上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

SCHOOL OF COMPUTING & DATA SCIENCE
The University of Hong Kong

Towards a Foundational Visual-Programmatic Interface for Code Intelligence

Qiushi Sun
qiushisun.github.io
𝕏 @qiushi_sun

JanusCoder: Towards a Foundational Visual-Programmatic Interface for Code Intelligence

Qiushi Sun*, Jingyang Gong*, Yang Liu*, Qiaosheng Chen*, Lei Li, Kai Chen, Qipeng Guo, Ben Kao, Fei Yuan

Background

The scope of code intelligence is rapidly expanding

beyond text-only coding tasks (e.g., LCB, SWE) to the rich visual outputs that programs generate (charts, UIs, animations), typically:

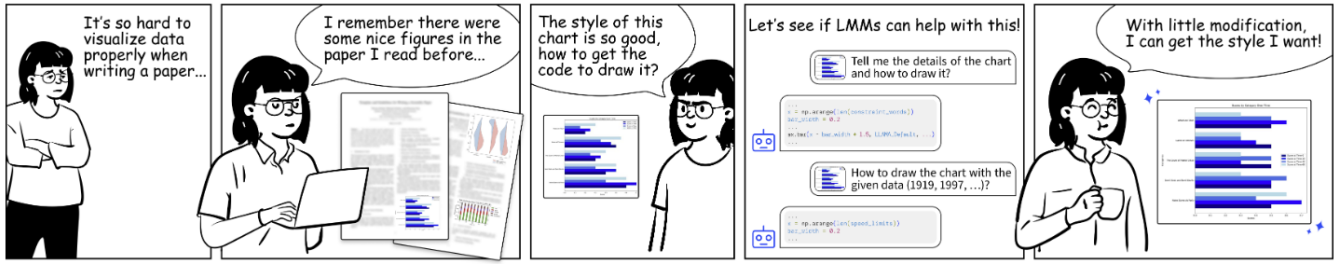
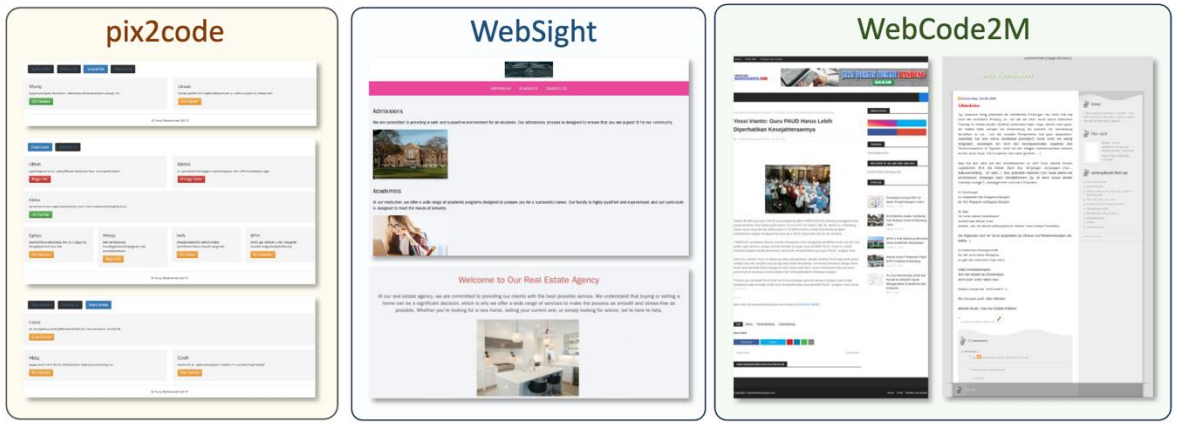


Chart2Code



Webpage to Code

Coding for visual contents!

Background

Coding for visual contents

Create a hierarchical sunburst (donut) chart to visualize the relationship between scientific software and their corresponding task categories, similar to the...

显示代码 <>

显示思路

scientific...e_sunburst
HTML

Python

```python
import plotly.graph_objects as go
import numpy as np

# 1. Define the hierarchical data structure
# The root node will be implicit ("") and we start with the applications.
# This structure creates the "hole" in the center.
labels = [
    # Level 1: Applications
    'GrassGIS', 'Celestia', 'Kalegebra', 'ChimeraX', 'Lean4', 'TeXStudio',

    # Level 2: Task Types (Children of GrassGIS)
    'Map Display', 'Image Analysis', 'Data Editing', 'Basic Manipulation',
    # Level 2: Task Types (Children of Celestia)
    'Simulation', 'Data Queries', 'Spatial Reasoning', 'Basic Settings',
    # Level 2: Task Types (Children of Kalegebra)
    'Basic Ops.', 'Direct Instructions', 'Open Problems', 'Adv. Knowledge',
    # Level 2: Task Types (Children of ChimeraX)
    'Software Config.', 'Struct. Analysis', 'Data Interp.', 'Struct. Prediction',
    # Level 2: Task Types (Children of Lean4)
    'Pure Math', 'Algebra', 'Geometry', 'Analysis',
    # Level 2: Task Types (Children of TeXStudio)
    'Basic Usage', 'Layout', 'Figure Editing'
]

parents = [
    # Level 1: Applications (All children of the implicit root "")
    "", "", "", "", "", "",

    # Level 2: Task Types (Children of GrassGIS)
    'GrassGIS', 'GrassGIS', 'GrassGIS', 'GrassGIS',
    # Level 2: Task Types (Children of Celestia)
```

NL → Code

I found a very beautiful data visualization in the ScienceBoard paper, and I'd like to create a similar one. Please help me implement it using Python code.

显示思路

Certainly. That's an excellent visualization from the ScienceBoard paper, which benchmarks GUI agents on scientific software. This type of chart is a **nested pie chart**, also known as a **sunburst chart**.

You can create this in Python using **Matplotlib**. The main technique is to draw two separate pie charts on the same axes: one for the outer ring and one for the inner ring, setting their radii and widths carefully so they stack.

Here is the complete Python code to replicate the chart. I've extracted the data directly from the image you provided.

Python Code (using Matplotlib)

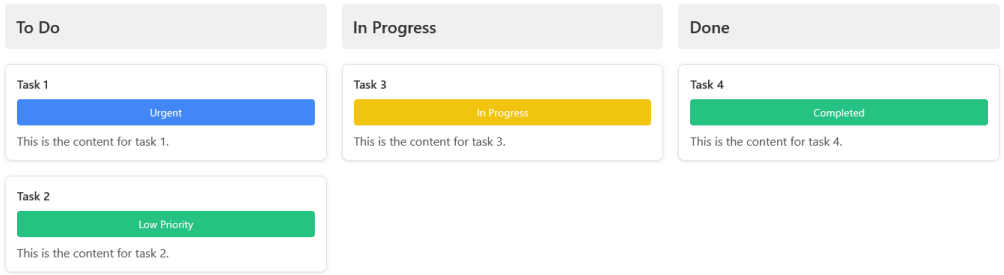
Python

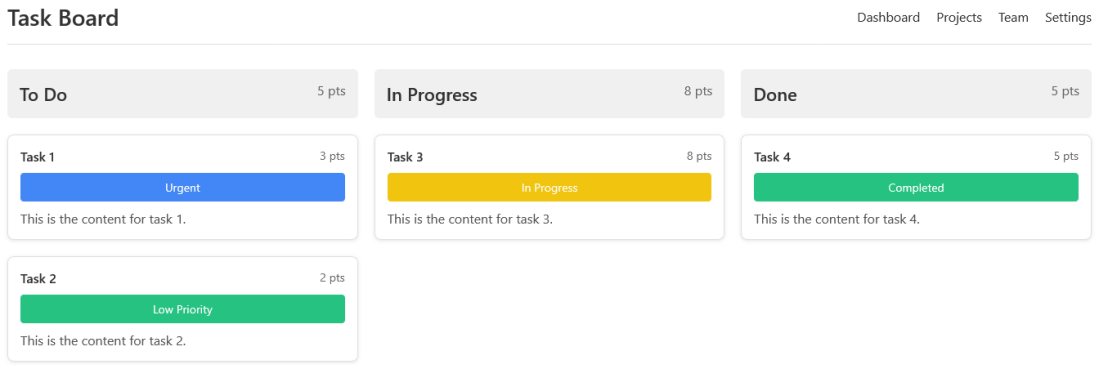
```python
import matplotlib.pyplot as plt
import numpy as np

# 1. Define the data structure from the image
# I've assumed sub-categories are equally weighted within their parent category.
```

NL + Figure → Code

Background

Coding for visual and interactive contents: WebUIs -> Generative UIs

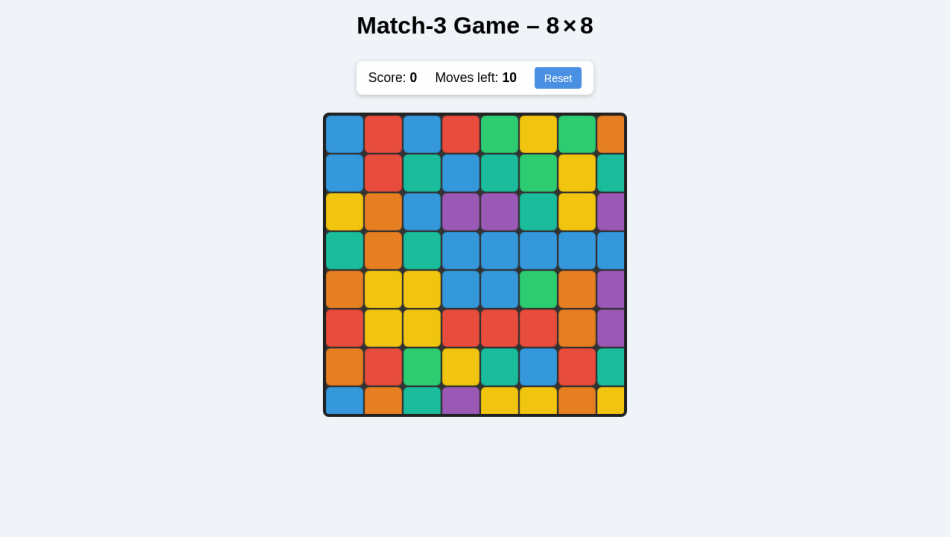


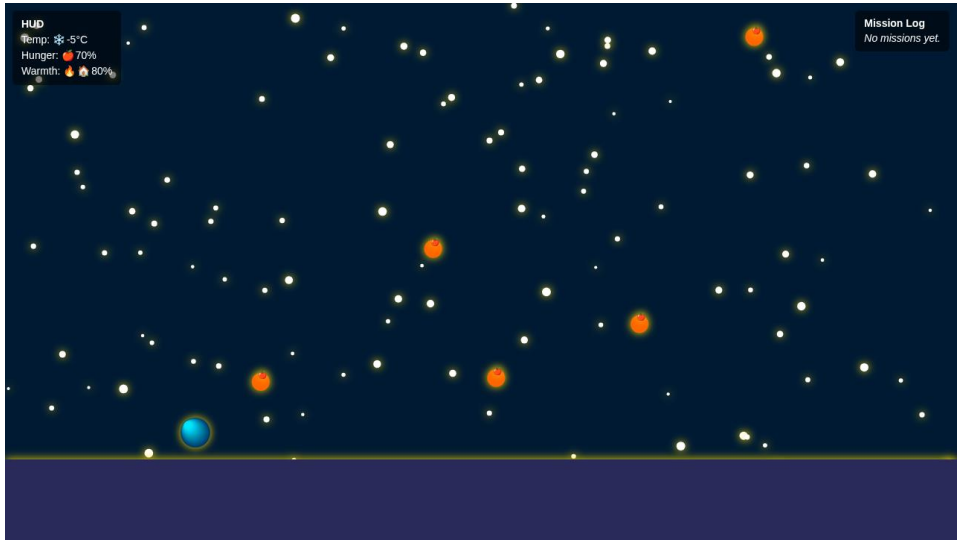


Given NL description + webpage screenshot, generate or edit the web UI.

Background

Coding for visual and interactive contents: artifacts / web demos



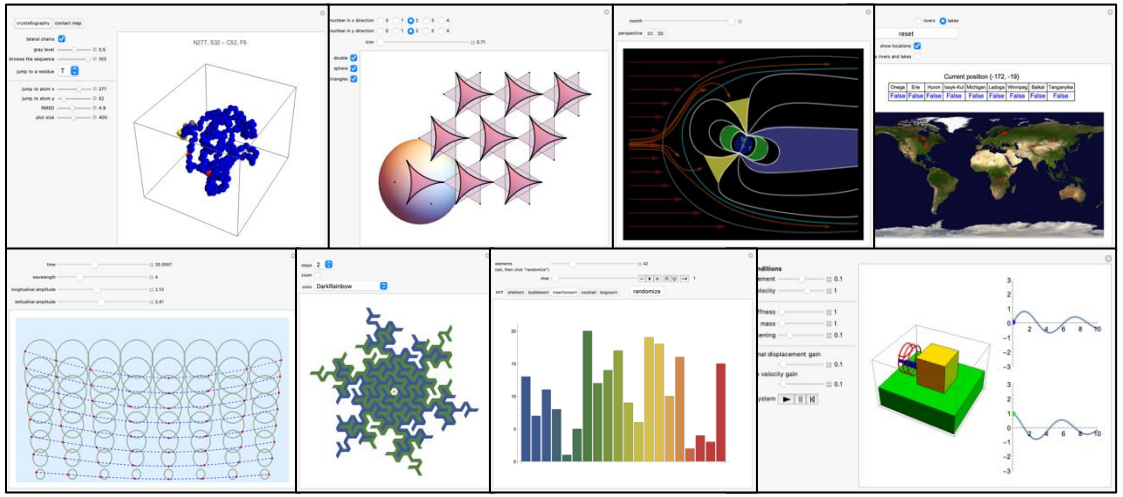


It is not only about visualization, but also about the underlying logic, algorithms ...

Background

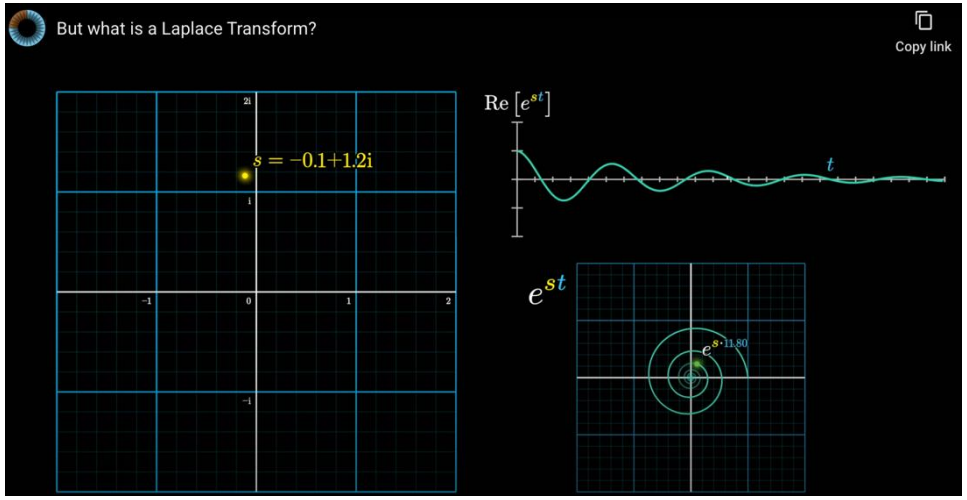
Coding for visual and interactive contents: scientific demos

WOLFRAM Demonstrations Project



"Code-driven demos"

3blue1brown video



"Code-driven videos"

Multimodal Code Intelligence

Leading players are also moving in this direction.

Qwen3-VL

3D Grounding	objects.	Open in Colab
Thinking with Images	Utilize image_zoom_in_tool and search_tool to facilitate the model's precise comprehension of fine-grained visual details within images.	Open in Colab
MultiModal Coding	Generate accurate code based on rigorous comprehension of multimodal information.	Open in Colab
Long Document Understanding	Achieve rigorous semantic comprehension of ultra-long documents.	Open in Colab
Spatial Understanding	See, understand and reason about the spatial information	Open in Colab

Quickstart

Below, we provide simple examples to show how to use Qwen3-VL with 🤖 ModelScope and 🤗 Transformers.

Gemini 2.5 Pro

The Gemini 2.X series are all built to be natively multimodal, supporting long context inputs of >1 million tokens and have native tool use support. This allows them to comprehend vast datasets and handle complex problems from different information sources, including text, audio, images, video and even entire code repositories. These extensive capabilities can also be combined to build complex agentic systems, as happened in the case of Gemini Plays Pokémon[1] (Zhang, 2025). Different models in the series have different strengths and capabilities: (1) Gemini 2.5 Pro is our most intelligent thinking model, exhibiting strong reasoning and code capabilities. It excels at producing interactive web applications, is capable of codebase-level understanding and also exhibits emergent multimodal coding abilities. (2) Gemini 2.5 Flash is our hybrid reasoning model with a controllable thinking budget, and is useful for most complex tasks while also controlling the tradeoff between quality, cost, and latency. (3) Gemini 2.0 Flash is our fast and cost-efficient non-thinking model for everyday tasks and (4) Gemini 2.0 Flash-Lite is our fastest and most cost-efficient model, built for at-scale usage. A full comparison of the models in the Gemini 2.X model family is provided in Table 1. Taken together,

However, it is clear that such capabilities come with a high barrier.

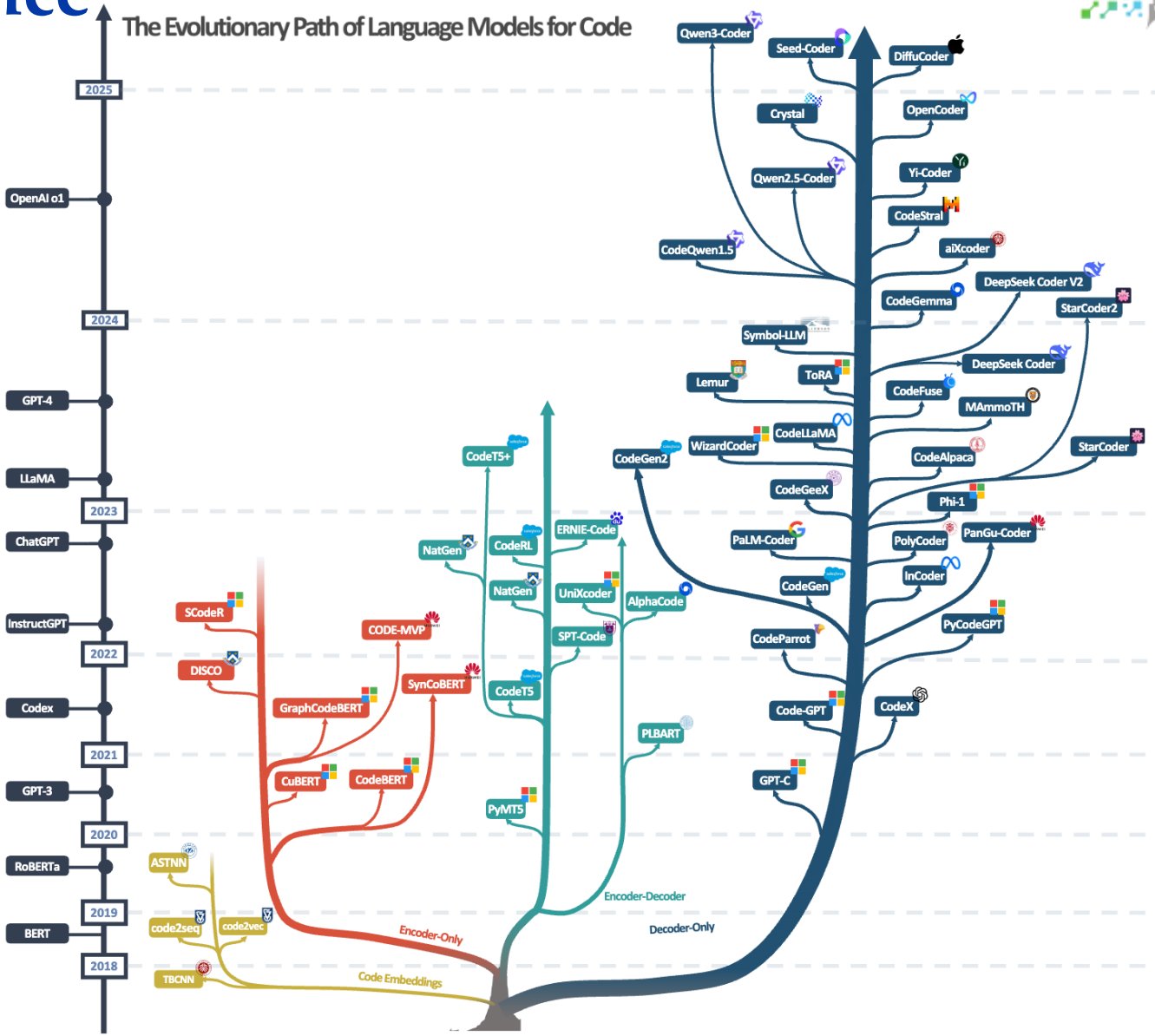
[1] https://github.com/QwenLM/Qwen3-VL

[2] Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities

Multimodal Code Intelligence

These capabilities are cool,
but they often require specialized coding skills and multimodal capability.

Challenge 1: Most current CodeLLMs are text-based and lack understanding of visual content.



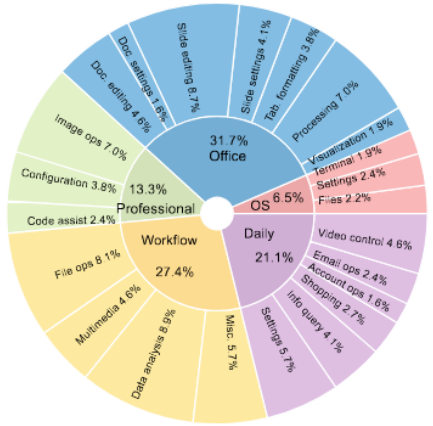
[3] A Survey of Neural Code Intelligence: Paradigms, Advances and Beyond

Multimodal Code Intelligence

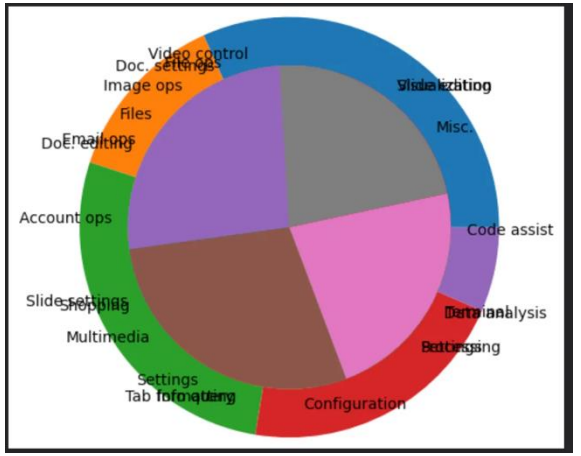
These capabilities are cool,
but they often require specialized coding skills and multimodal capability.

Challenge 2: General-purpose open-source VLMs perform poorly on code + visual tasks.

User: Please help me replicate….



User: 😓



Multimodal Code Intelligence

Challenge 3: Existing models, whether single or multimodal, are task-specific and lack generalization ability.

General models are fine-tuned to enable specific python visual, web-UI, or interactive content generation

This leads to the value and user experience differ greatly from proprietary ones.

Pure Text-based

VisCoder: Fine-Tuning LLMs for Executable Python Visualization Code Generation

Yuansheng Ni[1], Ping Nie[4], Kai Zou[3], Xiang Yue[2], Wenhu Chen[1],
[1]University of Waterloo, [2]Carnegie Mellon University, [3]Netmind.ai, [4]Independent Researcher,
{yuansheng.ni, wenhuchen}@uwaterloo.ca
https://tiger-ai-lab.github.io/VisCoder

Multimodal

ChartCoder: Advancing Multimodal Large Language Model for Chart-to-Code Generation

Xuanle Zhao[1,*], Xianzhen Luo[2,*], Qi Shi[1,†], Chi Chen[1,†],
Shuo Wang[1], Zhiyuan Liu[1], Maosong Sun[1]
[1] Tsinghua University, Beijing, China
[2] Harbin Institute of Technology, Harbin, China
2429527z@gmail.com, xzluo@ir.hit.edu.cn

VisCodex: Unified Multimodal Code Generation via Merging Vision and Coding Models

Lingjie Jiang[1,2] Shaohan Huang[1†] Xun Wu[1] Yixia Li[1,3] Dongdong Zhang[1] Furu Wei[1]
[1] Microsoft Research [2] Peking University [3] Southern University of Science and Technology
https://aka.ms/GeneralAI

Why Do We Need JanusCoder?

Landscape:

1. Research on such capabilities is still in its early stage, yet highly valuable (*e.g.*, for scientific demonstrations).

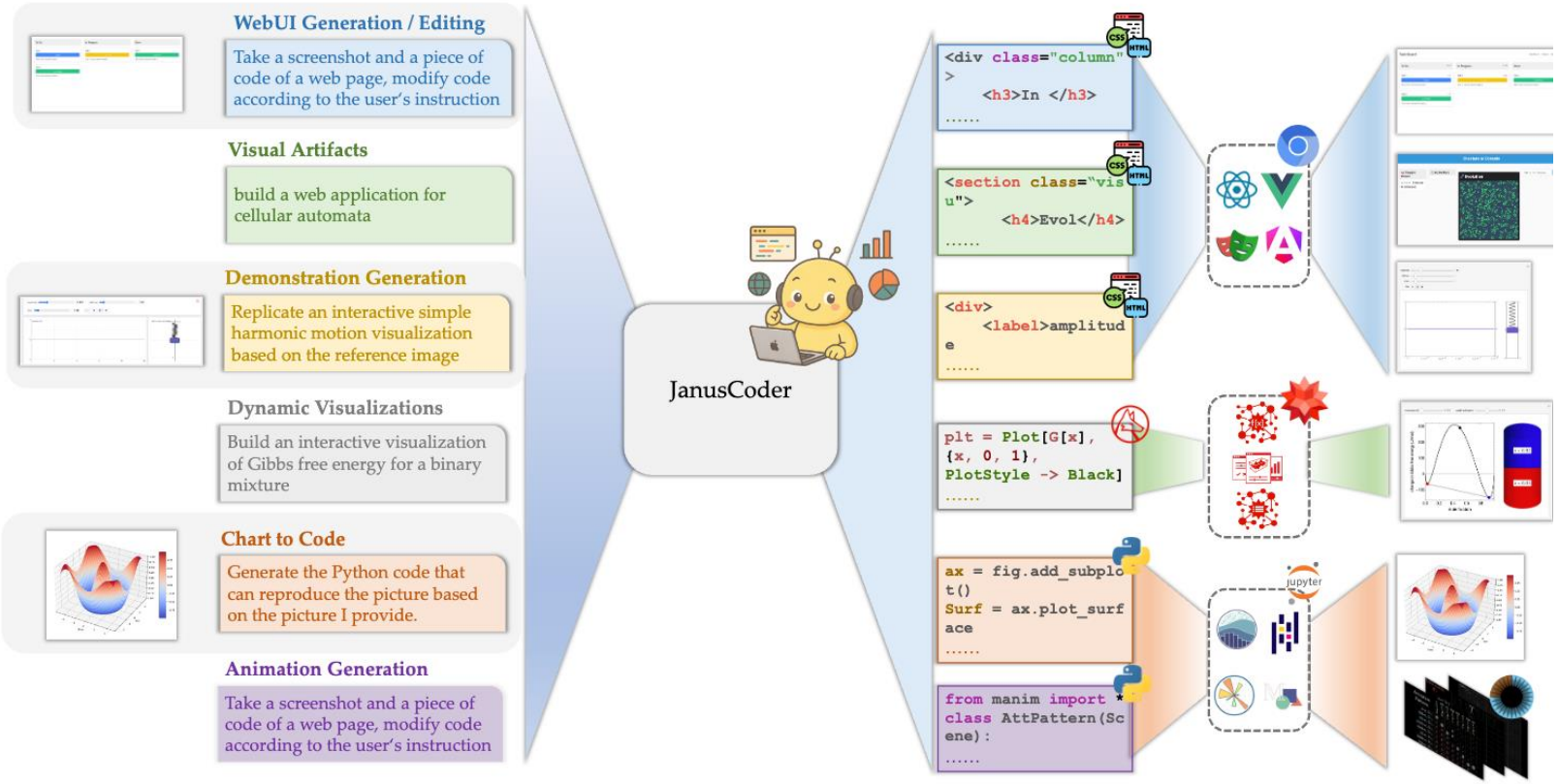
2. Proprietary models come with high costs and restricted research accessibility.

3. Existing works often built specialized models for isolated tasks (*e.g.*, one for charts, another for web UIs), resulting in limited generalization and poor scalability.

Therefore, we need a **unified, open-source Visual–Programmatic Interface** to **advance research in this domain**.

JanusCoder

We built the first suite of unified Visual–Programmatic Foundation Model.



JanusCoder

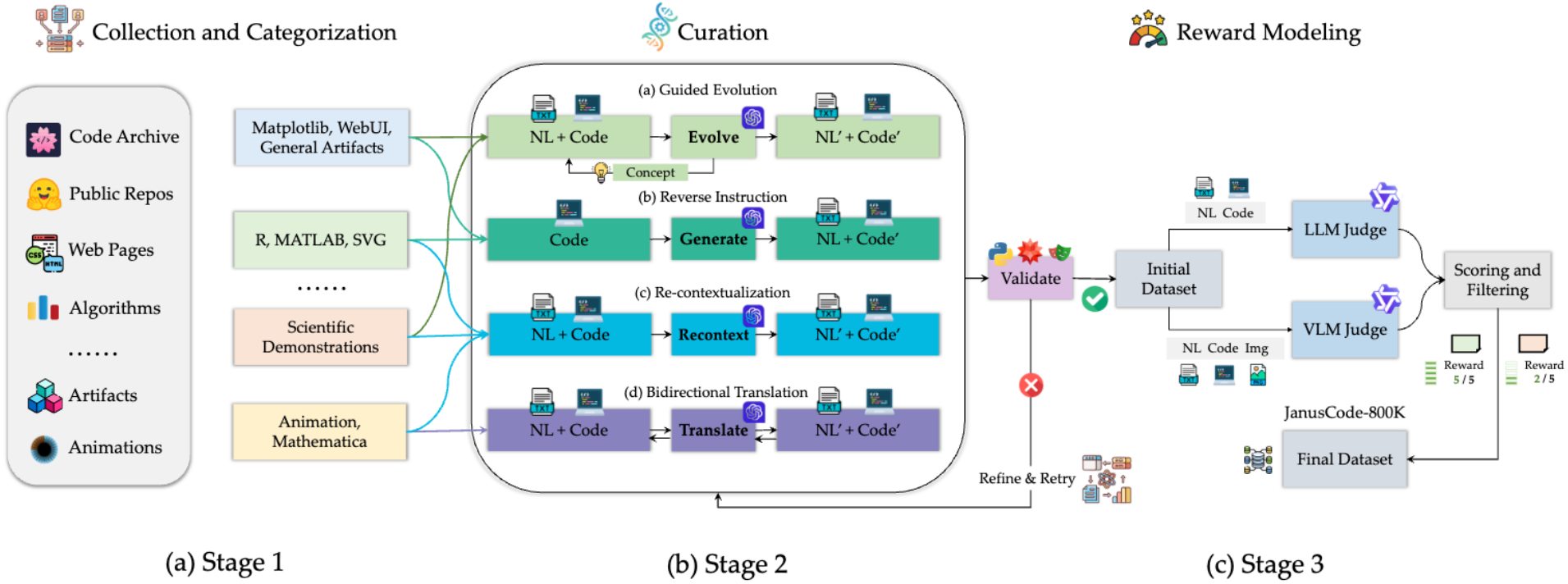
The Core Problem: Progress in this domain has been blocked by the scarcity of high-quality, large-scale, and diverse data.

We need to cover multiple data types, but face several challenges:
(1) High-quality + Labeled data is scarce,
(2) Category imbalance is severe, and
(3) Demo and artifact data must be collected, cleaned, and constructed from scratch.

In this work, we try to solve the Data Bottleneck at its Source

JanusCoder

A comprehensive multimodal code data synthesis pipeline.



Rewarding: deterministic signals (compiler, renderer, etc.) + VLM/LLM-as-a-Judge evaluation.

JanusCoder

For the same set of seed data, we **adopt diverse, adaptive strategies** to **synthesize data accordingly**.

Table 1: Overview of the strategies used to construct JanusCode Data from multiple sources, different colored squares represent different strategies: Guided Evolution; Re-contextualization; Reverse Instruction; Bidirectional Translation.

Source Data Type	Size	Validation	Reward	Strategies
Matplotlib	200K	Python	VLM	
Charts	77K	Python	VLM	
Algorithm	100K	Python	VLM	
Mathematica	11K	Wolfram Engine	LLM	
Animation	5K	Python + Manim Engine	VLM	
Scientific PLs	400K	-	LLM	
SVG	400K	-	VLM	
WebUI	270K	Playwright	VLM	
General Artifacts	10K	Playwright	VLM	
Scientific demonstration	10K	Playwright	VLM	

With the scale and diversity of data, we further explore the synergistic effects across different data types.

JanusCoder

The largest dataset to date for coding-driven visual content generation.

Table 2: Statistics of JANUSCODE-800K.

Data Type	Statistics
Text-centric	
Python Visualization: Generation	127.5K
Python Visualization: Editing	51.8K
Scientific PLs	31.8K
SVG	20.0K
Animation	19.5K
General Artifacts	56.8K
Algorithm Data	100.0K
Vision-centric	
Chart-to-Code	70.0K
WebUI Generation	200.0K
WebUI Editing	69.5K
Scientific demonstration	53.0K

Figure 3: Distribution of JANUSCODE-800K.

JanusCoder

The largest dataset to date for coding-driven visual content generation.

Table 2: Statistics of JANUSCODE-800K.

Data Type	Statistics
Text-centric	
Python Visualization: Generation	127.5K
Python Visualization: Editing	51.8K
Scientific PLs	31.8K
SVG	20.0K
Animation	19.5K
General Artifacts	56.8K
Algorithm Data	100.0K
Vision-centric	
Chart-to-Code	70.0K
WebUI Generation	200.0K
WebUI Editing	69.5K
Scientific demonstration	53.0K

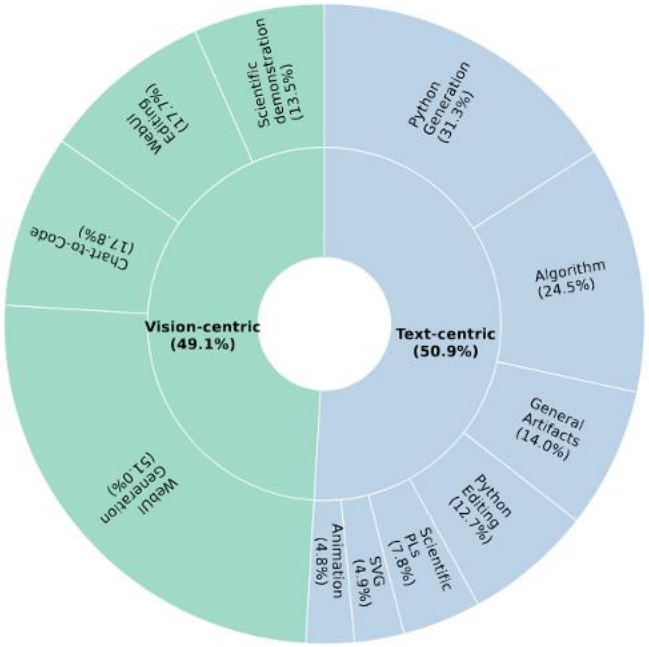


Figure 3: Distribution of JANUSCODE-800K.

It also includes **data types never seen in previous works**.

JanusCoder

It features (1) widest range of tasks, (2) largest data scale, and (3) broadest PL coverage.

Figure 3: Distribution of JANUSCODE-800K.

Competitors:

1. VisCode-200K: Python visualization only, single-modal *(Ni et al, 2025)*
2. MCD dataset: Traditional HTML, chart, QA tasks, 598K *(Jiang et al, 2025)*

This enables us to build a model that supports all major single- and multimodal code-based visual content generation.

JanusCoder

We present a suite of models to advance multimodal code intelligence.

JanusCoder-8B / 14B: based on Qwen3-8B / 14B

JanusCoderV-7B / 8B: based on Qwen2.5-VL-7B and InternVL3.5-8B

We further validate the effectiveness of JanusCode-800K through experiments across backbones.

I plan to replace JanusCoderV's backbone with Qwen3-VL soon.

JanusCoder

Table 3: Results on PandasPlotBench, ArtifactsBench, and DTVBENCH.

Model	PandasPlotBench			ArtifactsBench	DTVBENCH	
	Incorrect Code↓ (%)	Visual	Task		Manim	Wolfram
	Open-Source					
LLaMA3-8B-Instruct	26.9	59	69	36.5	4.92	3.15
Qwen3-8B	20.0	63	74	36.5	6.20	5.18
Qwen2.5-Coder-7B-Ins	21.1	63	76	26.0	8.56	4.04
Qwen3-14B	12.6	65	78	39.8	6.63	5.08
Qwen2.5-Coder-32B-Ins	12.0	66	82	35.5	9.61	4.98
JANUSCODER-8B	14.9	63	80	39.6	**9.70**	**6.07**
JANUSCODER-14B	**9.7**	**67**	**86**	**41.1**	8.41	5.97
	Proprietary					
GPT-4o	9.7	72	85	37.9	10.60	4.92

Let's start with the most typical case: Python visualization.

JanusCoder shows strong performance,

significantly outperforming open-source models of similar or larger size in code acc. and task completion

with some results matching or surpassing GPT-4o.

JanusCoder

Table 3: Results on PandasPlotBench, ArtifactsBench, and DTVBENCH.

Model	PandasPlotBench			ArtifactsBench	DTVBENCH	
	Incorrect Code↓ (%)	Visual	Task		Manim	Wolfram
Open-Source						
LLaMA3-8B-Instruct	26.9	59	69	36.5	4.92	3.15
Qwen3-8B	20.0	63	74	36.5	6.20	5.18
Qwen2.5-Coder-7B-Ins	21.1	63	76	26.0	8.56	4.04
Qwen3-14B	12.6	65	78	39.8	6.63	5.08
Qwen2.5-Coder-32B-Ins	12.0	66	82	35.5	9.61	4.98
JANUSCODER-8B	14.9	63	80	39.6	**9.70**	**6.07**
JANUSCODER-14B	**9.7**	**67**	**86**	**41.1**	8.41	5.97
Proprietary						
GPT-4o	9.7	72	85	37.9	10.60	4.92

Strong results on the high-difficulty artifacts generation

Outperforming open-source baselines;

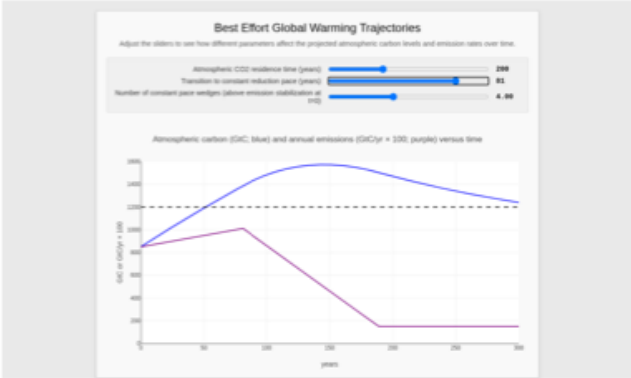
Application: Scientific visualization and demonstration

It can serve as a viable alternative to proprietary models,
Let's look at a few examples.

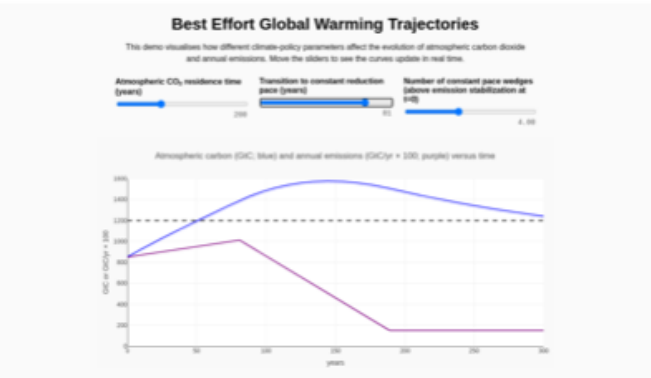
JanusCoder

JanusCoder v.s. Gemini 2.5 Pro

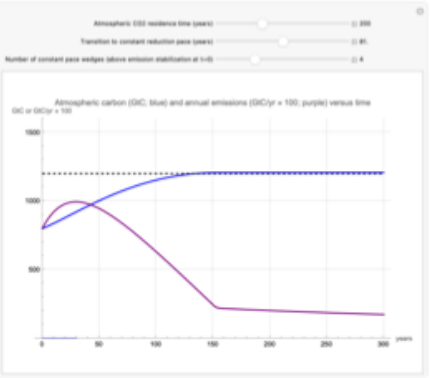
This demo visualises how different climate-policy parameters affect the evolution of atmospheric carbon dioxide and annual emissions. Move the sliders to see the curves update in real time.



Gemini-2.5-Pro

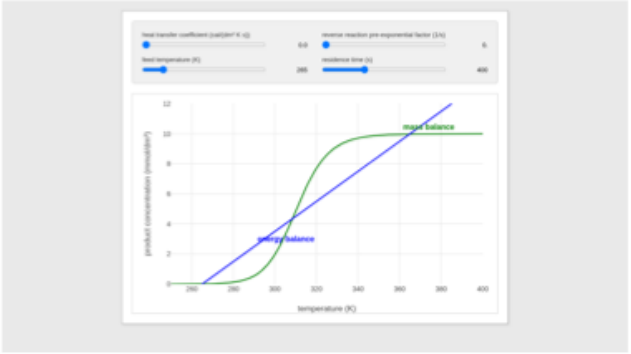


JanusCoder-8B

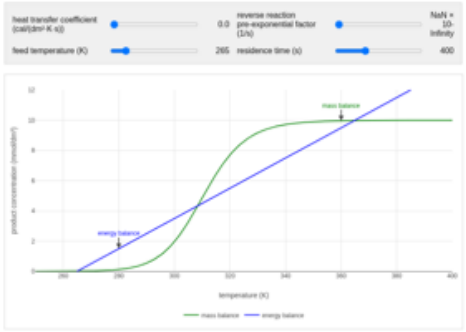


References

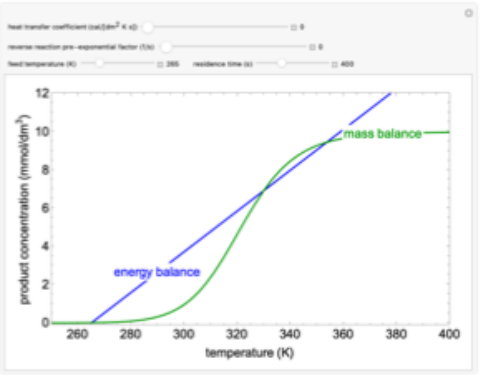
Multiple Steady States in a Continuously Stirred Tank Reactor



Gemini-2.5-Pro



JanusCoder-8B



References

JanusCoder

Table 3: Results on PandasPlotBench, ArtifactsBench, and DTVBENCH.

Model	PandasPlotBench			ArtifactsBench	DTVBENCH	
	Incorrect Code↓ (%)	Visual	Task		Manim	Wolfram
Open-Source						
LLaMA3-8B-Instruct	26.9	59	69	36.5	4.92	3.15
Qwen3-8B	20.0	63	74	36.5	6.20	5.18
Qwen2.5-Coder-7B-Ins	21.1	63	76	26.0	8.56	4.04
Qwen3-14B	12.6	65	78	39.8	6.63	5.08
Qwen2.5-Coder-32B-Ins	12.0	66	82	35.5	9.61	4.98
JANUSCODER-8B	14.9	63	80	39.6	**9.70**	**6.07**
JANUSCODER-14B	**9.7**	**67**	**86**	**41.1**	8.41	5.97
Proprietary						
GPT-4o	9.7	72	85	37.9	10.60	4.92

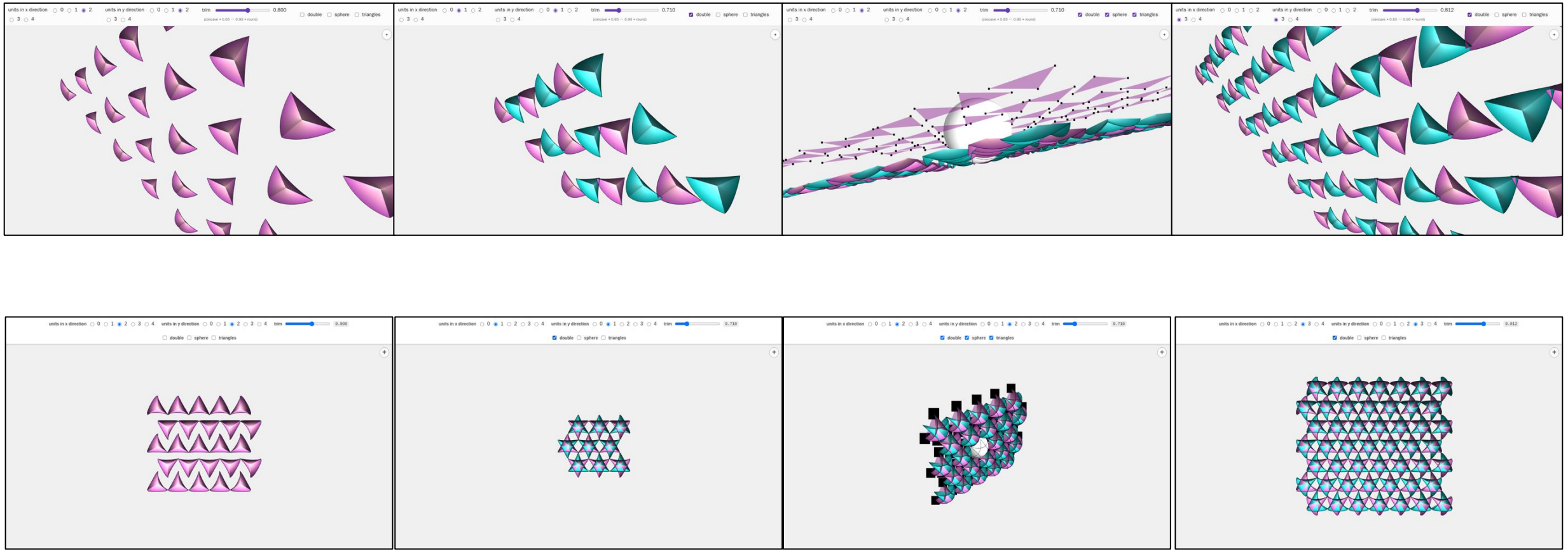
On the most challenging dynamic visualization tasks,

The performance is comparable to proprietary models.

Application: Creating 3Blue1Brown-style explanatory videos

JanusCoder

wolfram



JanusCoder

Manim

Helps create explanatory videos for scientific and educational subjects.

JanusCoderV

Table 4: Results on ChartMimic, DesignBench, WebCode2M, and InteractScience.

Model	ChartMimic				DesignBench		WebCode2M		InteractScience		
	Customized		Direct		Gen.	Edit.	Visual	TreeBLEU	Func.	Visual	
	Low	High	Low	High					Overall	CLIP	VLM
Open-Source											
Qwen2.5-VL-7B-Ins	51.07	58.69	40.73	41.70	72.73	6.85	73.42	12.83	8.40%	45.86	19.83
InternVL3-8B	51.88	60.04	48.48	55.41	69.34	7.76	**79.62**	12.40	8.93%	53.35	22.05
InternVL3.5-8B	51.56	59.55	46.02	53.39	71.73	8.63	79.09	11.95	11.47%	56.79	24.17
MiniCPM-V-2-6	27.53	48.18	21.82	45.26	66.25	4.56	45.85	9.73	0.13%	20.65	7.70
Llama-3.2-11B-Vision-Ins	18.87	39.63	19.32	28.37	62.24	6.61	51.54	6.57	6.67%	32.87	13.24
JANUSCODERV-7B	64.72	72.77	65.73	72.73	**73.31**	**8.79**	75.78	**26.21**	**17.73%**	60.56	27.67
JANUSCODERV-8B	**66.68**	**74.20**	**65.79**	**73.18**	68.86	8.63	66.34	18.28	17.60%	**61.52**	**33.32**
Proprietary											
GPT-4o	59.4	67.42	57.16	64.62	76.83	9.23	82.67	13.00	27.20%	70.14	46.01

JanusCoderV largely outperforms proprietary models on the chart-to-code task.

JanusCoderV

Table 4: Results on ChartMimic, DesignBench, WebCode2M, and InteractScience.

Model	ChartMimic				DesignBench		WebCode2M		InteractScience		
	Customized		Direct		Gen.	Edit.	Visual	TreeBLEU	Func.	Visual	
	Low	High	Low	High					Overall	CLIP	VLM
Open-Source											
Qwen2.5-VL-7B-Ins	51.07	58.69	40.73	41.70	72.73	6.85	73.42	12.83	8.40%	45.86	19.83
InternVL3-8B	51.88	60.04	48.48	55.41	69.34	7.76	**79.62**	12.40	8.93%	53.35	22.05
InternVL3.5-8B	51.56	59.55	46.02	53.39	71.73	8.63	79.09	11.95	11.47%	56.79	24.17
MiniCPM-V-2-6	27.53	48.18	21.82	45.26	66.25	4.56	45.85	9.73	0.13%	20.65	7.70
Llama-3.2-11B-Vision-Ins	18.87	39.63	19.32	28.37	62.24	6.61	51.54	6.57	6.67%	32.87	13.24
JanusCoderV-7B	64.72	72.77	65.73	72.73	**73.31**	**8.79**	75.78	**26.21**	**17.73%**	60.56	27.67
JanusCoderV-8B	**66.68**	**74.20**	**65.79**	**73.18**	68.86	8.63	66.34	18.28	17.60%	**61.52**	**33.32**
Proprietary											
GPT-4o	59.4	67.42	57.16	64.62	76.83	9.23	82.67	13.00	27.20%	70.14	46.01

On web UI tasks, it performs strongly in both in-domain and out-of-domain settings.

The web UI generation and editing capabilities are closely related to its strength in scientific demonstrations.

JanusCoderV

Table 4: Results on ChartMimic, DesignBench, WebCode2M, and InteractScience.

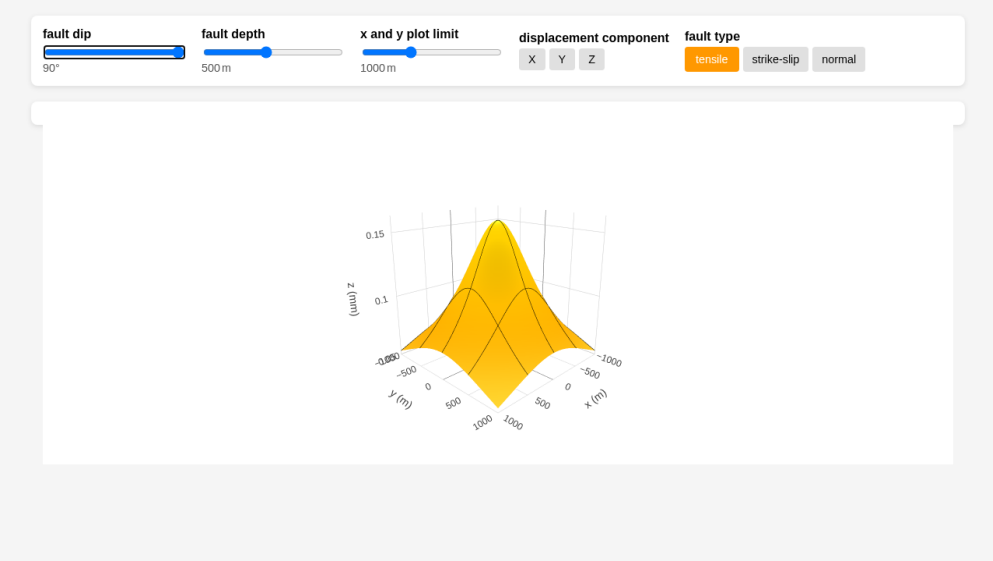
Model	ChartMimic				DesignBench		WebCode2M		InteractScience		
	Customized		Direct		Gen.	Edit.	Visual	TreeBLEU	Func.	Visual	
	Low	High	Low	High					Overall	CLIP	VLM
Open-Source											
Qwen2.5-VL-7B-Ins	51.07	58.69	40.73	41.70	72.73	6.85	73.42	12.83	8.40%	45.86	19.83
InternVL3-8B	51.88	60.04	48.48	55.41	69.34	7.76	**79.62**	12.40	8.93%	53.35	22.05
InternVL3.5-8B	51.56	59.55	46.02	53.39	71.73	8.63	79.09	11.95	11.47%	56.79	24.17
MiniCPM-V-2-6	27.53	48.18	21.82	45.26	66.25	4.56	45.85	9.73	0.13%	20.65	7.70
Llama-3.2-11B-Vision-Ins	18.87	39.63	19.32	28.37	62.24	6.61	51.54	6.57	6.67%	32.87	13.24
JanusCoderV-7B	64.72	72.77	65.73	72.73	**73.31**	**8.79**	75.78	**26.21**	**17.73%**	60.56	27.67
JanusCoderV-8B	**66.68**	**74.20**	**65.79**	**73.18**	68.86	8.63	66.34	18.28	17.60%	**61.52**	**33.32**
Proprietary											
GPT-4o	59.4	67.42	57.16	64.62	76.83	9.23	82.67	13.00	27.20%	70.14	46.01

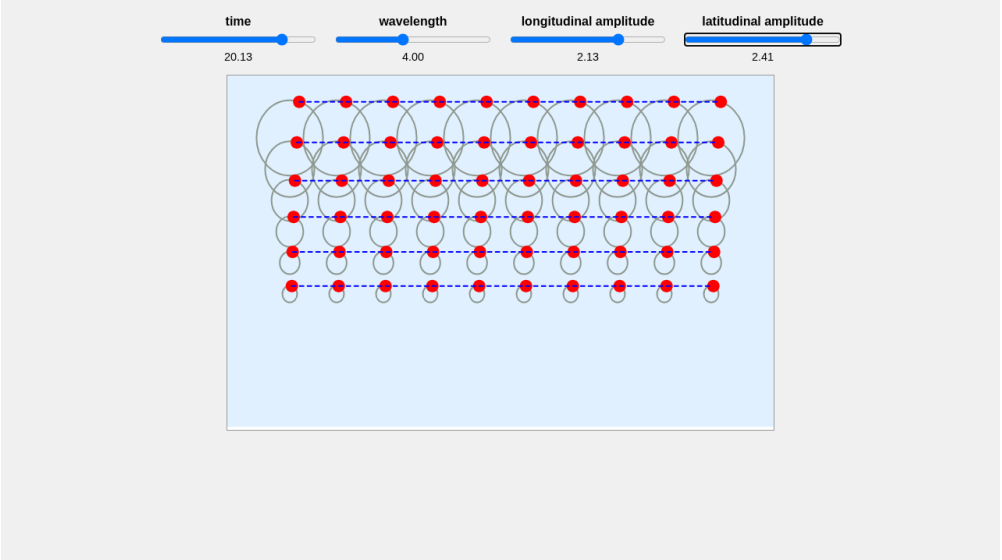
Exhibits strong multimodal scientific demo generation,
achieving performance close to proprietary models.

Can extend the application scope of scientific multimodal foundation models.

JanusCoderV

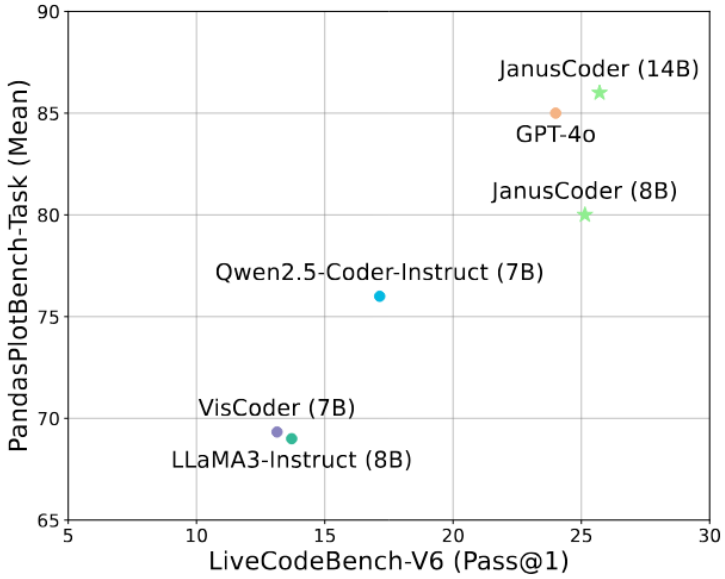
Given a sketch, it can generate interactive scientific demonstrations.

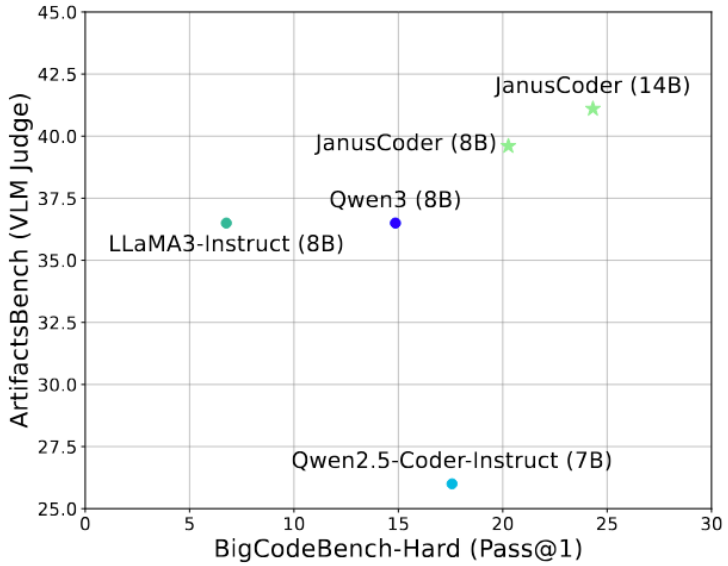




JanusCoder

While achieving strong visual content generation, JanusCoder also retains good general coding capability.





Potential: Multimodal **Programming + Visualization Assistant**

JanusCoder

Our unified data generalize well :)

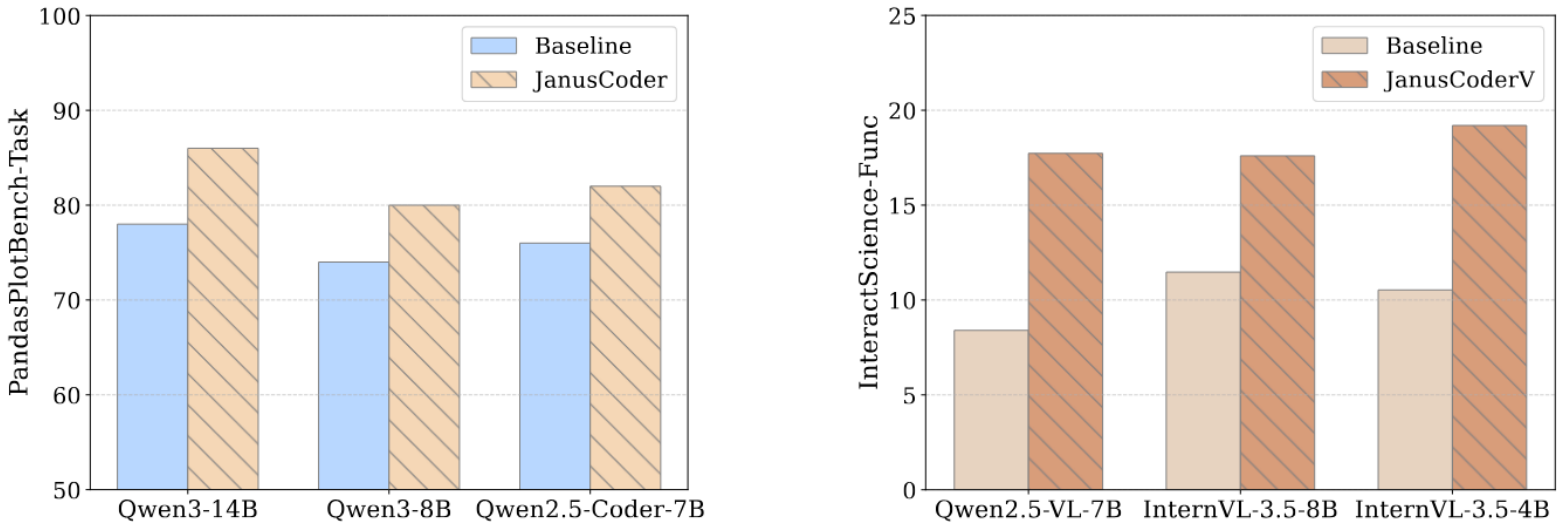


Figure 4: Effectiveness on different model backbones

Performs strongly across **different backbones**, suggesting the **data synthesis pipeline is generalizable and reusable**.

Thanks for listening

Contact: qiushisun@connect.hku.hk