prompt:
Which group do you believe will benefit more from the integrated text labels in the diagram?
Choices:
A. Low Prior Knowledge Group
B. High Prior Knowledge Group
C. Both Groups Equally
Feedback: Answer question below for feedback

Explain - MCQ Prompt:
Why do you think the group you chose in the previous question will benefit more from the integrated text labels in the diagram?
Choices:
A. The integrated text reduces the cognitive load for learners with low prior knowledge by minimizing the split-attention effect.
B. The integrated text supports the high prior knowledge learners by providing them with a more organized representation of the information.
C. The integrated text supports both groups by reducing the need for learners to search for corresponding information between separate sources.
Feedback:
A. Correct. Integrated text reduces the cognitive load for learners with low prior knowledge by minimizing the split-attention effect, leading to better learning outcomes.
B. Incorrect. High prior knowledge learners can already manage the cognitive load effectively; the integrated text is more beneficial for low prior knowledge learners.
C. Incorrect. Although integrated text can help both groups, it is more beneficial for learners with low prior knowledge who experience a higher cognitive load.

Observe:
The study found that participants with low prior knowledge benefited more from integrated text labels in the diagram, as it reduced the cognitive load by minimizing the split-attention effect.

Explain - Short Answer Prompt:
Explain the study's outcomes. Why do you think the integrated text labels were more effective for low prior knowledge learners?
Feedback:
Integrated text labels were more effective for low prior knowledge learners because they reduced the cognitive load by minimizing the split-attention effect. Learners with low prior knowledge have limited working memory capacity, and having to search for corresponding information between separate sources (text and diagram) can overload their working memory, resulting in poorer learning outcomes. Integrated text labels help manage this cognitive load more effectively.

Figure 1: Example of Predict-Explain-Observe-Explain activity created by GPT-4, focusing on the Spatial Contiguity Principle for one cited reference. A subject matter expert has added annotations to evaluate the quality of the content. The figure highlights both the strengths and limitations of using GPT-4 generated content in this context.

1. Can you give me 2 examples of hands-on exercises that cover the following learning objective "Implement a predictive model using Python" in a module called Classifiers.
2. Can you provide a worked example in Python, including the corresponding code for the following hands-on exercise: [feed one example from the previous prompt output here]
3. (3 - 5 Prompts): [Debug any errors that appear when trying to run code provided in Prompt 2 output in Google Colab.; This average of 3 - 5 is based on the development of three exercises using different datasets for the learning objective above]
4. For each step in Jupyter Notebook,
   - I want to create practice activities like these:
     [examples provided in the appendix]
     Convert following code into above format where [code for this step] where students need to enter the given code with test cases to verify if students entered correctly.
   - (Only if a test case inadvertently revealed the answer in a previous step, we asked

Create a binary 'at-risk' column

```python
[ ] # Replace this with your solution
    student_info['at-risk'] = None

    # The code below is to provide you with feedback, please do not modify.
    try:
        # Generate random indices
        np.random.seed(0)  # This ensures the random indices are the same every time
        random_indices = np.random.choice(student_info.shape[0], size=5, replace=False)

        # Check if 'at-risk' column is correctly computed for the random indices
        for idx in random_indices:
            expected_value = 1 if student_info.loc[idx, 'final_result'] in ['Fail', 'Withdrawn'] else 0
            if student_info.loc[idx, 'at-risk'] == expected_value:
                print_passed(f"The 'at-risk' value for index {idx} is correctly computed.")
            else:
                print_failed(f"The 'at-risk' value for index {idx} is not correctly computed. It should be set to 1 if 'final_result' is 'Fai
                raise AssertionError

    except:
        print_failed("Your solution did not compute the 'at-risk' column correctly. You should use the apply() function to compute the 'at-ri
        raise
```

Preprocess the data: Count the number of interactions each student has had with the VLE

```python
# Replace this with your solution
vle_interactions = None

# The code below is to provide you with feedback, please do not modify.
try:
    # Generate random indices
    np.random.seed(0)  # This ensures the random indices are the same every time
    random_indices = np.random.choice(student_vle.shape[0], size=5, replace=False)

    # Check if vle_interactions is correctly computed for the random indices
    for idx in random_indices:
        expected_value = student_vle.loc[student_vle['id_student'] == student_vle.loc[idx, 'id_student'], 'id_student'].count()
        if vle_interactions.loc[vle_interactions['id_student'] == student_vle.loc[idx, 'id_student'], 'id_student'].values[0] == expected_
            print_passed(f"The number of interactions for student id {student_vle.loc[idx, 'id_student']} is correctly computed.")
        else:
            print_failed(f"The number of interactions for student id {student_vle.loc[idx, 'id_student']} is not correctly computed. It sl
            raise AssertionError
```

Figure 2: A segment of a Jupyter Notebook showcasing a sequence of practice activities that were designed with the aid of GPT-4. The objective of these activities is to help students learn how to identify at-risk students using predictive models in Python.

> for more complex combinations to check for correct usage without giving away the answer) can you use more complex combinations to check for correct usage of above step without giving away the answer if students actually read the test cases

We first attempted to use the same one-shot prompting strategy as in the previous case study but found that few-shot prompting yielded better results. Interestingly, GPT-4 did not generalize the Altair library in the output Python code as shown in Figure 2, even though all examples as shown in Appendix consisted of that. We only edited a few instructions where steps like reading the datasets or training classifiers were not suitable for this format and were covered in the student's prior knowledge.

4. Prompt Engineering - Best Practices for Instructional Design

Drawing from our experiences with GPT-4 in educational content creation, we have garnered invaluable insights into the potential advantages and obstacles of integrating AI in education. The lessons we've learned and their implications can guide educators and instructional designers to successfully implement AI-driven language models such as GPT-4, maximizing benefits while mitigating potential challenges. Here are our proposed best practices when utilizing large language models:

4.1. Utilizing Templates for Instructional Design Tasks

Prior research [9] indicates that as LLMs become more powerful, employing several examples (few-shot prompting) might not be as effective as zero examples (zero-shot prompting). Our case studies demonstrated the usefulness of examples in some prompts and zero-shot prompting in others. For complex tasks, like assessments involving specific instructional design principles, using well-defined examples in the form of templates can improve the quality of the generated output. Templates help enhance the structure and consistency of the materials generated by AI-driven language models, promoting a more streamlined content creation process.

4.2. Fine-Tuning for Novel Instructional Problems

When creating new problems similar to the input's problem structure, a lower temperature value in the prompt can maintain a focus on the same knowledge components. Conversely, for diversity and problems in various contexts, such as a story or equation variable problems, a higher temperature can facilitate the broader transfer [10].

4.3. Handling Unexpected Output

At times, LLMs may not behave as expected, even if all examples follow the same pattern. To counter this, consider a lower temperature for output similar to input or defining explicit rules in the prompt. Defining explicit rules needed involves iterating your prompt, identifying the patterns it struggles with, and explicitly stating these as rules in your prompt.

4.4. Implementing LLM Chains for Multi-step Instructional Tasks

Although the surface quality of AI-generated educational content may be tempting, it's often best to break the task into the smallest subtasks initially. Once you have established the output quality, you can consider combining them, provided the interaction between them doesn't compromise the output. This approach, known as LLM chaining, helps improve output quality by avoiding the pitfalls of asking LLMs to handle multiple or nested tasks in one prompt [11].

4.5. Citing References in AI-Generated Instructional Materials

Including credible sources or citations in AI-generated content enhances its accuracy and credibility. However, caution is advised as LLMs can occasionally fabricate references.

4.6. Evaluating Output with AI Critique and SMEs

AI critique can augment content quality as shown in Figure 3. However, content should be verified with Subject Matter Experts (SMEs) before incorporating it. Despite LLM's ability to critique and improve its own output, users should critically evaluate generated content.

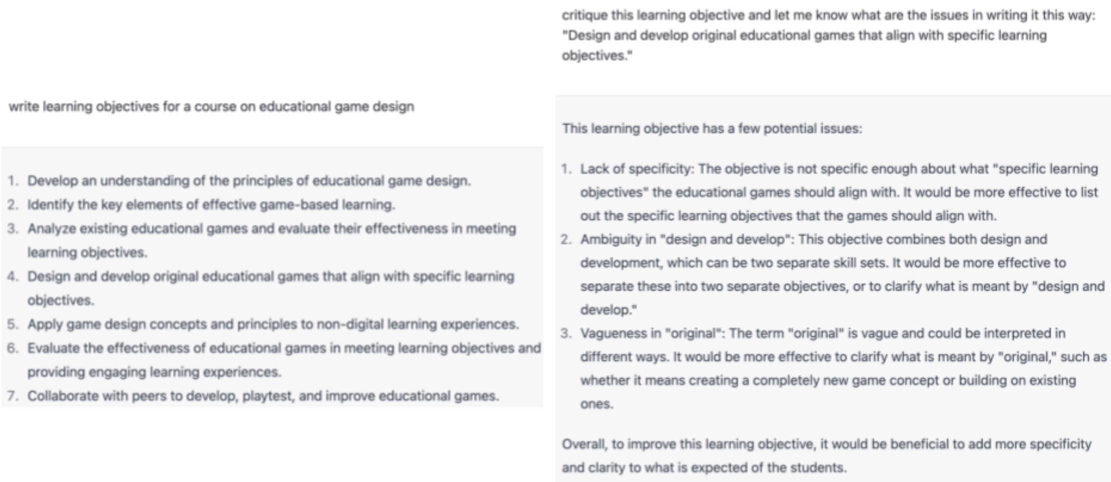


Figure 3: ChatGPT tasked to generate learning objectives for an educational game design, followed by critiquing one of them. The critique provided by ChatGPT gives meaningful recommendations for improvement, thereby demonstrating the model's self-awareness and its capacity to evaluate its own output.

4.7. Creating Rubrics and Grading

AI can expedite rubric creation and grading, but educator involvement is crucial for reliability and fairness. Combined with a human educator, GPT-4 can enhance grading consistency and remove bias.

4.8. Generating Distractors

Interestingly, we have noticed GPT-3.5 outperforms GPT-4 in creating distractors. Asking GPT-3.5 for five characteristics or applications related to a problem can help identify potential distractors, typically arising from incorrectly generated solutions of better quality compared to explicitly asking to create distractors in GPT-4.

5. Future Work

In our future work, we aim to develop a sophisticated recommendation system, essentially 'closing the loop' in our educational technology solution.

At the heart of our proposed system is a customized version of GPT-4, which we use to extract crucial information from empirical educational studies such as each paper's instructional design principles and identify the conditions under which they thrive. This extraction process enables us to encode and store data such as the educational domain, cognitive load, and learners' prior knowledge into a dedicated database, primed for future retrieval and application.

Harnessing the capabilities of GPT-4, we then create multiple assessment applications for each instructional design principle archived in our database. We adopt a few-shot prompting approach to devise these examples, aiming to guide users in effectively applying these principles across a broad range of educational contexts.

Our recommendation system is designed with the user's specific needs in mind. Users can input their unique instructional design requirements, including target learners, learning objectives, subject area, and other pertinent conditions. Our GPT-4 based system uses this information to generate evidence-supported instructional design strategies, tailored to the user's specific context. Each recommended strategy is paired with example applications and supported by original references from the studies they were drawn from, enabling users to further verify and delve into the source material.

The overarching goal of this endeavor is to democratize instructional design expertise, making it widely accessible to instructional designers and teachers alike. By doing so, we aim to streamline the design process, enhance educational outcomes, and ultimately drive forward the future of educational technology.

Acknowledgments

I extend my sincere gratitude to Prof. Ken Koedinger and John Stamper, whose subject matter expert guidance was indispensable to the success of this research. Their profound wisdom and unwavering support enriched this work immeasurably. I also want to acknowledge students from Human Learning and How to Optimize It for their contribution to crafting the Model Human Learner document which inspired this idea.

References

[1] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.

[2] S. Elkins, E. Kochmar, J. C. Cheung, I. Serban, How useful are educational questions generated by large language models?, arXiv preprint arXiv:2304.06638 (2023).

[3] E. Prihar, M. Lee, M. Hopman, A. Kalai, S. Vempala, A. Wang, G. Wickline, N. Heffernan, Comparing different approaches to generating mathematics explanations using large language models, in: Proceedings of the AIED2023 Conference, 2023. To be published.

[4] Z. A. Pardos, S. Bhandari, Learning gain differences between chatgpt and human tutor generated algebra hints, arXiv preprint arXiv:2302.06871 (2023).

[5] D. R. Chine, P. Chhabra, A. Adeniran, J. Kopko, C. Tipper, S. Gupta, K. R. Koedinger, Scenario-based training and on-the-job support for equitable mentoring, in: Innovative Approaches to Technology-Enhanced Learning for the Workplace and Higher Education: Proceedings of 'The Learning Ideas Conference'2022, Springer, 2022, pp. 581–592.

[6] R. J. Salden, V. Aleven, R. Schwonke, A. Renkl, The expertise reversal effect and worked examples in tutored problem solving, Instructional Science 38 (2010) 289–307.

[7] T. Nagashima, A. N. Bartel, G. Yadav, S. Tseng, N. A. Vest, E. M. Silla, M. W. Alibali, V. Aleven, Using anticipatory diagrammatic self-explanation to support learning and performance in early algebra., Grantee Submission (2021).

[8] K. R. Koedinger, J. L. Booth, D. Klahr, Instructional complexity and the science to constrain it, Science 342 (2013) 935–937.

[9] L. Reynolds, K. McDonell, Prompt programming for large language models: Beyond the few-shot paradigm, in: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–7.
[10] G. LLC, Getting started with the vertex ai palm api & python sdk, 2023.
[11] T. Wu, M. Terry, C. J. Cai, Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts, in: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022, pp. 1–22.

A. Worked Examples vs. Practice: A Content/Knowledge Demand Boundary Condition

There were 4 research studies created for worked example principle to help students construct their own understanding of are boundary conditions of when to use worked examples instead of problem-solving along with the summary, here we present the first research paper only for reference on the format used for case study 1-

Kalyuga et al. (2001) Experiment 1 - Novices - Prediction: Consider a study where participants were asked to write simple programmable logic controller (PLC) programs, and all participants had not been exposed to any training materials on relay circuits or PLC programming prior to the study. In one group (group A), participants alternated between 12 worked examples and 12 practice problems, whereas the other group (group B) received 24 practice problems.

Experiment 1 Description: The procedure for the worked examples condition was identical to the problem-solving condition except that it included examples of relay circuits with corresponding steps in programming these circuits. Participants were requested to mentally follow all the steps according to a numbered sequence to ensure the programs were correct. The worked examples condition contained a series of 12 examples of relay circuits and programs for those circuits. Each example was followed by a problem-solving task. The 24 circuits used in this condition were identical to the circuits used in the problem-solving condition. The circuits with even numbers were problem-solving tasks identical for both treatments, while the circuits with uneven numbers were presented as worked examples in the worked example condition. Thus, in the worked example condition, participants studied 12 examples and attempted 12 problems, but in the problem-solving condition, participants attempted 24 problems. To avoid a split attention effect, separate program lines were embedded into the circuit with numbers of elements as closely as possible to the corresponding elements in the circuit. Participants could study each example as long as they wished.

Experiment 1 Outcomes: For these more novice students, worked examples produce greater instructional efficiency, that is, higher post-test question scores in less instruction time.

B. Data Visualization Problem-Solving Activities

Figures 4 and 5 show examples of data visualization problems that were used for few-shot prompting in case study 2.

Question Type	MCQ
Prompt	Which group do you believe will exhibit better learning outcomes?
Choices	1. Worked Examples Group 2. Problem-Solving Group 3. No Difference
Feedback	Answer question below for feedback

```
solution = None

# The code below is to provide you with feedback, please do not modify.
try:
   # Display your chart
   display(solution)

   # Check if the type of mark used in your chart is correct
   if check_altair_chart_object_attribute('solution.mark', 'point') or
check_altair_chart_object_attribute('solution.mark.type', 'point'):
       print_passed("The type of mark used in your chart is correct.")
   else:
       print_failed("The type of mark used in your chart is incorrect. You should use
mark_point() to create a scatter plot.")

   # Check if the encoding channels used in your chart are correct
   if check_altair_chart_object_attribute('solution.encoding.x.field', 'Body Mass (g)'):
       print_passed("The x encoding channel used in your chart is correct.")
   else:
       print_failed("The x encoding channel used in your chart is incorrect. The x-axis should be
encoded with 'Body Mass (g)'.")
except:
   print_failed("Your solution did not generate the expected chart. You should use mark_point()
to create a scatter plot. The x-axis should be encoded with 'Body Mass (g)'.")
   raise
```

Figure 4: Example 1: Creating Scatter Plots in Altair

Question Type	MCQ
Prompt	Why do you think the group you chose in the previous question will exhibit better learning outcomes?
Choices	1. This group had less experienced learners who benefited from the reduced cognitive load of a worked example (Worked Example Effect). 2. This group had more experienced learners who benefited from the practice opportunity without the cognitive overhead of seeing a worked example (Redundancy Effect). 3. There should be no difference in the groups because each group experienced 24 learning opportunities.
Feedback	1. Correct. In their first experiment, the Worked Example Effect was found with inexperienced learners but disappeared after training (where more experienced students were exposed to training materials for a long enough time). See results for instructional efficiency below. 2. Incorrect. In their first experiment, the Worked Example Effect was found with inexperienced learners but disappeared after training (where more experienced students were exposed to training materials for a long enough time). See results for instructional efficiency below. 3. Incorrect. In their first experiment, the Worked Example Effect was found with inexperienced learners but disappeared after training (where more experienced students were exposed to training materials for a long enough time). See results for instructional efficiency below.

Question Type	Short Answer
Prompt	Explain experiment 1 outcomes. Why do you think that the worked examples were more effective for this group?
Feedback	According to Kalyuga et al. (2001): "For inexperienced learners, problem solving-based learning might overload the limited capacity of working memory, resulting in poor learning outcomes compared to a worked examples-based approach."

```python
solution = None

# The code below is to provide you with feedback, please do not modify.
try:
   # Display your chart
   display(solution)

   # Check if the type of mark used in your chart is correct
   if check_altair_chart_object_attribute('solution.mark', 'point') or
check_altair_chart_object_attribute('solution.mark.type', 'point'):
       print_passed("The type of mark used in your chart is correct.")
   else:
       print_failed("The type of mark used in your chart is incorrect. You should use
mark_point() to create a scatter plot.")

   # Check if the encoding channels used in your chart are correct
   if check_altair_chart_object_attribute('solution.encoding.x.field', 'Beak Length (mm)') and
check_altair_chart_object_attribute('solution.encoding.y.field', 'Beak Depth (mm)'):
       print_passed("The encoding channels used in your chart are correct.")
   else:
       print_failed("The encoding channels used in your chart are incorrect. The x-axis should be
encoded with 'Beak Length (mm)' and the y-axis should be encoded with 'Beak Depth (mm)'.")

   # Check if the axis scale zero is set to False
   if check_altair_chart_object_attribute('solution.encoding.x.scale.zero', False) and
check_altair_chart_object_attribute('solution.encoding.y.scale.zero', False):
       print_passed("The axis scale zero is set to False.")
   else:
       print_failed("The axis scale zero is not set to False. Both the x-axis scale zero and
y-axis scale zero should be set to False to prevent misleading visualizations.")

   # Check if the tooltip is included in the chart
   if check_altair_chart_object_attribute('solution.encoding.tooltip.field', 'Island'):
       print_passed("The tooltip is included in the chart.")
   else:
       print_failed("The tooltip is not included in the chart. The tooltip should be encoded with
'Island' to provide additional information.")

except:
   print_failed("Your solution did not generate the expected chart. You should use mark_point()
to create a scatter plot. The x-axis should be encoded with 'Beak Length (mm)' and the y-axis
should be encoded with 'Beak Depth (mm)'. Both the x-axis scale zero and y-axis scale zero should
be set to False to prevent misleading visualizations. The tooltip should be encoded with 'Island'
to provide additional information.")
   raise
```

Figure 5: Example 2: Adding a Tooltip in Altair