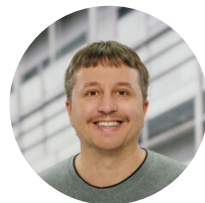


Topic 3. Multimodal Content Generation and Understanding

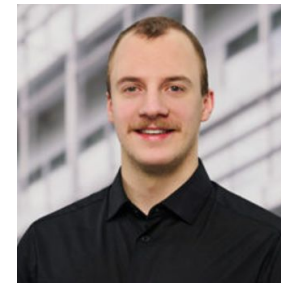


Steffen Eger

Professor
University of Technology
Nuremberg
steffen.eger@utn.de

Agenda

- ❑ Scientific Figure Understanding
- ❑ Scientific Figure Generation
 - ❑ Code Generation
 - ❑ Text-to-Code (TikZ, Python, Others)
 - ❑ Image-to-Code
 - ❑ Direct Image Generation
- ❑ Scientific Slide and Poster Generation



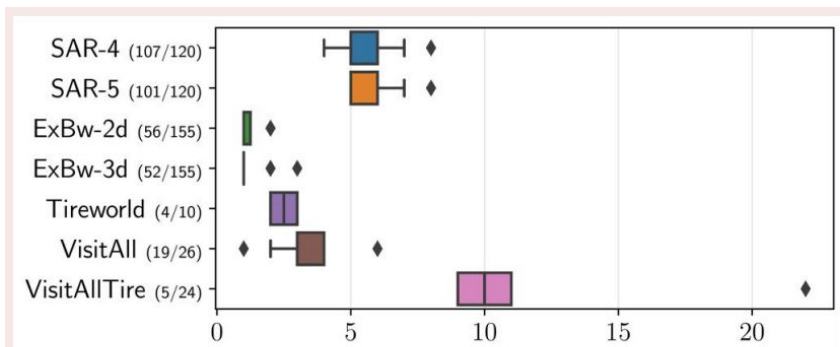
Agenda

- ❑ Scientific Figure Understanding
- ❑ Scientific Figure Generation
- ❑ Scientific Slide and Poster Generation

Scientific Figure Understanding

Scientific Figure Understanding

Mostly QA over charts (e.g., FigureQA, DVQA, PlotQA, ChartQA, CharXiv, SPIQA, ...)



Question: Is ExBw-2d greater than Tireworld?

Answer: No

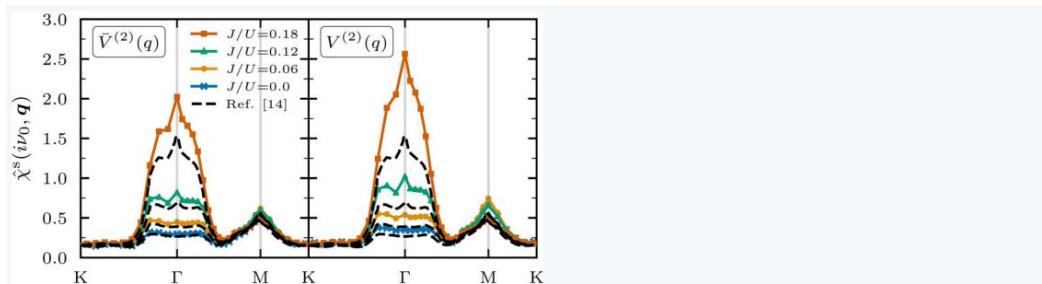
Scientific Figure Understanding

CharXiv (NeurIPS 2024) [1]:

- ❑ Unbounded chart types
 - ❑ e.g., not restricted to Scatter, Line, Bar, Pie
- ❑ Descriptive questions: Examining basic chart elements
 - ❑ e.g., axis, title, ...
- ❑ Reasoning questions: Synthesizes information
 - ❑ e.g., maximum number of consecutive datapoints which forms a decreasing sequence, ...

Scientific Figure Understanding

Descriptive:



Question: For the subplot at row 1 and column 2, how many lines are there?

- * Your final answer should be the number of lines in the plot. Ignore grid lines, tick marks, and any vertical or horizontal auxiliary lines.
- * If the plot does not contain any lines or is not considered a line plot, answer "Not Applicable".

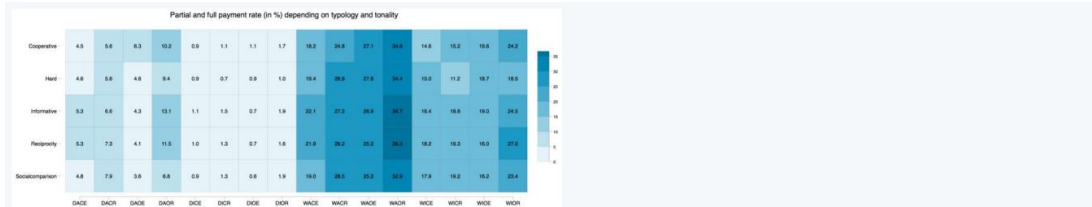
Answer: 8

GPT-4o: The subplot at row 1 and column 2 contains 5 lines.

Claude 3 Sonnet: For the subplot in the second column, there are 5 lines plotted, corresponding to different values of $J/U = 0.18, 0.12, 0.06, 0.0$, and the reference line labeled "Ref. [14]".

Scientific Figure Understanding

Reasoning:



Question: Adding up all numbers in each individual column, which column achieves the smallest total value?

- * Your final answer must be grounded to some text that is explicitly written and relevant to the question in the chart.
- * If you need to answer multiple terms, separate them with commas.
- * Unless specified in the question (such as answering with a letter), you are required to answer the full names of subplots and/or labels by default.

Answer: DIOE

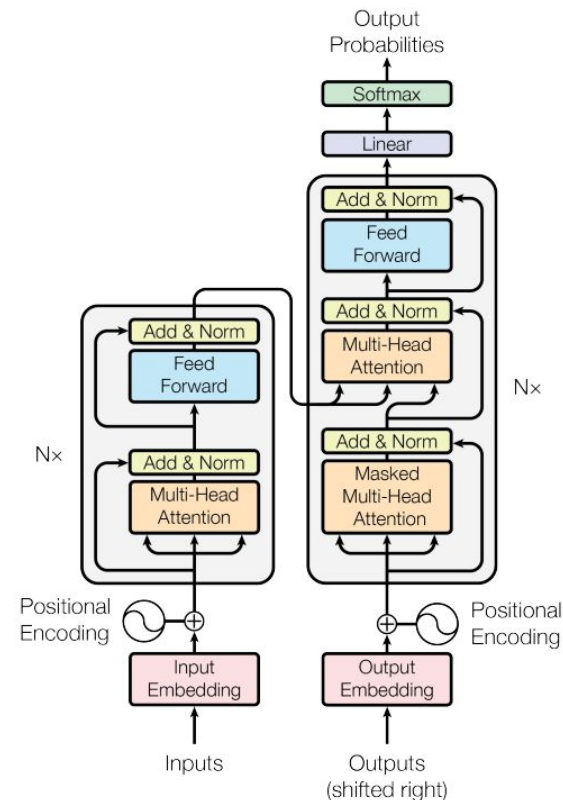
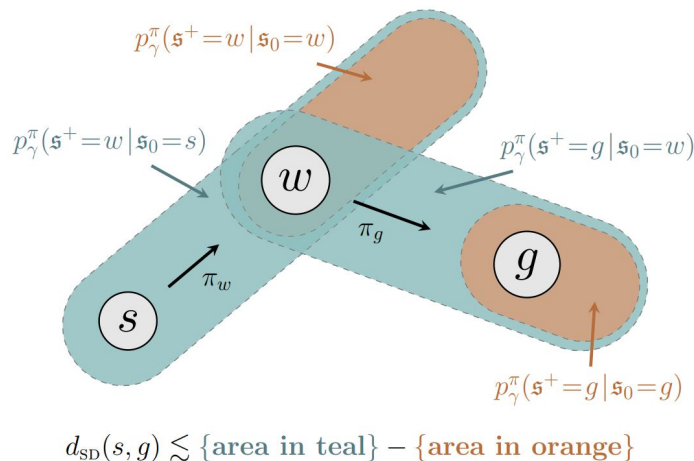
GPT-4o: The column with the smallest total value is "DICE."

Claude 3 Sonnet: The column that achieves the smallest total value when adding up all the numbers is **DICR**, which corresponds to the "Informative, Contradict, Reject" condition in the chart.

Scientific Figure Generation

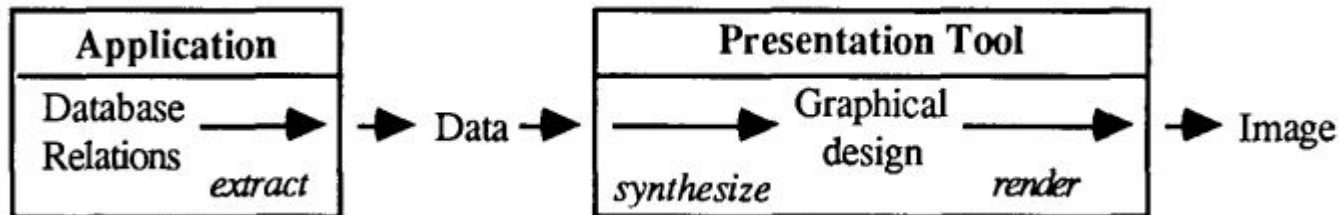
Motivation

- ❑ Good scientific figures can be impactful → Increase chances of acceptance and enhances citation counts
- ❑ Difficult for humans to generate high-quality scientific figures

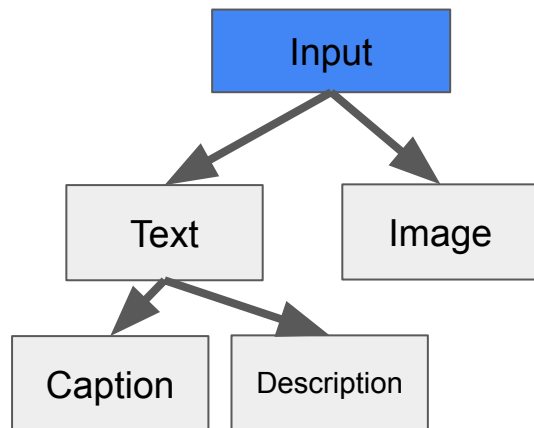


History

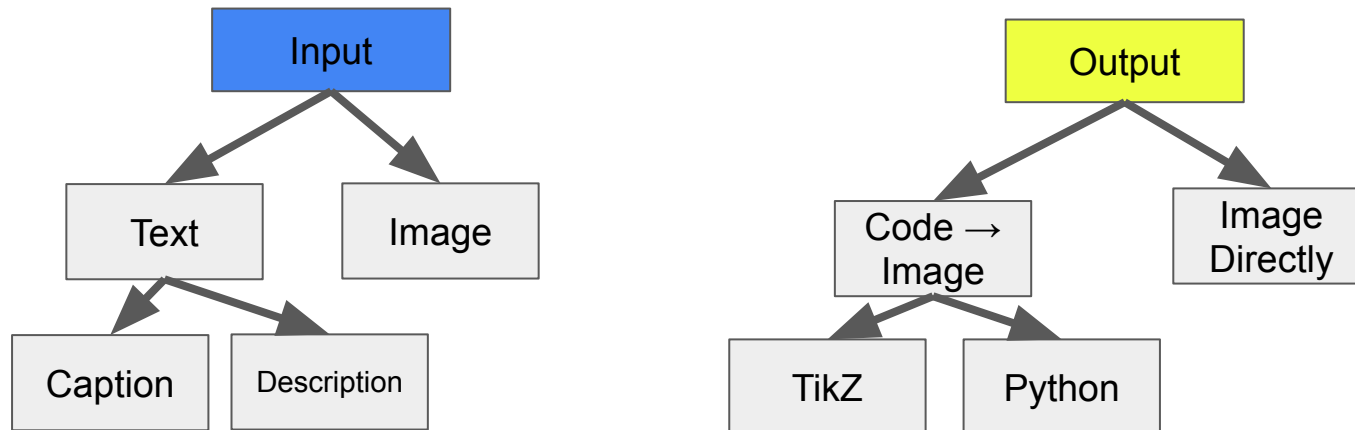
- Early work dates back to the 1980s and 1990s
 - e.g. APT by Jock Mackinlay (1986)
- Motivation:
 - help users make figures faster and better
 - empower users → foster inclusivity



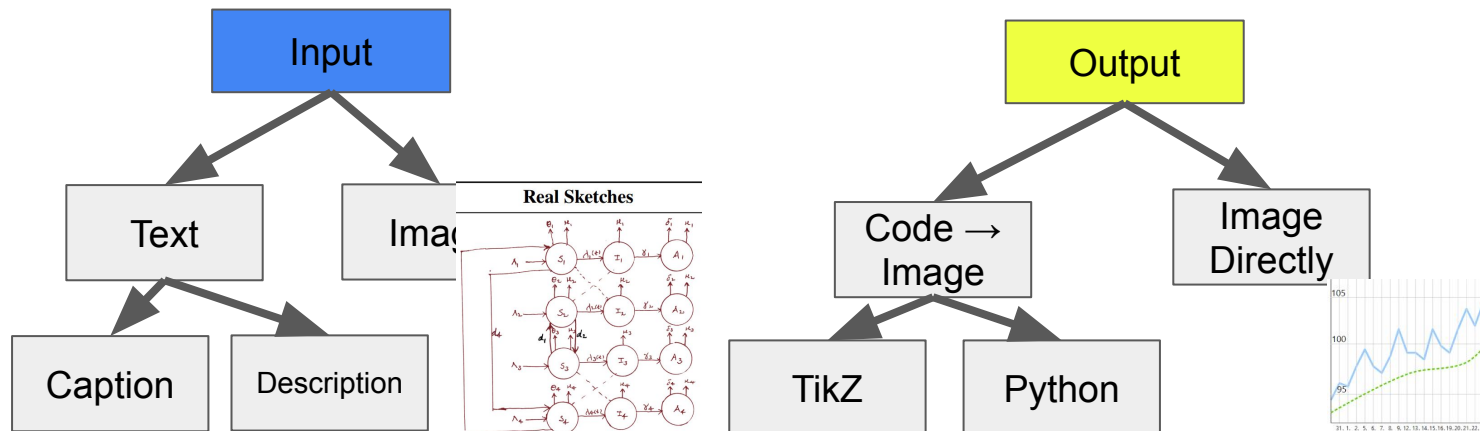
Schematic Overview



Schematic Overview



Schematic Overview



“Graphical representation of the network”

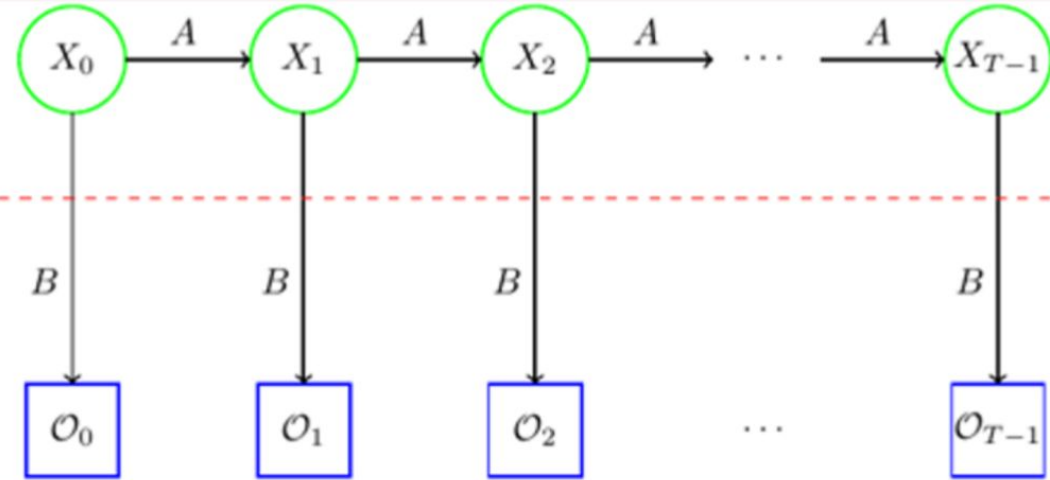
“The image shows a graph with three vertices 2 colored blue, 1 colored red”

```
\begin{tikzpicture}
\draw[style=dashed]
(2,.5) circle (0.5); ...
```

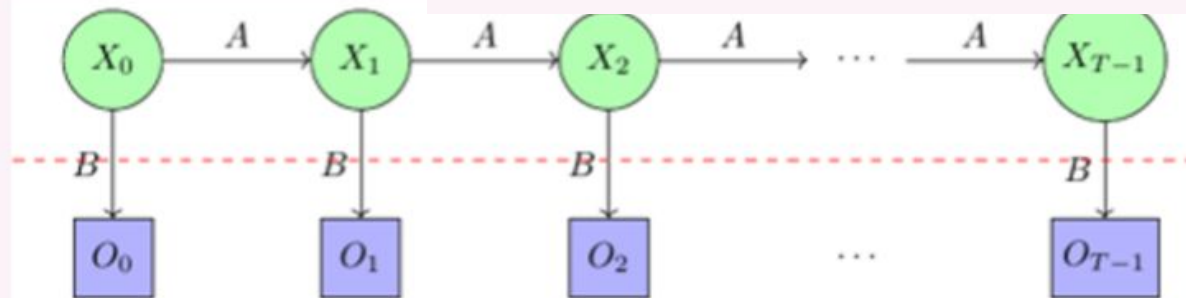
```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```

A concrete illustration

A sequence of green circles is arranged horizontally from left to right, labeled $X_0, X_1, X_2, \dots, X_{T-1}$. Below each circle, there is a corresponding blue square, labeled $O_0, O_1, O_2, \dots, O_{T-1}$. Each circle X_i is connected to its corresponding square O_i by a vertical arrow labeled B . The circles are also connected to each other by horizontal arrows labeled A . The sequence continues.



```
\documentclass[tikz]{standalone}
\usepackage{tikz}
\usetikzlibrary{arrows}
\begin{tikzpicture}
\begin{scope}
\node[green,circle] (X0) at (0,0) {$X_0$};
\node[green,circle] (X1) at (1,0) {$X_1$};
\node[green,circle] (X2) at (2,0) {$X_2$};
\node[green,circle] (XT) at (T,0) {$X_{T-1}$};
\end{scope}
\begin{scope}
\node[blue,square] (O0) at (0,-1) {$O_0$};
\node[blue,square] (O1) at (1,-1) {$O_1$};
\node[blue,square] (O2) at (2,-1) {$O_2$};
\node[blue,square] (OT) at (T,-1) {$O_{T-1}$};
\end{scope}
\draw[red,dashed] (0,-0.5) -- (T,-0.5);
\draw[black] (X0) -->[A] (X1);
\draw[black] (X1) -->[A] (X2);
\draw[black] (X2) -->[A] (XT);
\draw[black] (X0) -->[B] (O0);
\draw[black] (X1) -->[B] (O1);
\draw[black] (X2) -->[B] (O2);
\draw[black] (XT) -->[B] (OT);
\end{tikzpicture}
```



```
{_0$};
$X_1$};
_2$};
```

A concrete illustration

A sequence of green circles labeled X_0 , X_1 , X_2 , and X_{T-1} is arranged horizontally from left to right. Each circle is connected to the next by a rightward-pointing arrow labeled A . Below each circle, there is a corresponding blue square labeled O_0 , O_1 , O_2 , and O_{T-1} , respectively. Each circle is connected to its corresponding square by a vertical black arrow labeled B . A dashed red horizontal line runs across the image, intersecting the vertical arrows. The sequence continues with ellipses between X_2 and X_{T-1} , and after X_{T-1} , indicating continuation.

Several influential recent papers I

	Input	Output	Method	Dataset	Domain	Where Published
AutomaTikZ	Captions	TikZ	SFT (PEFT with LoRA)	120k (caption/code) pairs	ArXiv (cs.CL , ...), TeX SE	ICLR'24
TikZero	Captions	TikZ	SFT	460k (caption/code) pairs	ArXiv (cs.CL , ...), TeX SE	ICCV'25
TikZilla	Descriptions	TikZ	SFT + RL	2M (description/code) pairs	ArXiv (cs.CL , ...), GitHub, TeX SE	ICLR'26
ChartMimic	Instruction/Reference Image/Data	Python (Matplotlib)	LLMs out of the box	4800 (figure, instruction, code) triplets	ArXiv, Matplotlib gallery, Stackoverflow, Twitter, Reddit	ICLR'25

Several influential recent papers I

	Input	Output	Method	Dataset	Domain	Where Published
AutomaTikZ	Captions	TikZ	SFT (PEFT with LoRA)	120k (caption/code) pairs	ArXiv (cs.CL , ...), TeX SE	ICLR'24
TikZero	Captions	TikZ	SFT	460k (caption/code) pairs	ArXiv (cs.CL , ...), TeX SE	ICCV'25
TikZilla	Descriptions	TikZ	SFT + RL	2M (description/code) pairs	ArXiv (cs.CL , ...), GitHub, TeX SE	ICLR'26
DeTikZify	Sketches + Images	TikZ	SFT + Monte Carlo Tree Search	360k (code/image) pairs	ArXiv (cs.CL , ...), TeX SE	NeurIPS'24

Scientific Figure Generation

	Input	Output	Method	Dataset	Domain	Where Published
AutomaTikZ	Captions	TikZ	SFT (PEFT with LoRA)	120k (caption/code) pairs	ArXiv (cs.CL , ...), TeX SE	ICLR'24
TikZero	Captions	TikZ	SFT	460k (caption/code) pairs	ArXiv (cs.CL , ...), TeX SE	ICCV'25
TikZilla	Descriptions	TikZ	SFT + RL	2M (description/code) pairs	ArXiv (cs.CL , ...), GitHub, TeX SE	ICLR'26
ChartMimic	Instuction/Reference Image/Data	Python (Matplotlib)	LLMs out of the box	4800 (figure, instruction, code) triplets	ArXiv, Matplotlib gallery, Stackoverflow, Twitter	ICLR'25

Several influential recent papers II

	Input	Output	Method	Dataset	Domain	Where Published
ScImage	Instructions	TikZ, Python, Direct Image	LLMs out of the box	404 (chart, math function, matrix, table) types	Human- curated	ICLR'25
VisCoder2	Task + Style Description	TikZ, Python, SVG, VegaLite, ...	SFT + Self-debug	680K (bars, lines, areas, 3D, scatter, matrix, heatmap, ...) types	Open-source repositories	ICLR'26
MatPlotAgent	User query, Raw data	Python (Matplotlib)	Agentic (Code agent + Visual agent)	100 (sankey diagram, sunburst chart, radial plots, chord diagram, streamplot) types	Matplotlib Gellary, OriginLab Graph Gellary	ACL Findings'24

Several influential recent papers III

	Input	Output	Method	Dataset	Domain	Where Published
ChartMimic	Image Instruction+ Data	Python (Matplotlib)	LLMs out of the box	4800 (figure, instruction, code) triplets	ArXiv, Matplotlib gallery, Stackoverflow, Twitter, Reddit	ICLR'25
PaperBanana	Methodology Description/ Diagram Caption	Direct Image	Agentic (Linear planning + Iterative refinement, Retrieval) based on Nano-Banana-Pro	292 (science & application, generative & learning, agent & reasoning, vision & perception) categories	NeurIPS 2025 publications	ArXiv'2026

Summary

- Dataset building and Finetuning vs. Benchmarking
- Finetuning vs. Agentic Modeling
- Direct Image vs. Code (TikZ vs. Python vs. Others)
- Inputs: Text (Captions vs. Descriptions vs. Instructions) vs. Image

Summary

- Dataset building and Finetuning vs. Benchmarking
- Finetuning vs. Agentic Modeling
- Direct Image vs. Code (TikZ vs. Python vs. Others)
- Inputs: Text (Captions vs. Descriptions vs. Instructions) vs. Image

Other issues:

- Domain of the data
- Input language (English, German, Arabic, Farsi, Chinese, ...)

Evaluation

- Human Evaluation (Text-to-Image, Image-to-Image)
- Automatic Evaluation

- CLIPScore

- DreamSim

- TeX Edit Distance

- CrystalBLEU

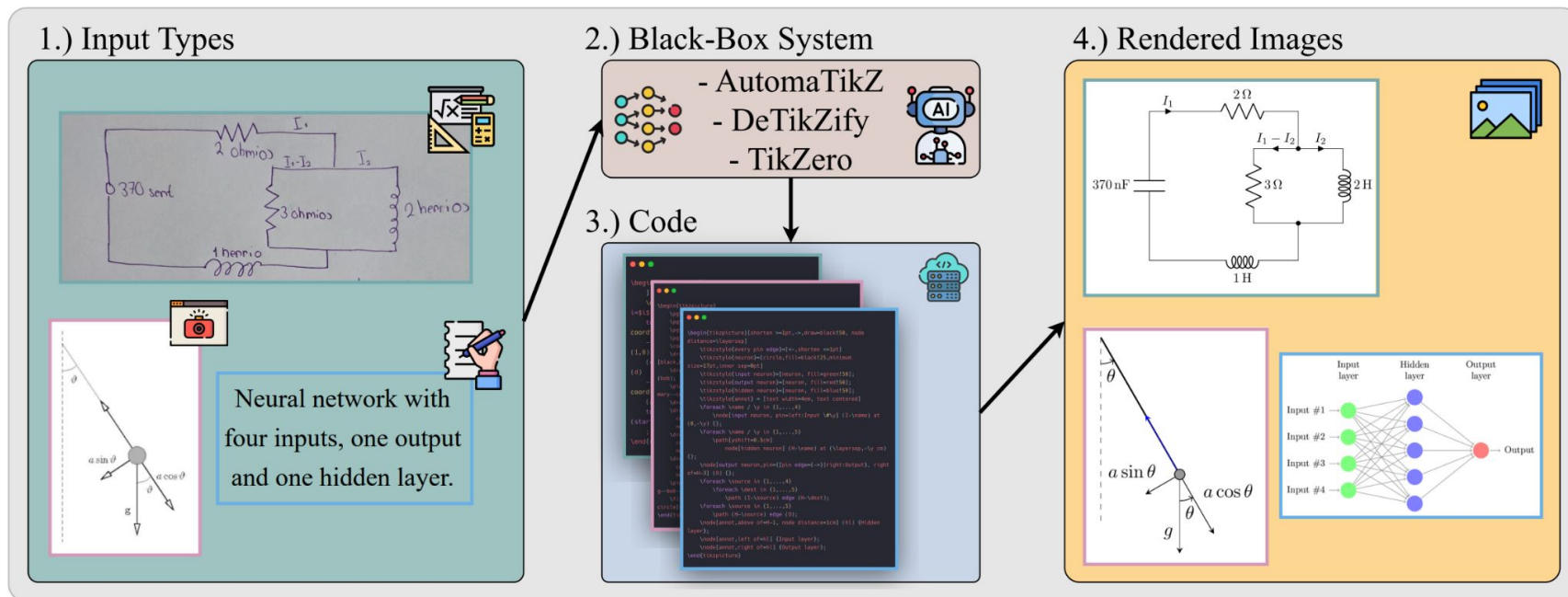
- VLM-as-a-Judge

- SciEval (forthcoming)



Scientific Figure Generation - Code Generation

- Using code to generate scientific figures is gold standard due to precision and human interpretability (recommended by top conferences) [2]



Text-to-Code with TikZ: AutomaTikZ [3]

Fine-tunes LLaMA models on

caption/TikZ pairs from ArXiv and

TeX StackExchange discussions

- also integrates CLIP with

Llama

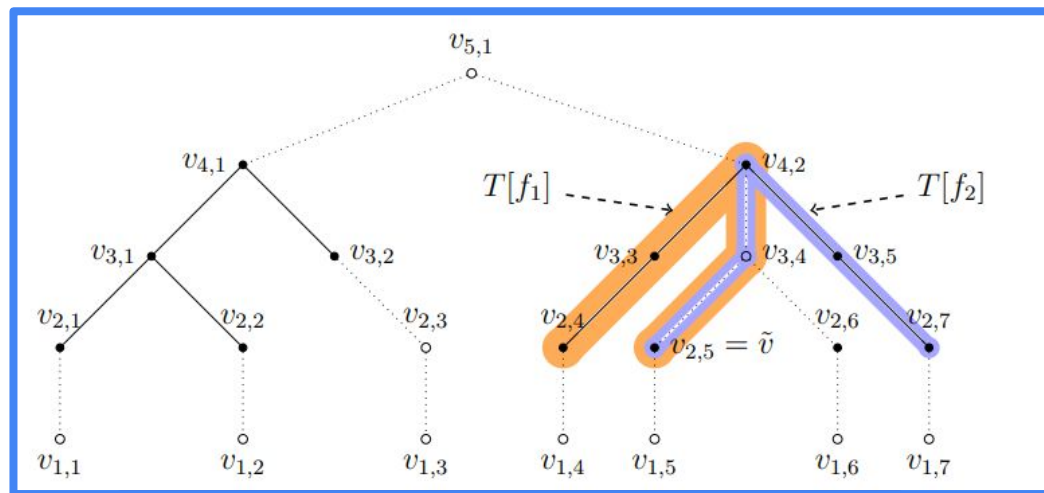
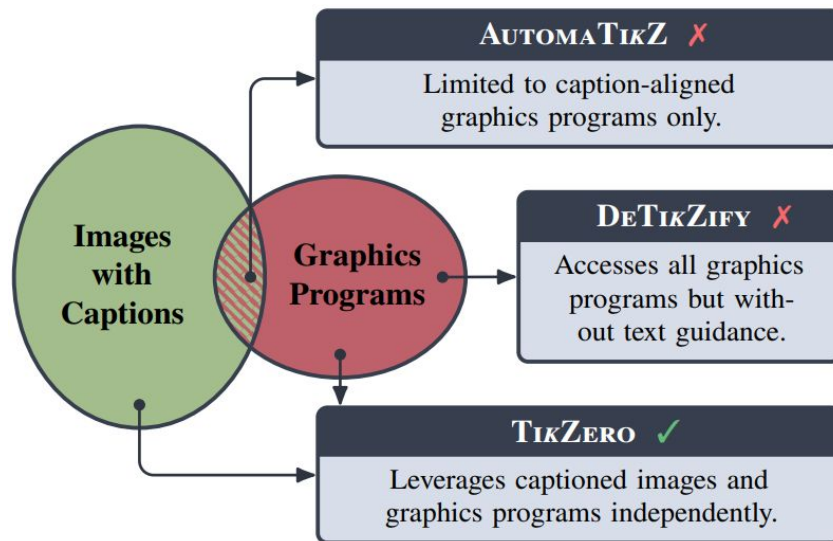


FIGURE 2. Illustration of the proof of Theorem 2.7

Text-to-Code with TikZ: TikZero [4]

- Distills knowledge from scientific images without available graphics programs → More data
- Reminiscent of “pivot language idea” in MT

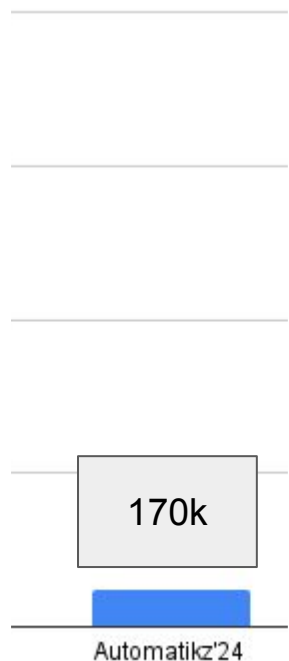


Text-to-Code with TikZ: TikZilla [5]

- ❑ Data size:
 - ❑ Increases dataset size
- ❑ Data quality:
 - ❑ Uses VLM-generated image descriptions
- ❑ Modeling:
 - ❑ Applies reinforcement learning (RL) for post-training after SFT

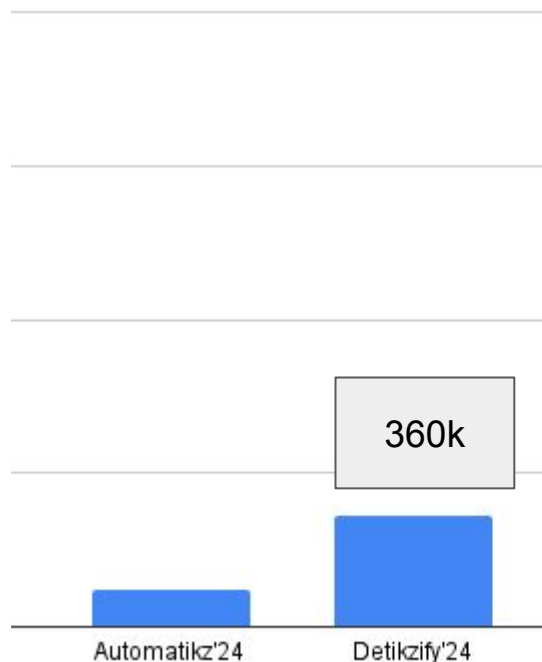
TikZilla - Deep Dive

1) Dataset size increase (in 1000)



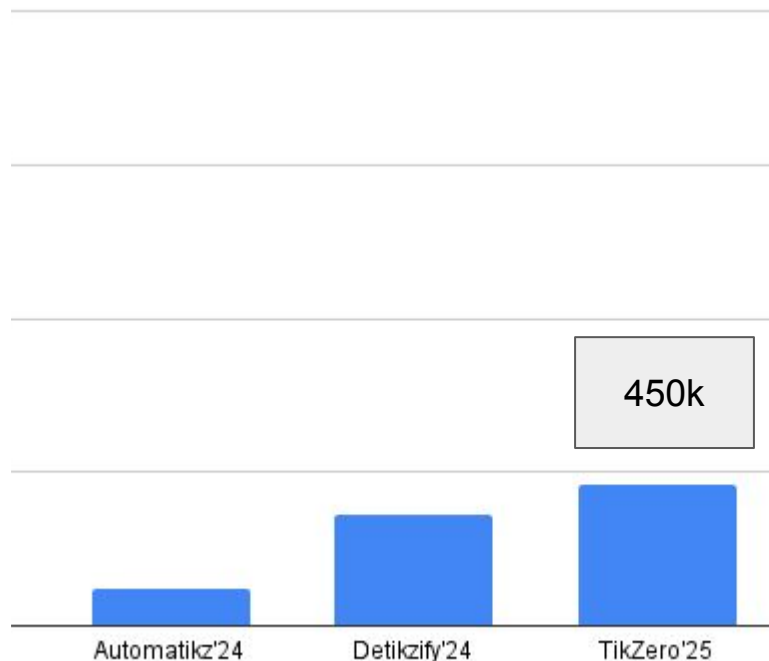
TikZilla - Deep Dive

1) Dataset size increase (in 1000)



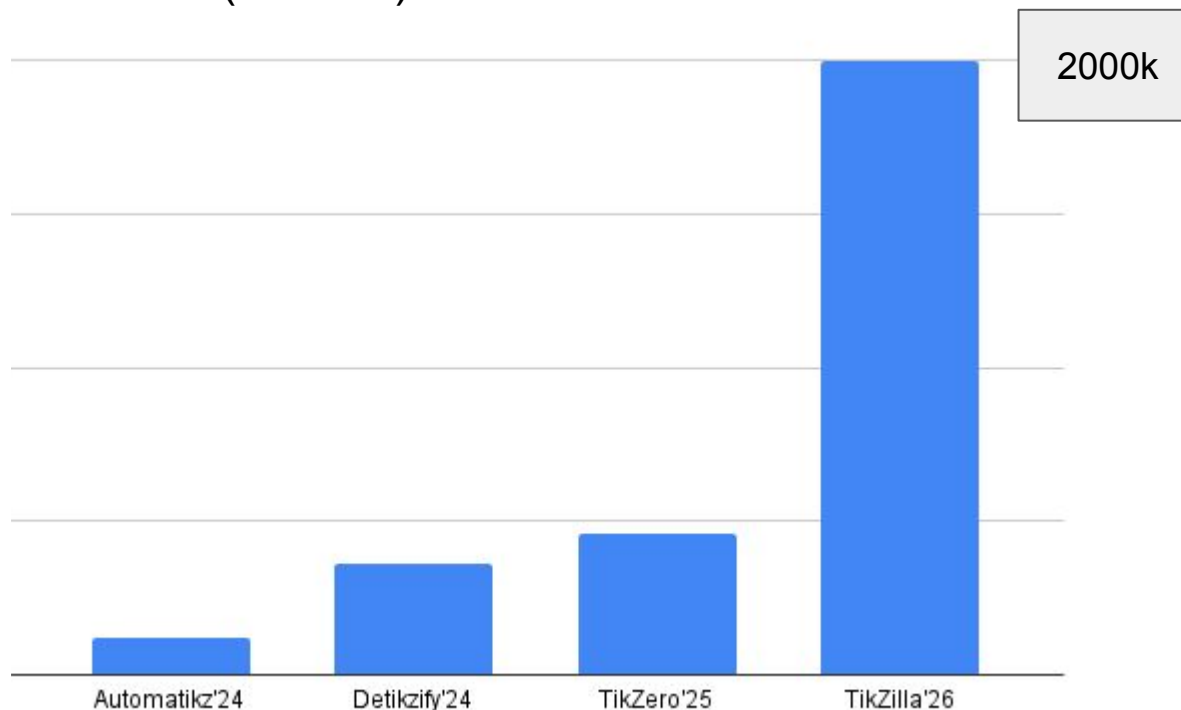
TikZilla - Deep Dive

1) Dataset size increase (in 1000)



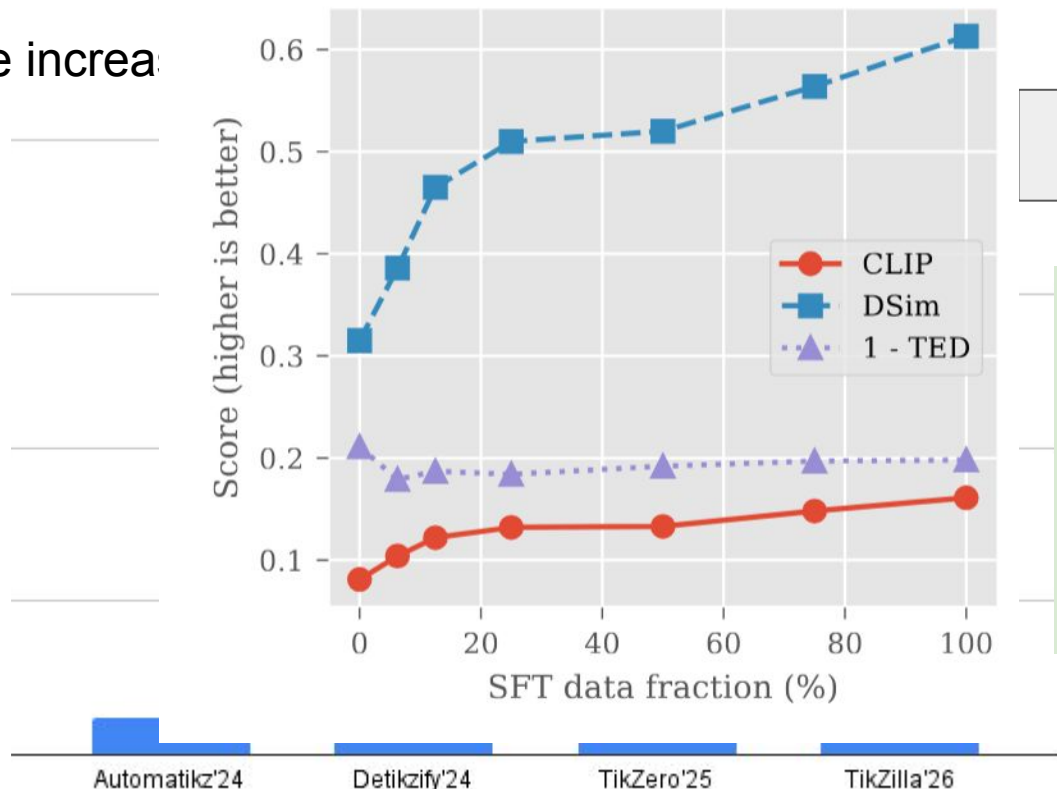
TikZilla - Deep Dive

1) Dataset size increase (in 1000)



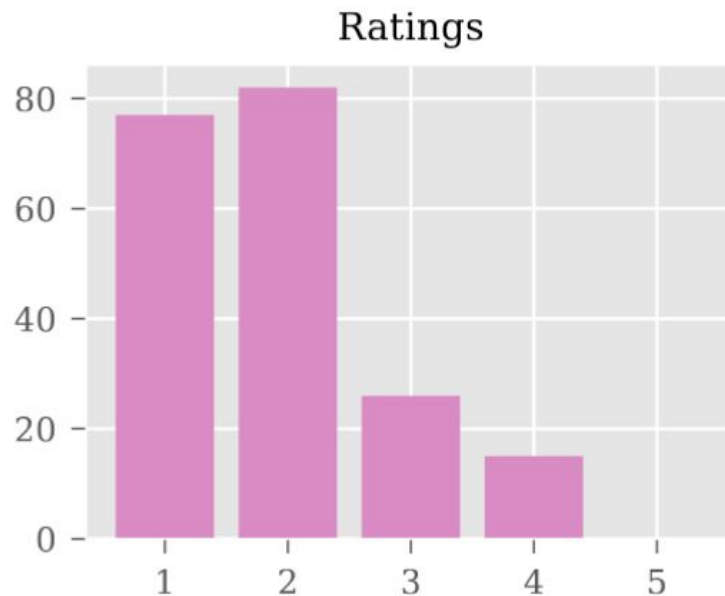
TikZilla - Deep Dive

1) Dataset size increa



TikZilla - Deep Dive

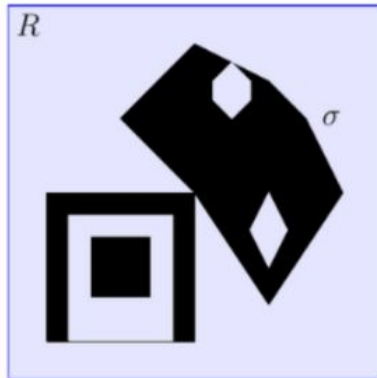
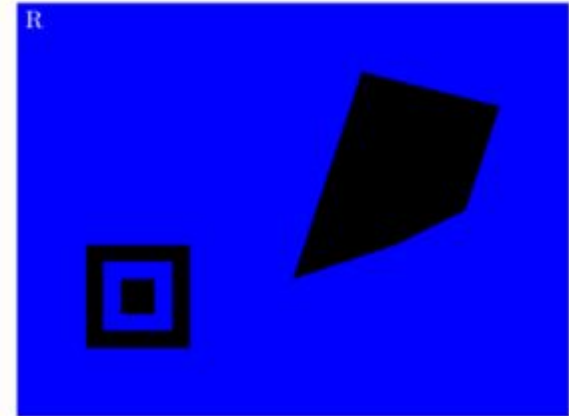
2) Dataset **quality** increase



TikZilla - Deep Dive

Description: A blue rectangle labeled R in the top-left corner. Inside the rectangle, there are two black geometric figures. At the lower-left side, is a layered square pattern composed of three squares, a small black square at the center, surrounded by a blue square matching the background color of the rectangle, surrounded by a black square. Toward the upper-right is an irregular black polygon labeled σ . The two shapes have the black background color of the rectangle. The other is diamond shaped at the bottom.

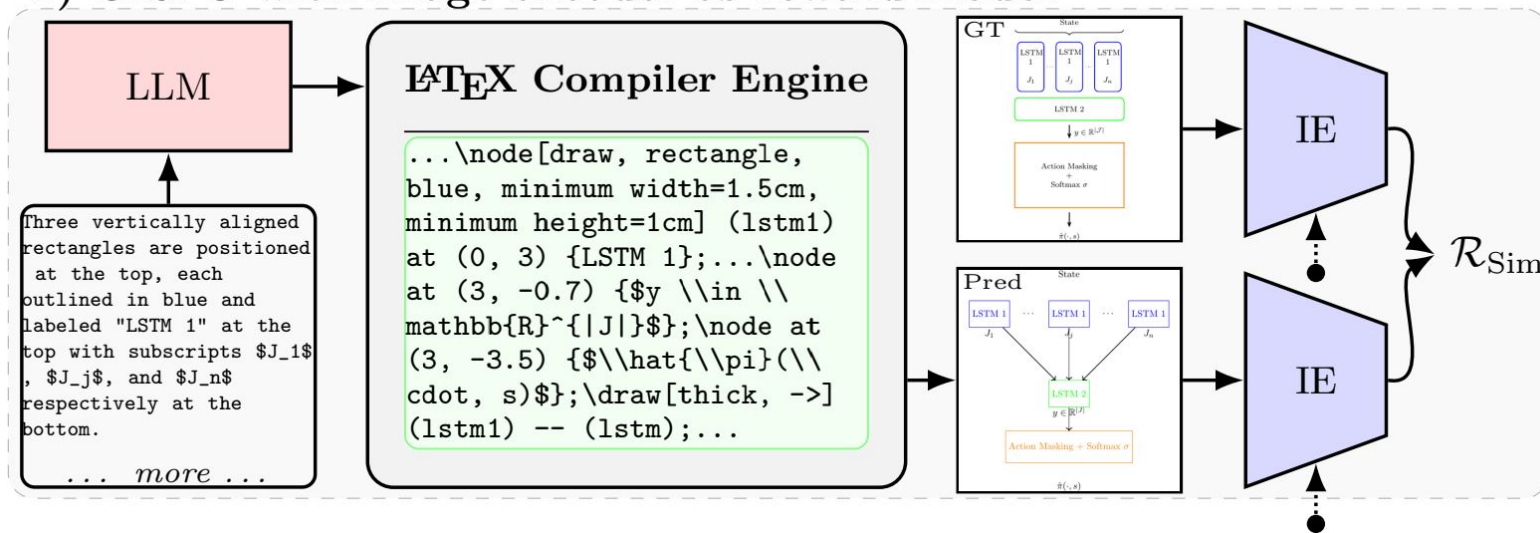
Description (GPT-4o)



TikZilla - Deep Dive

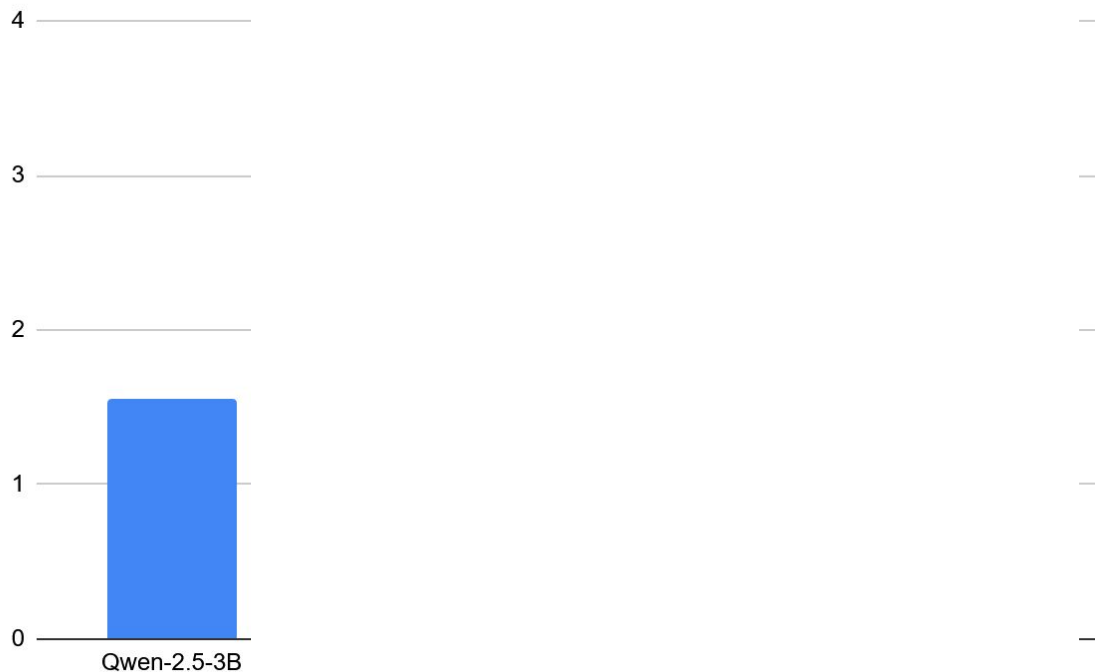
3) Better Modeling: SFT + RL

2) GRPO with image encoder as reward model



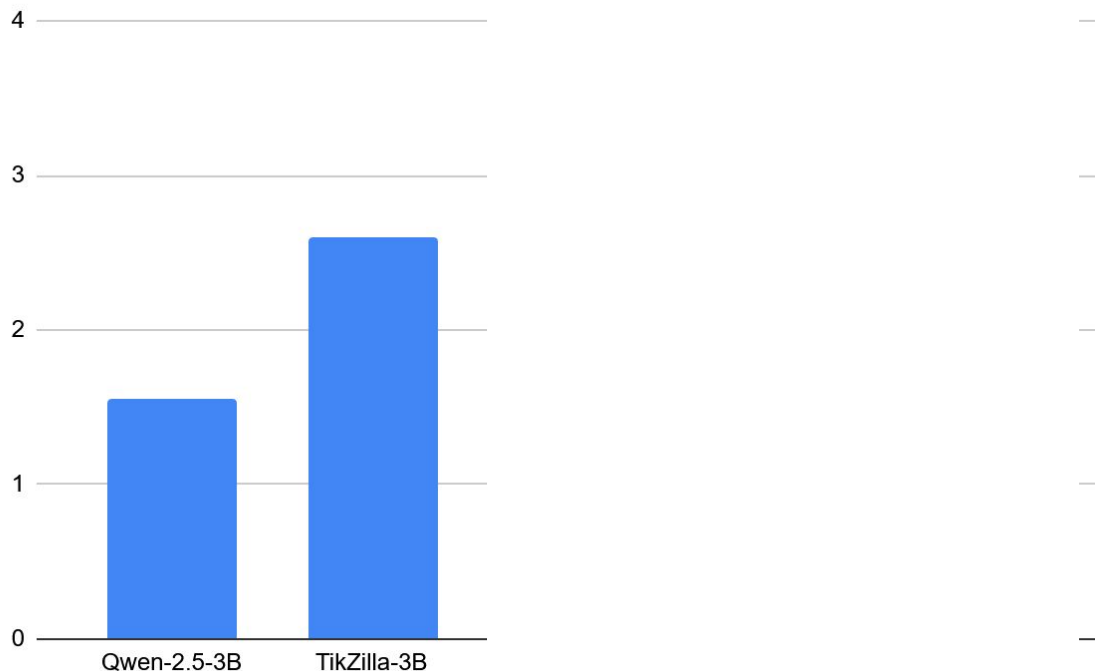
TikZilla - Deep Dive

3) Results: Small open-source models competitive to large proprietary models



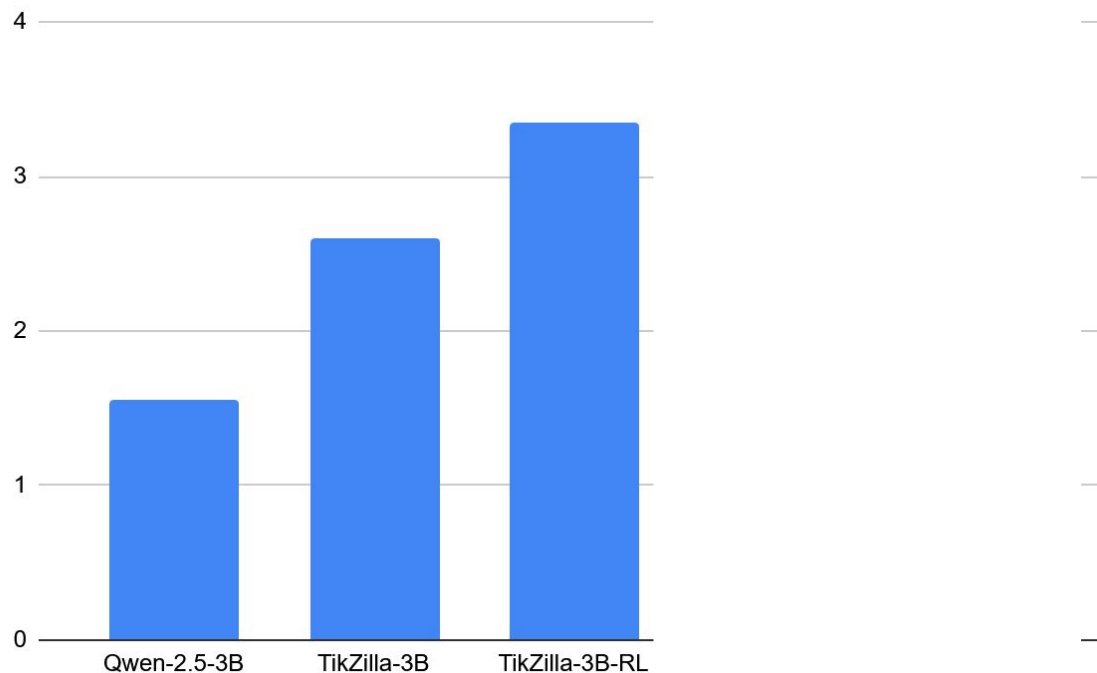
TikZilla - Deep Dive

3) Results: Small open-source models competitive to large proprietary models



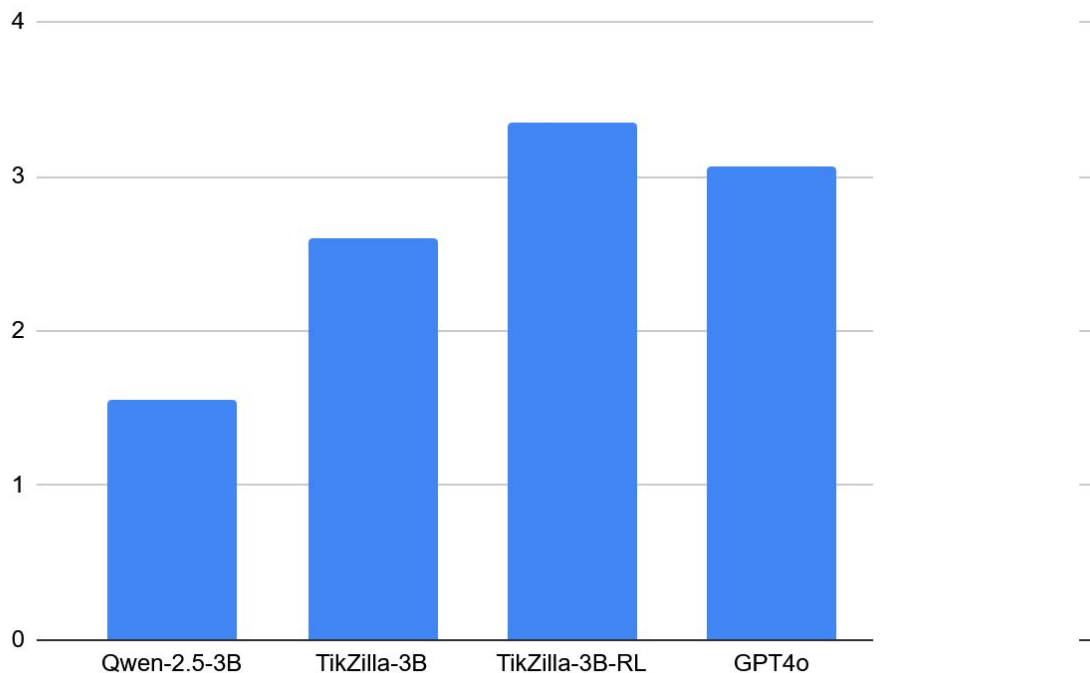
TikZilla - Deep Dive

3) Results: Small open-source models competitive to large proprietary models



TikZilla - Deep Dive

3) Results: Small open-source models competitive to large proprietary models



Text-to-Code with Python: MatPlotAgent [6]

Agentic system to generate Matplotlib figures with raw data and user

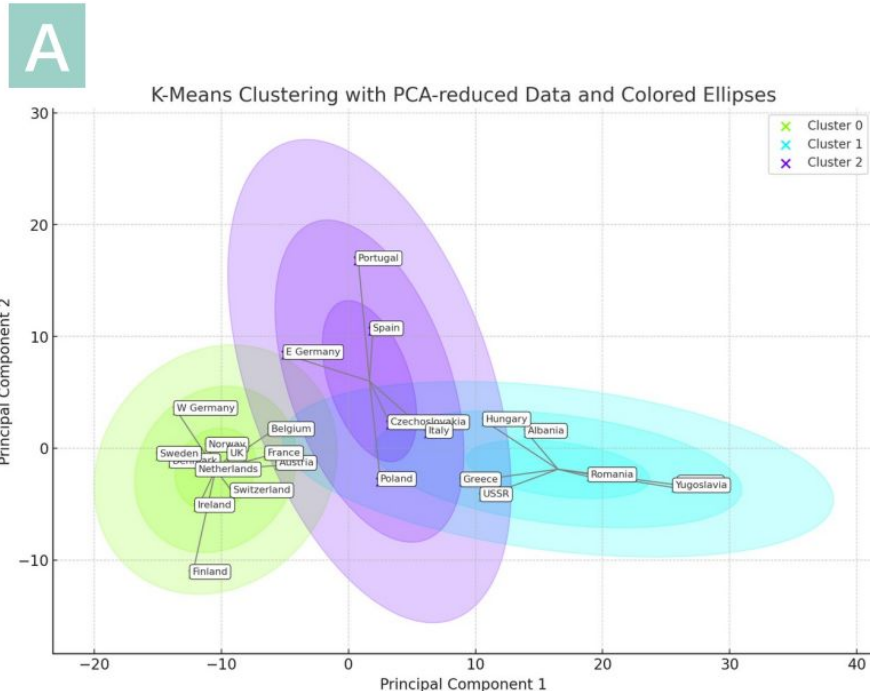
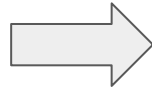
instructions

A

Country	Red Meat	White Meat	Eggs
Albania	10.1	1.4	0.5
Austria	8.9	14	4.3
Belgium	13.5	9.3	4.1

A

I have data of protein consumption in 24 European countries named data.csv. ... Write a Python code to visualize this data using a 2D scatter plot with K-Means clustering into three distinct color-coded clusters. ...



Text-to-Code (Multiple): VisCoder2 [7]

LLMs finetuned on 12
programming
languages


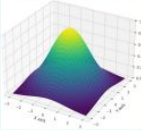

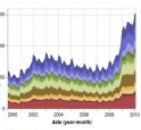

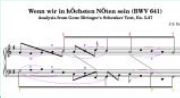



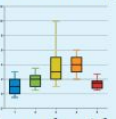

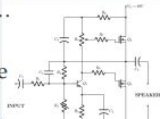

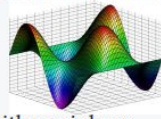


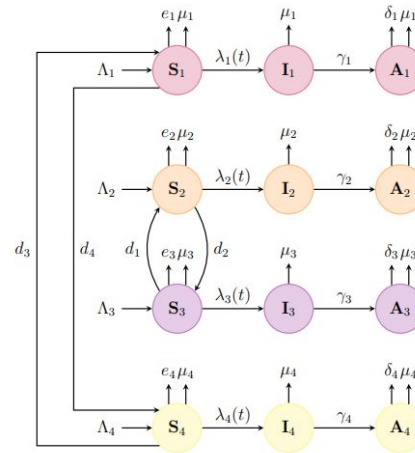
<p> Python</p> <p>Task: Generate a three-dimensional surface plot, the plot visualizes the relationship between three variables, X, Y, and Z,</p> <p>Style: The visual features a smooth, colorful surface with a gradient transitioning</p>  <p>Visual Cate. : 3D; Subtype: Surface</p>	<p> Vega-Lite</p> <p>Task: Create an area chart to visualize the sum of counts over time, categorized by different series. The x-axis represents time in</p> <p>Style: The area segments are stacked on top of each other, each filled with erse palette</p>  <p>Visual Cate. : Areas; Subtype : Stacked-Area</p>	<p> LilyPond</p> <p>Task: Generate a musical score for piano that presents an analysis of J.S. Bach's piece "Wenn wir in höchsten Nöten sein"</p> <p>Style: The staves are spaced apart, with the right hand above the left hand. Notes are</p>  <p>Visual Cate. : Music; Subtype : Sheet Music</p>	<p> Mermaid</p> <p>Task: Generate a Gantt chart to visualize a project timeline, illustrating tasks across different sections such as "A section," "Critical</p> <p>Style: The visual consists of horizontal bars aligned with labeled sections on the left, each</p>  <p>Visual Cate. : Diagramming; Subtype : Gantt</p>
<p> SVG</p> <p>Task: Generate a box plot to display the distribution of data across five different categories. Each category is represented by a separate</p> <p>Style: The visual consists of five colored boxes aligned horizontally, each with a distinct color</p>  <p>Visual Cate. : Distribution; Subtype : Box-Plot</p>	<p> LaTeX</p> <p>Task: Generate a schematic diagram of an electronic circuit that includes various components such as resistors, capacitors,</p> <p>Style: The components are labeled with their respective symbols and names, and</p>  <p>Visual Cate. : Diagramming; Subtype : Electrical Circuit</p>	<p> Asymptote</p> <p>Task: Generate a 3D surface plot representing a mathematical function defined by the cosine and sine of the coordinates. The surface is</p> <p>Style: Create a colorful, grid-like 3D visualization with a rainbow gradient applied to the</p>  <p>Visual Cate. : 3D; Subtype : Surface</p>	<p> HTML</p> <p>Task: Generate a polar area chart to represent the market share percentages of different smartphone brands. The chart includes</p> <p>Style: The chart is centered on a light gray background with a title above it</p>  <p>Visual Cate. : Radial & Polar; Subtype : Radial Area</p>

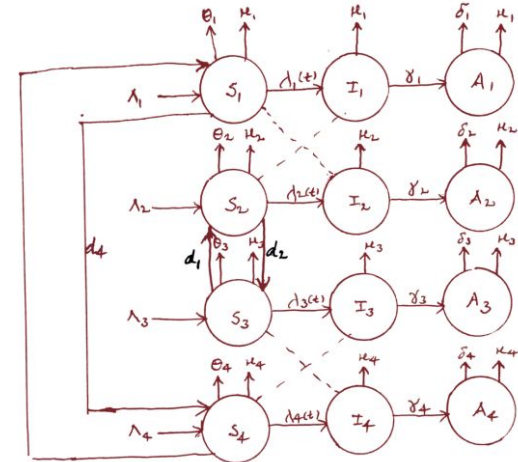
Image-to-Code with TikZ: DeTikZify [8]

Reconstructs scientific figures from images or sketches.

Reference Figures

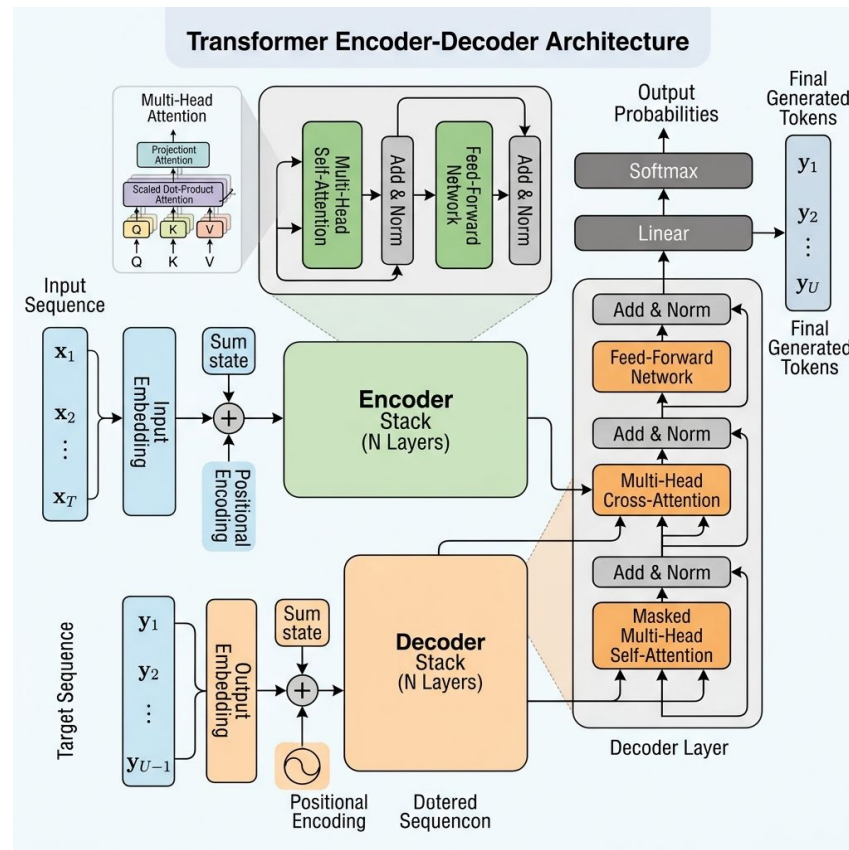


Real Sketches



Direct Image Generation: PaperBanana [9]

- ❑ Google's Agentic framework that generates scientific figures directly from text
- ❑ Example: "A transformer-based encoder-decoder architecture with multi-head self-attention"

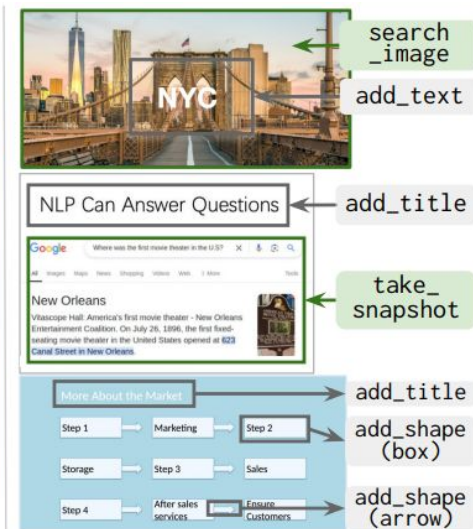
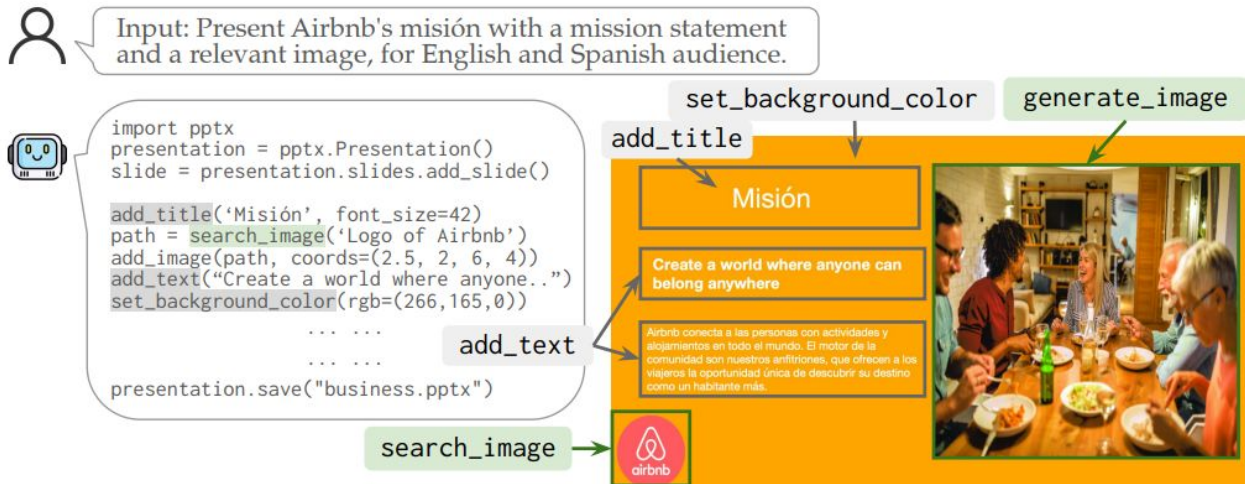


Scientific Slide and Poster Generation

Scientific Slide and Poster Generation

AutoPresent [10]:

- Fine-tunes an LLM to generate slides from detailed natural language instructions with images, detailed instructions only, or high-level instructions.




Scientific Slide and Poster Generation

Paper2Poster [11]:

☐ Transforms 22-page paper into a finalized and editable .pptx poster for just \$0.005.

Bisimulation Makes Analogies in Goal-Conditioned Reinforcement Learning

Philippe Hansen-Estruch¹, Amy Zhang², Ashvin Nair¹, Patrick Yeo¹, Sergey Levine¹
¹University of California, Berkeley, ²Meta AI Research



Introduction

Goal-conditioned RL has the potential to train agents that can accomplish a variety of tasks when given goals. In this work, we study a harder problem, for this is not always optimal. In this work, we study a harder problem: to train goal-conditioned agents that generalize across tasks while retaining invariance to information not changeable by the agent.

Method

Algorithm: GCB Training Algorithm

- Train ϕ , ψ , ψ' using \mathcal{D}
- For each task t , train π_t using \mathcal{D}_t
- For each task t , train π_t using \mathcal{D}_t
- For each task t , train π_t using \mathcal{D}_t

Visualizations and Experimental Results

Analogy Arithmetic Visualized:

- State + Analogous Task = Inferred Goal

$\psi(\text{Apple}) + \phi(\text{Banana}) = \psi(\text{Banana})$

$\psi(\text{Banana}) + \phi(\text{Apple}) = \psi(\text{Apple})$

Value Bounds and Sufficiency Results

Proposition 6.2. For any two state goal pairs $(s_1, g_1), (s_2, g_2) \in (S, G)$ and a given policy π ,

$$|V^\pi(s_1, g_1) - V^\pi(s_2, g_2)| \leq d^\pi(s_1, s_2, g_1, g_2). \quad (9)$$

Proposition 6.5. For any reward function $R \in \mathbb{R}$ or \mathcal{A} as defined above and a given policy π ,

$$|V^\pi(s_1, g_1) - V^\pi(s_2, g_2)| \leq \sum_{i=0}^{\infty} \gamma^i d^\pi(s_1, s_2, g_1, g_2). \quad (10)$$

which shows that such a metric is universal for any deterministic task with any potential reward function.

(a) Author-designed poster.

Bisimulation Makes Analogies in Goal-Conditioned Reinforcement Learning

Philippe Hansen-Estruch¹, Amy Zhang², Ashvin Nair¹, Patrick Yeo¹, Sergey Levine¹
¹University of California, Berkeley, ²Meta AI Research

Abstract

- Building generalizable goal-conditioned agents is crucial for RL.
- Traditional RL provides exact goals, often unrealistic.
- Propose goal-conditioned bisimulation for skill reuse.
- Captures functional equivalence for new goals.
- Generalizes to new goals in simulation tasks.
- Sufficient for downstream tasks with state-only reward.

Introduction

- Goal-conditioned RL enables training agents for diverse tasks.
- Goal representation is crucial for policy interpretation.
- Functional equivalence aids in generalizing across tasks.
- Agents can specify goals without exact goal images.
- Functionally equivariant representations capture state-goal changes.

Related Work

- Our approach enhances goal-conditioned RL with general-purpose representations.
- Focus on representation learning, comparing state abstractions and self-supervised learning.
- Bisimulation metrics help measure behavioral similarity for control tasks.

Preliminaries

- Goal-conditioned Markov Decision Process (GCMDP) includes state space, action space, dynamics model, goal space, and sparse reward function.
- Bisimulation groups states with equivalent behaviors, preserving reward sequences.

GCB: Embedding Spaces

- Combines representation learning with downstream control.
- Pairs with any goal-conditioned RL method.

Functional Equivalence

- Extends bisimulation to goal-conditioned metrics.
- Defines equivalence over tasks for compositional generalization.
- Constructs state abstraction using arithmetic in latent space.

Experiments and Results

- GCB evaluated against other representation learning methods.
- Captures functional equivalence using state-goal pairs.
- Strong performance on offline goal-conditioned tasks.
- Comparison with contrastive and reconstruction-based methods.
- GCB excels in capturing task analogies.
- Superior generalization capabilities demonstrated.

Discussion

- GCB introduces goal-conditioned bisimulation for functional equivalence.
- Enables agents to achieve unseen goals with analogous task representations.
- Outperforms existing methods in goal-conditioned tasks.
- Structured representations bound value differences across

Figure 1: State Goal Encoder

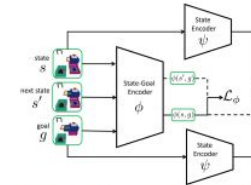


Table 1: Performance on various tasks

Task	Method	Success Rate
Task 1	GCB	0.95
	Baseline	0.85
Task 2	GCB	0.90
	Baseline	0.75
Task 3	GCB	0.88
	Baseline	0.70

(b) PosterAgent-generated poster.

Concluding Remarks

- ❑ A lot of progress recently in figure generation, understanding and beyond
- ❑ New large-scale datasets are being proposed
- ❑ Models: SFT, RL, Agentic
- ❑ Large Commercial often better, but fine-tuning small models can be competitive
- ❑ Future work:
 - ❑ Need for tailored evaluation metrics
 - ❑ Editing
 - ❑ Combination of code(s)+direct image generation

References

1. Wang, Zirui, et al. "Charxiv: Charting gaps in realistic chart understanding in multimodal llms." *Advances in Neural Information Processing Systems* 37 (2024): 113569-113697.
2. Eger, Steffen, et al. "Transforming science with large language models: A survey on ai-assisted scientific discovery, experimentation, content generation, and evaluation." *arXiv preprint arXiv:2502.05151* (2025).
3. Belouadi, Jonas, Anne Lauscher, and Steffen Eger. "Automatikz: Text-guided synthesis of scientific vector graphics with tikz." *ICLR 2024*
4. Belouadi, Jonas, et al. "Tikzero: Zero-shot text-guided graphics program synthesis." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2025.
5. Greisinger, Christian, and Steffen Eger. "TikZilla: Scaling Text-to-TikZ with High-Quality Data and Reinforcement Learning." *ICLR 2026*
6. Yang, Zhiyu, et al. "Matplotagent: Method and evaluation for llm-based agentic scientific data visualization." *Findings of the Association for Computational Linguistics: ACL 2024*. 2024.
7. Ni, Yuansheng, et al. "VisCoder2: Building Multi-Language Visualization Coding Agents." *ICLR 2026*
8. Belouadi, Jonas, Simone Ponzetto, and Steffen Eger. "Detikzify: Synthesizing graphics programs for scientific figures and sketches with tikz." *Advances in Neural Information Processing Systems* 37 (2024): 85074-85108.
9. Zhu, Dawei, et al. "PaperBanana: Automating Academic Illustration for AI Scientists." *arXiv preprint arXiv:2601.23265* (2026).
10. Ge, Jiaxin, et al. "Autopresent: Designing structured visuals from scratch." *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025.
11. Pang, Wei, et al. "Paper2poster: Towards multimodal poster automation from scientific papers." *ICLR 2026*

