Thematic session

Paper presentation: Group4

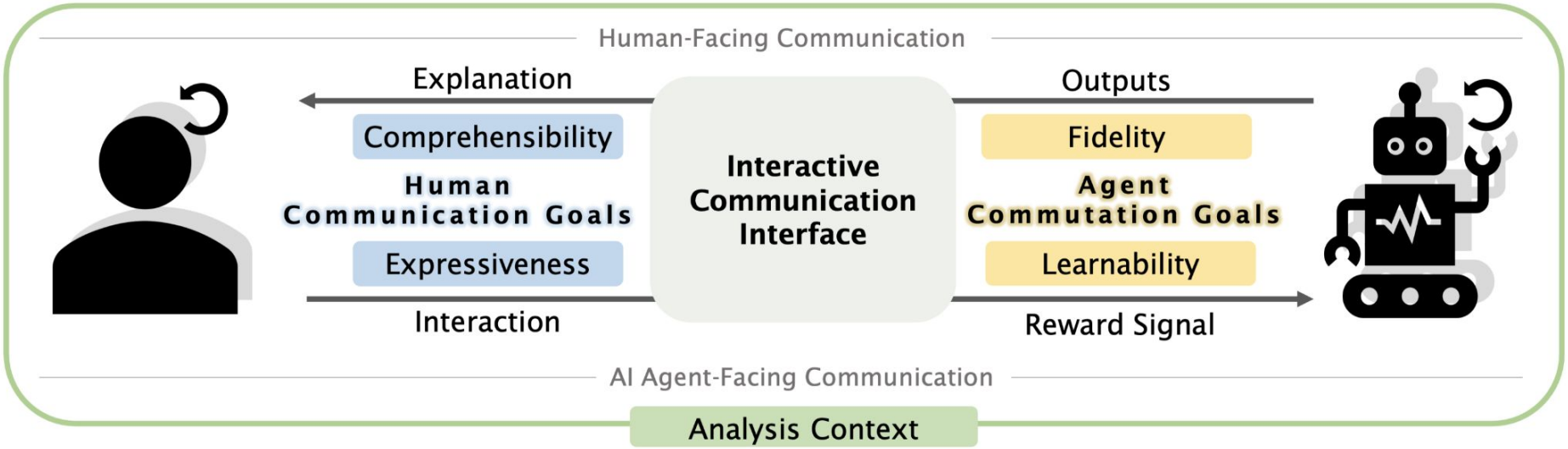
Marcel Worring

Mapping out the Space of Human Feedback for Reinforcement Learning: A Conceptual Framework

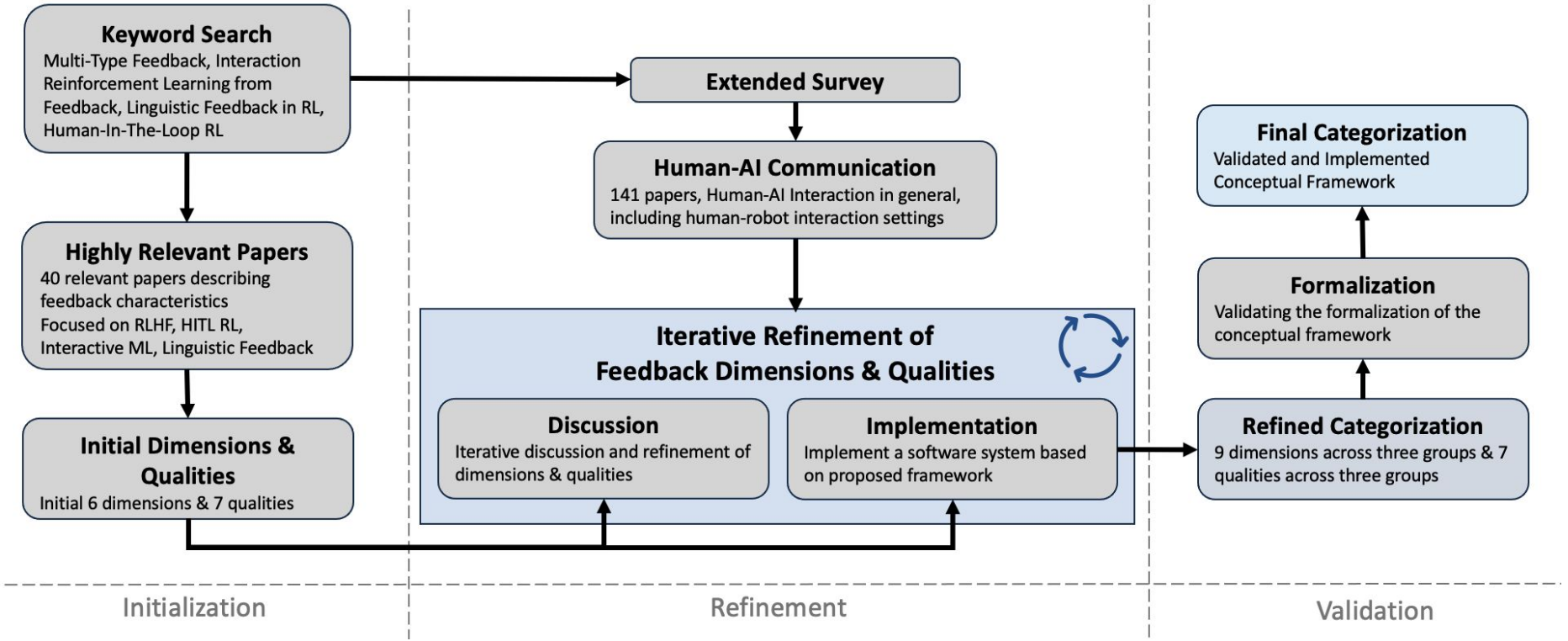
YANNICK METZ, University of Konstanz, Germany and ETH Zurich, Switzerland
DAVID LINDNER, ETH Zurich, Switzerland
RAPHAËL BAUR, ETH Zurich, Switzerland
MENNATALLAH EL-ASSADY, ETH Zurich, Switzerland

arXiv:2411.11761v2 [cs.LG] 20 Feb 2025

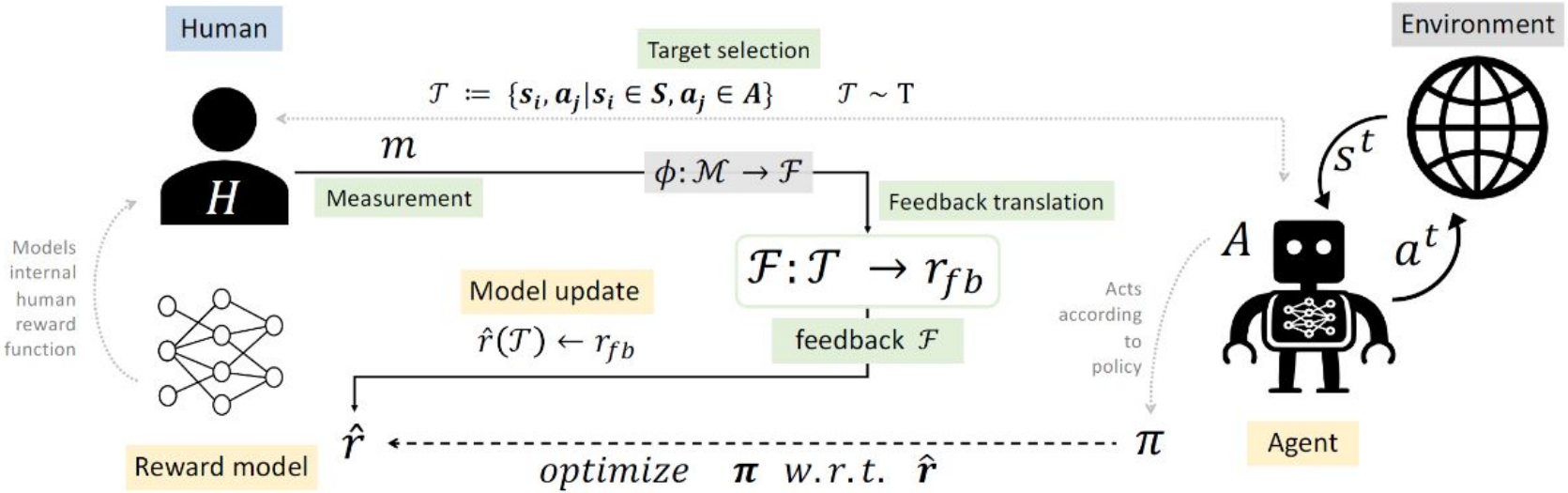
Human-AI Interaction



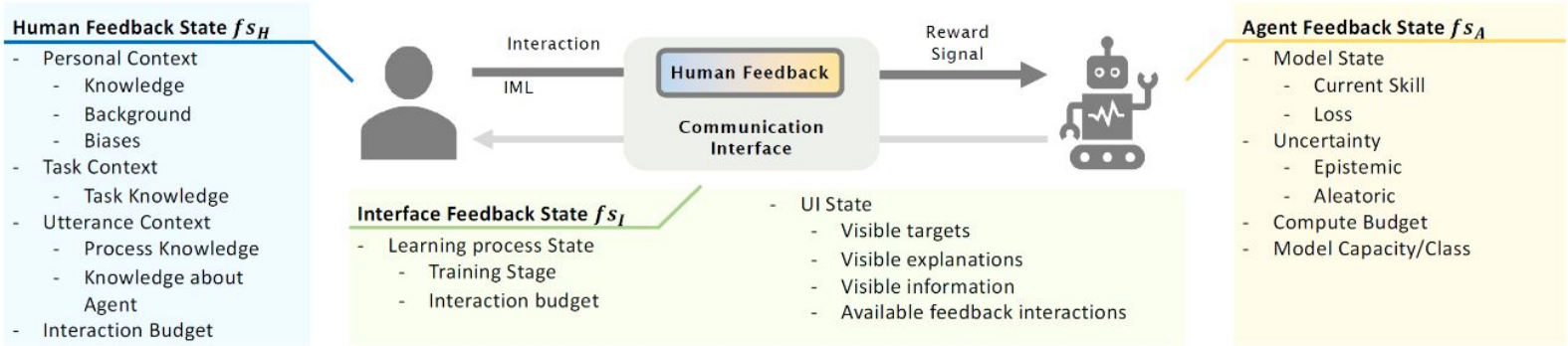
Creating a Review



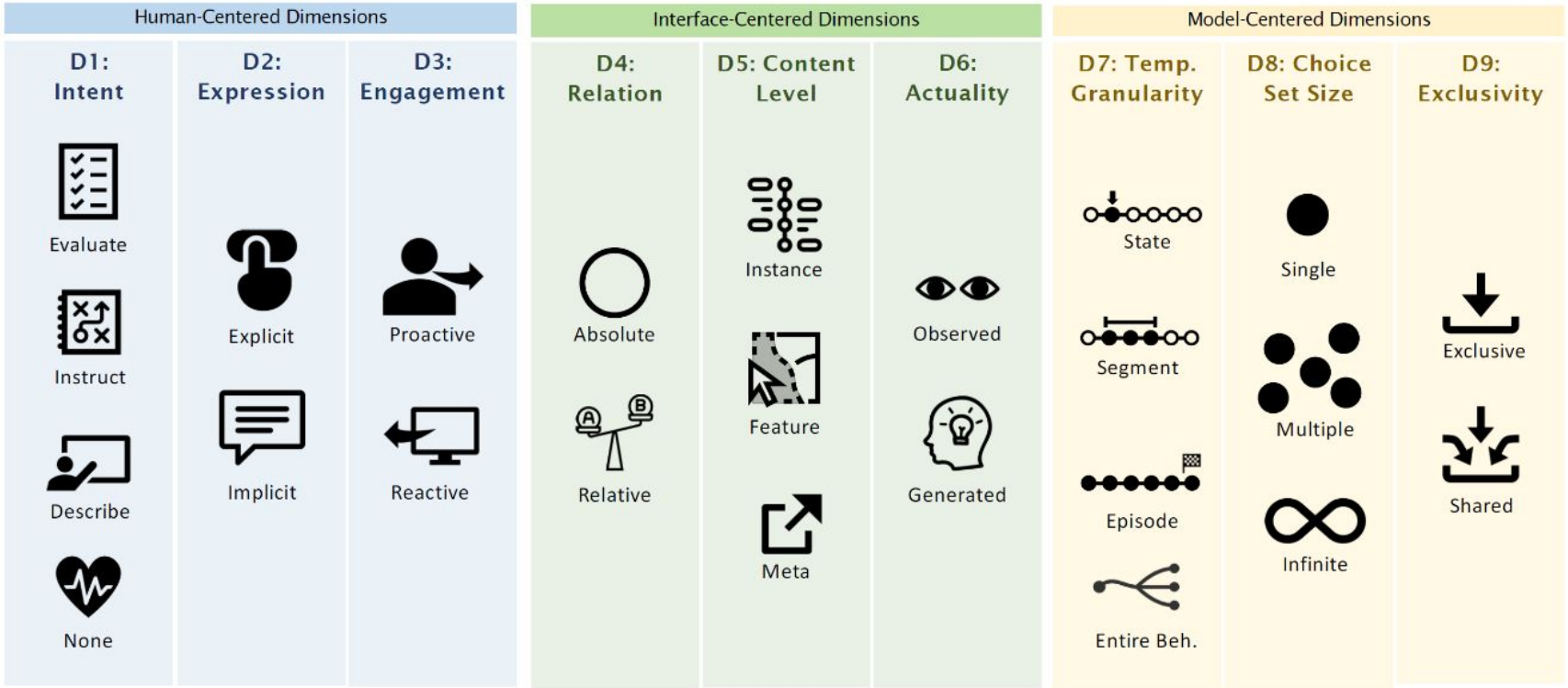
A more formal model



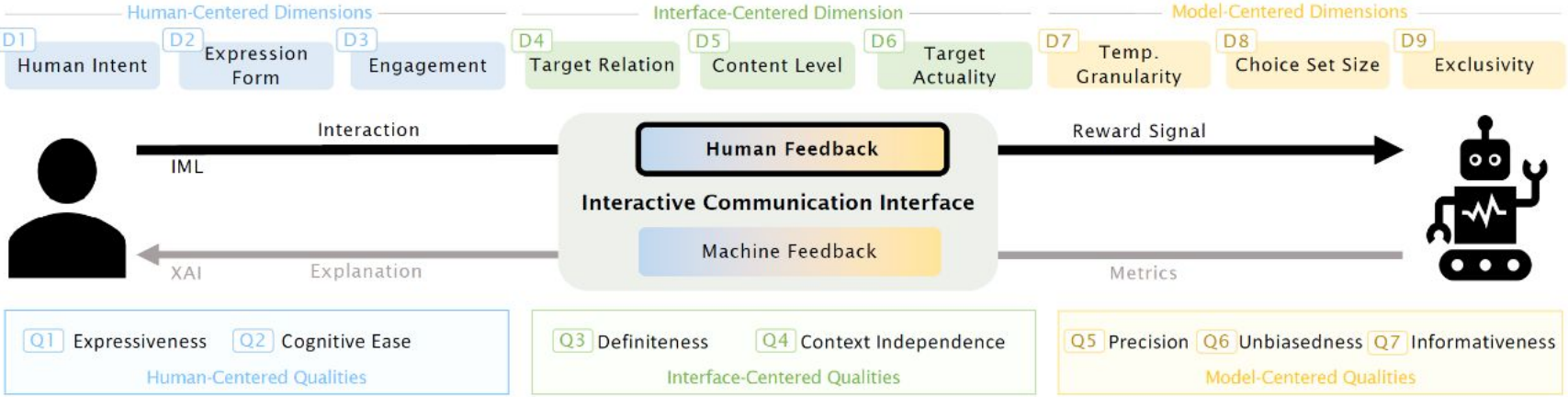
Different State Spaces



Feedback Dimensions



Summary


Human-Centered Dimensions
Interface-Centered Dimension
Model-Centered Dimensions
D1 Human Intent
D2 Expression Form
D3 Engagement
D4 Target Relation
D5 Content Level
D6 Target Actuality
D7 Temp. Granularity
D8 Choice Set Size
D9 Exclusivity
Interaction
IML
Human Feedback
Interactive Communication Interface
Machine Feedback
Reward Signal
XAI
Explanation
Metrics
Q1 Expressiveness
Q2 Cognitive Ease
Human-Centered Qualities
Q3 Definiteness
Q4 Context Independence
Interface-Centered Qualities
Q5 Precision
Q6 Unbiasedness
Q7 Informativeness
Model-Centered Qualities

Shuai Wang

Published as a conference paper at ICLR 2025

ColPali: Efficient Document Retrieval with Vision Language Models

Manuel Faysse[*1,3] **Hugues Sibille**[*1,4] **Tony Wu**[*1] **Bilel Omrani**[1]
Gautier Viaud[1] **Céline Hudelot**[3] **Pierre Colombo**[2,3]
[1]Illuin Technology [2]Equall.ai [3]CentraleSupélec, Paris-Saclay [4]ETH Zürich
manuel.faysse@centralesupelec.fr

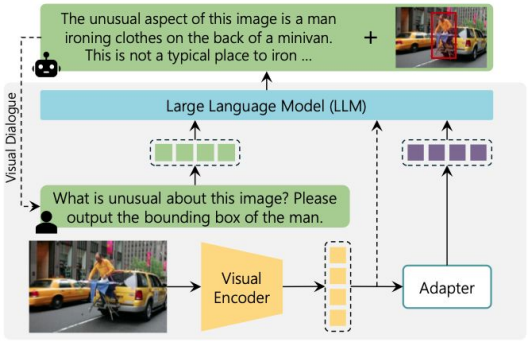
Faysse, M., Sibille, H., Wu, T., Omrani, B., Viaud, G., Hudelot, C., & Colombo, P. (2024). ColPali: Efficient Document Retrieval with Vision Language Models. *ArXiv, abs/2407.01449*.

Background –Multimodal Large Language Models (MLLMs) vs. Large Language Models (LLMs)

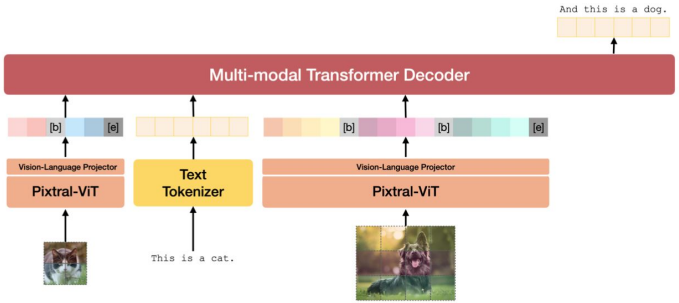
MLLMs are a **superset** of **Large Language Models LLMs**:

✅ They have the **added ability** to process and understand **images, charts, tables, and figures**.

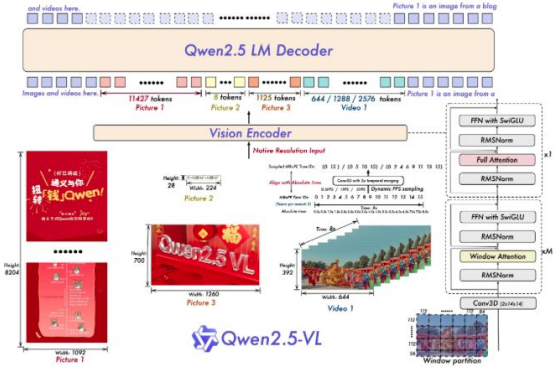
✅ They typically use an **LLM decoder as the backbone**, should retain **all text-based capabilities** (includes reasoning) of LLMs.



General architecture of MLLM



Pixtral 12B from 2024 Oct



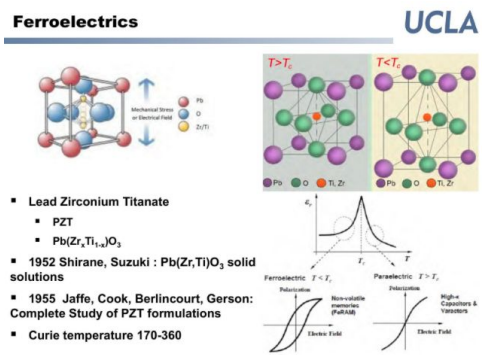
Qwen2.5 VL from 2025 Jan

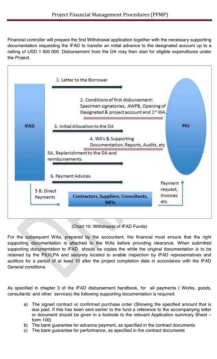
Why MLLMs for Document Understanding?

📄 **Documents are inherently multimodal**—they combine **text, charts, tables, and figures**, where LLM struggle to extract insights from.

🌍 **Documents represent knowledge across different domains with diverse formats** – Scientific papers, financial reports, legal contracts, and medical records…

🚀 **MLLMs unlock a new frontier** in **document retrieval, summarization, and comprehension** by handling all these modalities **simultaneously**.

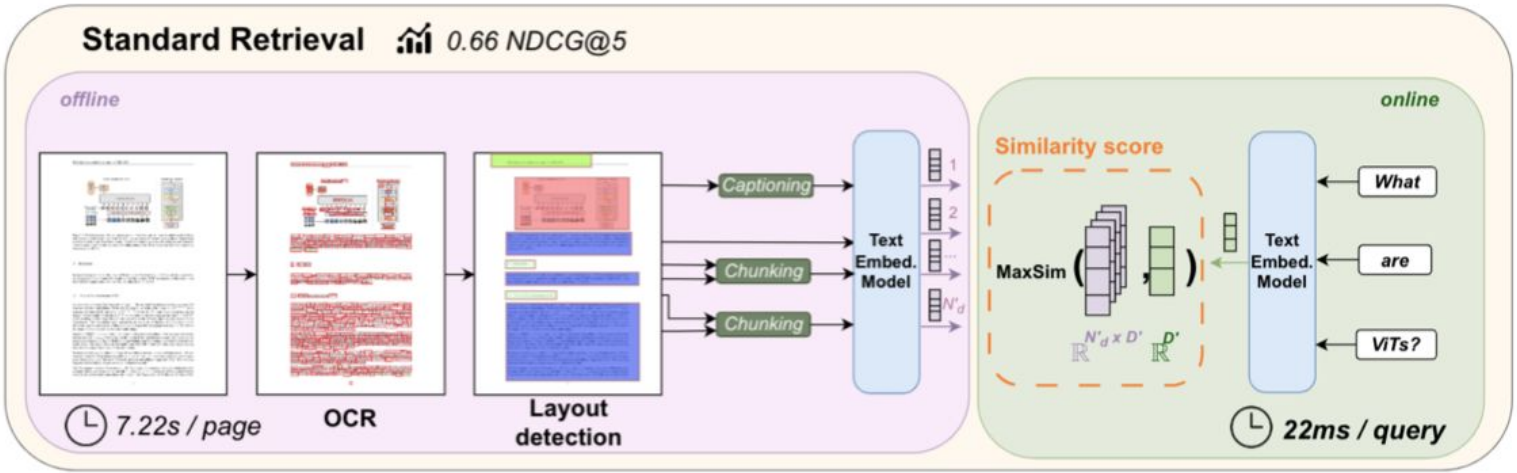




Scientific papers, slides, reports…

Main Paper Idea:

- Current document retrieval methods rely heavily on text extraction (OCR, parsing), **neglecting visual cues**.
- **ColPali** proposes a **vision-based retrieval model** that indexes document pages **directly from images** using **Vision-Language Models (VLMs)**.
- **Outcome:** ColPali is **faster, simpler, and more accurate** than conventional retrieval systems.



How ColPali Works?

Multi-Vector Vision Embeddings:

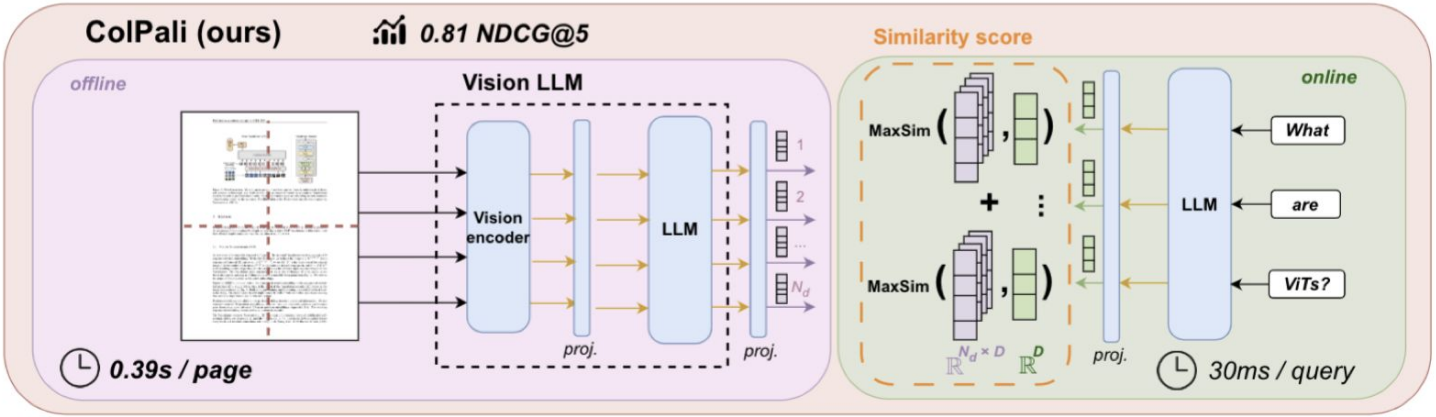
- Uses **PaliGemma-3B**, a Vision-Language Model, to encode images and query into **multi-vector embeddings**

Late Interaction Mechanism:

- Inspired by **ColBERT**, performs **fine-grained** matching between query and document embeddings.

End-to-End Training with Contrastive Learning:

- **Trained on 118K query-page pairs** (academic + synthetic datasets).



ViDoRe: A New Benchmark for Document Retrieval

Introduced in this paper to evaluate retrieval performance across:

- Different **document types** (academic, administrative, scientific).
- **Multiple modalities** (text, tables, infographics, figures).
- **Various languages** (English, French).
- **Two evaluation categories:**
  - **Academic Tasks:** Repurposed **VQA datasets** (e.g., DocVQA, InfoVQA, arXivQA).
  - **Practical Tasks:** Domain-specific retrieval benchmarks (e.g., government, healthcare, energy).

Dataset	Language	# Queries	# Documents	Description
Academic Tasks				
DocVQA	English	500	500	Scanned documents from UCSF Industry
InfoVQA	English	500	500	Infographics scrapped from the web
TAT-DQA	English	1600	1600	High-quality financial reports
arXiVQA	English	500	500	Scientific Figures from arXiv
TabFQuAD	French	210	210	Tables scrapped from the web
Practical Tasks				
Energy	English	100	1000	Documents about energy
Government	English	100	1000	Administrative documents
Healthcare	English	100	1000	Medical documents
AI	English	100	1000	Scientific documents related to AI
Shift Project	French	100	1000	Environmental reports

Table 1: *ViDoRe* comprehensively evaluates multimodal retrieval methods.

5 RESULTS

	ArxivQ	DocQ	InfoQ	TabF	TATQ	Shift	AI	Energy	Gov.	Health.	Avg.
Unstructured text-only											
- BM25	-	34.1	-	-	44.0	59.6	90.4	78.3	78.8	82.6	-
- BGE-M3	-	28.4↓5.7	-	-	36.1↓7.9	68.5↑8.9	88.4↓2.0	76.8↓1.5	77.7↓1.1	84.6↑2.0	-
Unstructured + OCR											
- BM25	31.6	36.8	62.9	46.5	62.7	64.3	92.8	85.9	83.9	87.2	65.5
- BGE-M3	31.4↓0.2	25.7↓11.1	60.1↓2.8	70.8↑24.3	50.5↓12.2	**73.2**↑8.9	90.2↓2.6	83.6↓2.3	84.9↑1.0	91.1↑3.9	66.1↑0.6
Unstructured + Captioning											
- BM25	40.1	38.4	70.0	35.4	61.5	60.9	88.0	84.7	82.7	89.2	65.1
- BGE-M3	35.7↓4.4	32.9↓5.4	71.9↑1.9	69.1↑33.7	43.8↓17.7	73.1↑12.2	88.8↑0.8	83.3↓1.4	80.4↓2.3	91.3↑2.1	67.0↑1.9
Contrastive VLMs											
Jina-CLIP	25.4	11.9	35.5	20.2	3.3	3.8	15.2	19.7	21.4	20.8	17.7
Nomic-vision	17.1	10.7	30.1	16.3	2.7	1.1	12.9	10.9	11.4	15.7	12.9
SigLIP (Vanilla)	43.2	30.3	64.1	58.1	26.2	18.7	62.5	65.7	66.1	79.1	51.4
Ours											
SigLIP (Vanilla)	43.2	30.3	64.1	58.1	26.2	18.7	62.5	65.7	66.1	79.1	51.4
BiSigLIP (+fine-tuning)	58.5↑15.3	32.9↑2.6	70.5↑6.4	62.7↑4.6	30.5↑4.3	26.5↑7.8	74.3↑11.8	73.7↑8.0	74.2↑8.1	82.3↑3.2	58.6↑7.2
BiPali (+LLM)	56.5↓-2.0	30.0↓-2.9	67.4↓-3.1	76.9↑14.2	33.4↑2.9	43.7↑17.2	71.2↓-3.1	61.9↓-11.7	73.8↓-0.4	73.6↓-8.8	58.8↑0.2
ColPali (+Late Inter.)	**79.1**↑22.6	**54.4**↑24.5	**81.8**↑14.4	**83.9**↑7.0	**65.8**↑32.4	**73.2**↑29.5	**96.2**↑25.0	**91.0**↑29.1	**92.7**↑18.9	**94.4**↑20.8	**81.3**↑22.5

Table 2: **Comprehensive evaluation of baseline models and our proposed method on *ViDoRe*.** Results are presented using nDCG@5 metrics, and illustrate the impact of different components. Text-only metrics are not computed for benchmarks with only visual elements.

Yijia Zheng

Mercury: Ultra-Fast Language Models Based on Diffusion

Inception Labs
Samar Khanna*, Siddhant Kharbanda*, Shufan Li*, Harshit Varma*, Eric Wang*
Sawyer Birnbaum^, Ziyang Luo^, Yanis Miraoui^, Akash Palrecha^
Stefano Ermon♯, Aditya Grover♯, Volodymyr Kuleshov♯
*^♯ equal core, cross-function, senior contributors listed alphabetically.
hello@inceptionlabs.ai

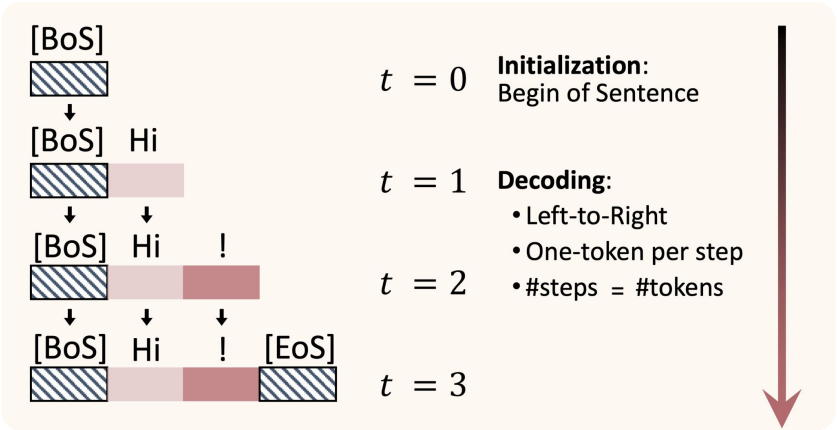
Autoregressive language models v.s. **diffusion language models** [1]

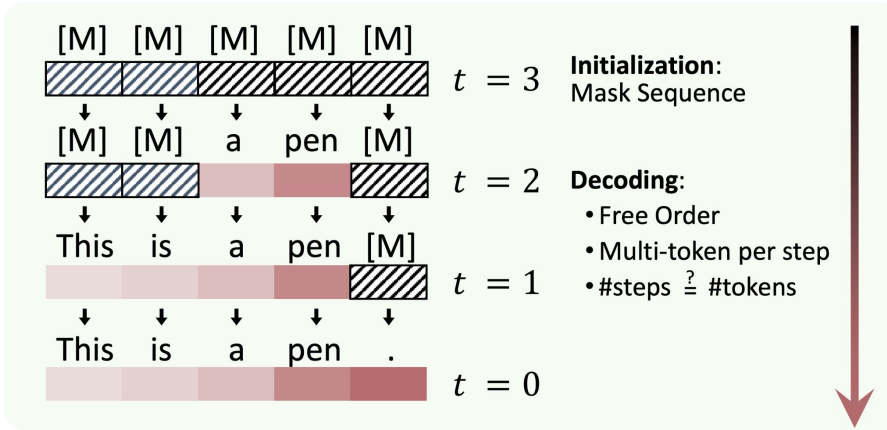
Autoregressive Language Models:

- Limited parallelism

Diffusion Language Models:

- Parallel Decoding
- Better Controllability
- Dynamic Perception

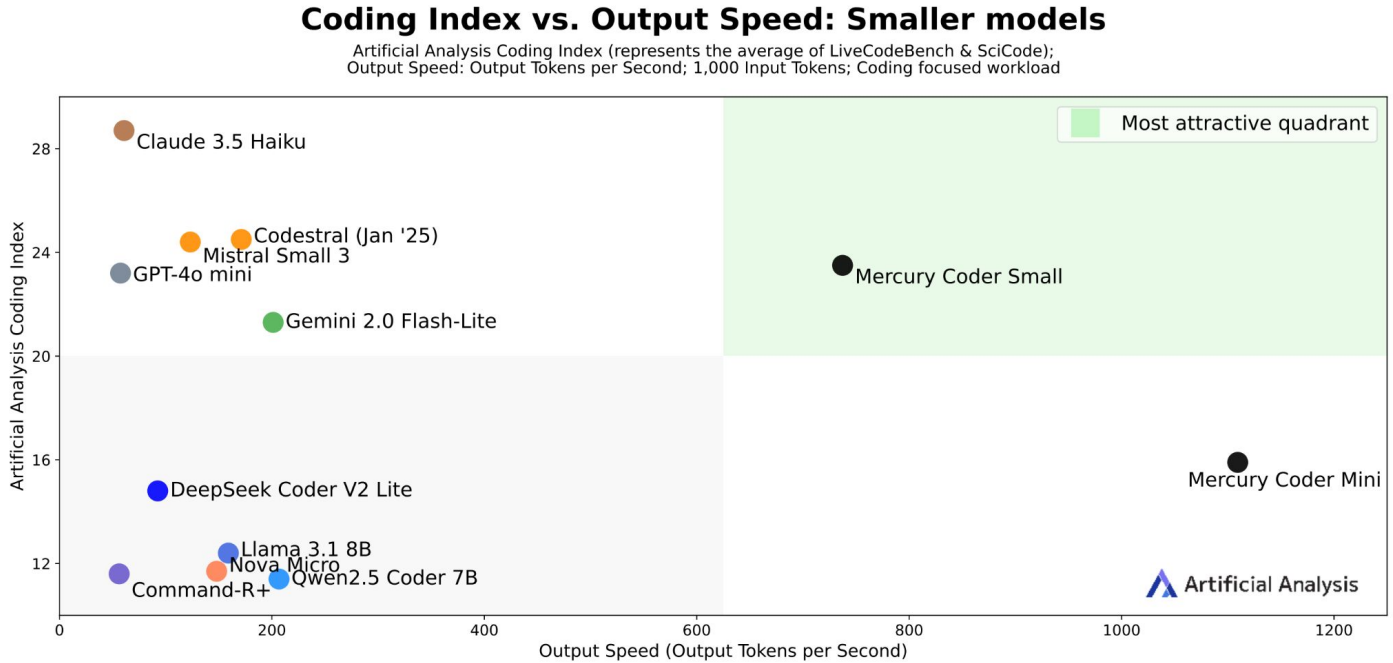




[1] Runpeng Yu, Qi Li, and Xinchao Wang. Discrete Diffusion in Large Language and Multimodal Models: A Survey. *arXiv:2506.13759*, 2025.

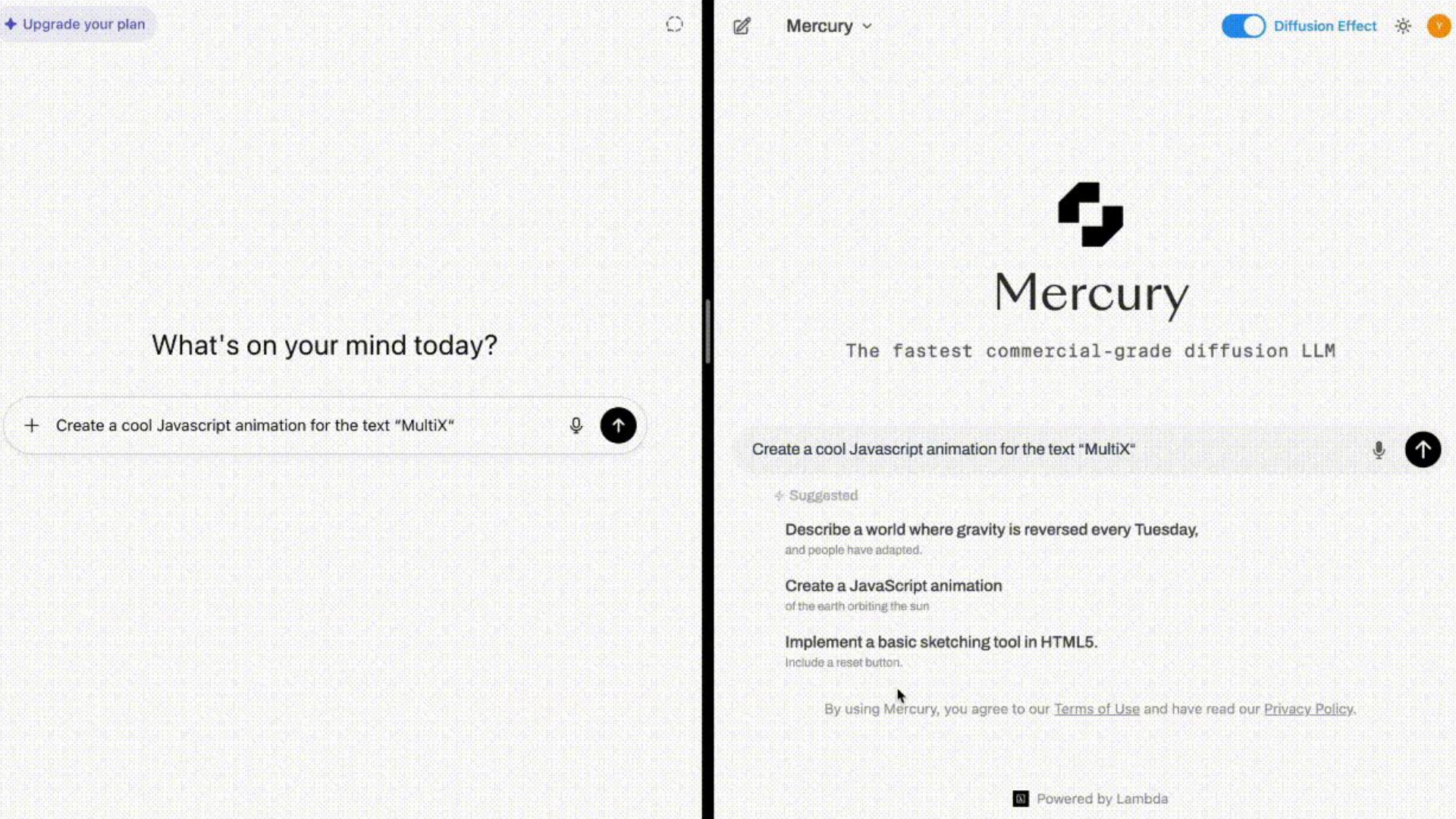
Mercury: Ultra-Fast Language Models Based on Diffusion [2]

Diffusion **Large** Language Model



https://artificialanalysis.ai/methodology/intelligence-benchmarking

[2] Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum et al. Mercury: Ultra-Fast Language Models Based on Diffusion. *arXiv:2506.17298*, 2025.


Upgrade your plan
What's on your mind today?
Create a cool Javascript animation for the text “MultiX“
Mercury
Diffusion Effect
Mercury
The fastest commercial-grade diffusion LLM
Create a cool Javascript animation for the text “MultiX“
Suggested
Describe a world where gravity is reversed every Tuesday,
and people have adapted.
Create a JavaScript animation
of the earth orbiting the sun
Implement a basic sketching tool in HTML5.
Include a reset button.
By using Mercury, you agree to our Terms of Use and have read our Privacy Policy.
Powered by Lambda

Gemini Diffusion

Our state-of-the-art, experimental text diffusion model

Join the waitlist

Yassin Mohamadi

NeurIPS 2023

Can Language Models Solve Graph Problems in Natural Language?

Heng Wang[*1], **Shangbin Feng**[*2], **Tianxing He**[2], **Zhaoxuan Tan**[3], **Xiaochuang Han**[2], **Yulia Tsvetkov**[2]
[1]Xi'an Jiaotong University [2]University of Washington [3]University of Notre Dame
wh2213210554@stu.xjtu.edu.cn, shangbin@cs.washington.edu

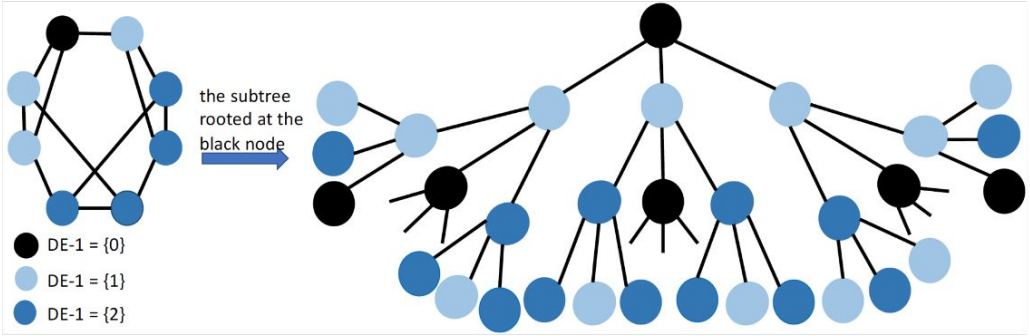
Intro

How can we obtain more accurate answers from a LLM?

How about Graphs?

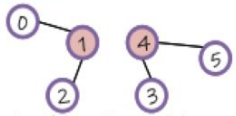
Can LLMs solve graph problems?

- COT: chain-of thought
- LTM: least-to-most
- SC: self-consistency



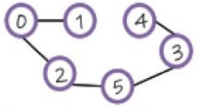
Can Language Models Solve Graph Problems in Natural Language?

1. Connectivity



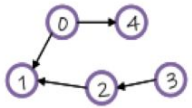
Determine if there is a path between two nodes in the graph. Note that (i,j) means that node i and node j are connected with an undirected edge.
Graph: (0,1) (1,2) (3,4) (4,5)
Q: Is there a path between node 1 and node 4?

2. Cycle



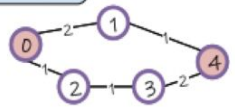
In an undirected graph, (i,j) means that node i and node j are connected with an undirected edge.
The nodes are numbered from 0 to 5, and the edges are: (3,4) (3,5) (1,0) (2,5) (2,0)
Q: Is there a cycle in this graph?

3. Topological Sort



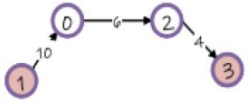
In a directed graph with 5 nodes numbered from 0 to 4:
node 0 should be visited before node 4, ...
Q: Can all the nodes be visited? Give the solution.

4. Shortest Path



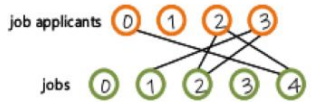
In an undirected graph, the nodes are numbered from 0 to 4, and the edges are: an edge between node 0 and node 1 with weight 2, ...
Q: Give the shortest path from node 0 to node 4.

5. Maximum Flow



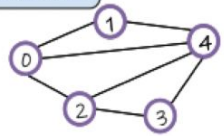
In a directed graph, the nodes are numbered from 0 to 3, and the edges are:
an edge from node 1 to node 0 with capacity 10,
an edge from node 0 to node 2 with capacity 6,
an edge from node 2 to node 3 with capacity 4.
Q: What is the maximum flow from node 1 to node 3?

6. Bipartite Graph Matching



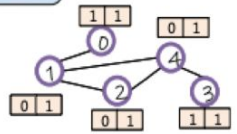
There are 4 job applicants numbered from 0 to 3, and 5 jobs numbered from 0 to 4. Each applicant is interested in some of the jobs. Each job can only accept one applicant and a job applicant can be appointed for only one job.
Applicant 0 is interested in job 4, ...
Q: Find an assignment of jobs to applicants in such that the maximum number of applicants find the job they are interested in.

7. Hamilton Path



In an undirected graph, (i,j) means that node i and node j are connected with an undirected edge.
The nodes are numbered from 0 to 4, and the edges are: (4,2) (0,4) (4,3) (0,1) (0,2) (4,1) (2,3)
Q: Is there a path in this graph that visits every node exactly once? If yes, give the path. Note that in a path, adjacent nodes must be connected with edges.

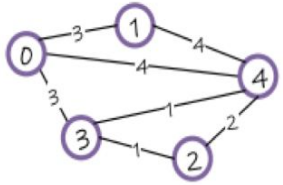
8. GNN



In an undirected graph, the nodes are numbered from 0 to 4, and every node has an embedding. (i,j) means that node i and node j are connected with an undirected edge.
Embeddings: node 0: [1,1], ⋯
The edges are: (0,1) ...
In a simple graph convolution layer, each node's embedding is updated by the sum of its neighbors' embeddings.
Q: What's the embedding of each node after one layer of simple graph convolution layer?

Method

Standard Prompting



<in-context exemplar>
In an undirected graph, the nodes are numbered from 0 to 4, and the edges are:
an edge between node 0 and node 4 with weight 4,
an edge between node 0 and node 3 with weight 3,
an edge between node 0 and node 1 with weight 3,
...
Q: Give the shortest path from node 0 to node 2.

Build-a-Graph Prompting

<in-context exemplar>
In an undirected graph, the nodes are numbered from 0 to 4, and the edges are:
an edge between node 0 and node 4 with weight 4, ⋯

Let's construct a graph with the nodes and edges first.

Q: Give the shortest path from node 0 to node 2.
A: All the paths from node 0 to node 2 are:
0,3,2 with a total weight of 3 + 1 = 4,
0,1,4,2 with a total weight of 3 + 4 + 2 = 9,
0,4,3,2 with a total weight of 4 + 1 + 1 = 6.
The weight of path 0,3,2 is the smallest, so the shortest path from node 0 to node 2 is 0,3,2 with a total weight of 4.

Algorithmic Prompting

We can use a Depth-First Search (DFS) algorithm to find the shortest path between two given nodes in an undirected graph.
The basic idea is to start at one of the nodes and use DFS to explore all of its adjacent nodes. At each node, you can keep track of the distance it takes to reach that node from the starting node.
Once you have explored all the adjacent nodes, you can backtrack and pick the node which has the shortest distance to reach the destination node.
<in-context exemplar>
In an undirected graph, the nodes are numbered from 0 to 4, and the edges are:
an edge between node 0 and node 4 with weight 4, ...
Q: Give the shortest path from node 0 to node 2.
A: All the paths from node 0 to node 2 are:
0,3,2 with a total weight of 3 + 1 = 4,
0,1,4,2 with a total weight of 3 + 4 + 2 = 9,
0,4,3,2 with a total weight of 4 + 1 + 1 = 6.
The weight of path 0,3,2 is the smallest, so the shortest path from node 0 to node 2 is 0,3,2 with a total weight of 4.

Results

Dataset	ZERO-SHOT	FEW-SHOT	CoT	0-CoT	CoT+SC	Avg.
GENERAL	74.67	83.33	85.33	66.00	82.67	78.40
CHAIN	51.67 (-23.00)	45.00 (-35.33)	40.83 (-44.50)	92.50 (+26.50)	44.17 (-38.50)	54.83 (-23.57)
CLIQUE	60.83 (-13.84)	73.33 (-10.00)	85.00 (-0.33)	52.50 (-13.50)	83.33 (+0.66)	71.00 (-7.40)

Method	Cycle				Shortest Path					Hamilton Path		
	Easy	Medium	Hard	Avg.	Easy	Hard	Easy (PC)	Hard (PC)	Avg.	Easy	Hard	Avg.
CoT	84.67	63.33	53.25	66.75	63.89	29.50	76.84	35.79	51.51	**40.00**	**8.00**	**24.00**
CoT+BAG	**86.00**	69.33	62.00	**72.44**	**67.78**	**33.50**	**79.20**	**42.56**	**55.76**	38.67	6.00	22.34
CoT+ALGORITHM	77.33	**74.00**	**64.00**	71.78	63.89	28.00	76.06	38.70	51.66	36.67	7.50	22.09

Take away message:

- ColPali **simplifies and enhances document retrieval** by leveraging Vision-Language Models (VLMs) t, eliminating the need for text extraction.

- ColPali also shows a **way to MLLM application into different specific domains**. This could go beyond retrieval to **knowledge extraction**, **Forensic/Financial/Cultural industry document analysis**, and **AI-powered research assistants**.

Gonçalo Marcelino

Understanding Barriers to Network Exploration with Visualization: A Report from the Trenches

Mashael AlKadi, Vanessa Serrano, James Scott-Brown, Catherine Plaisant,
Jean-Daniel Fekete, Uta Hinrichs and Benjamin Bach

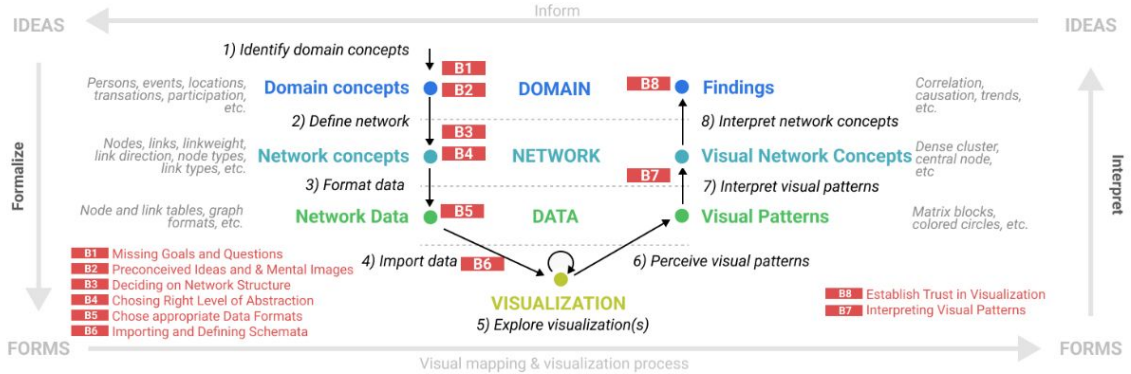
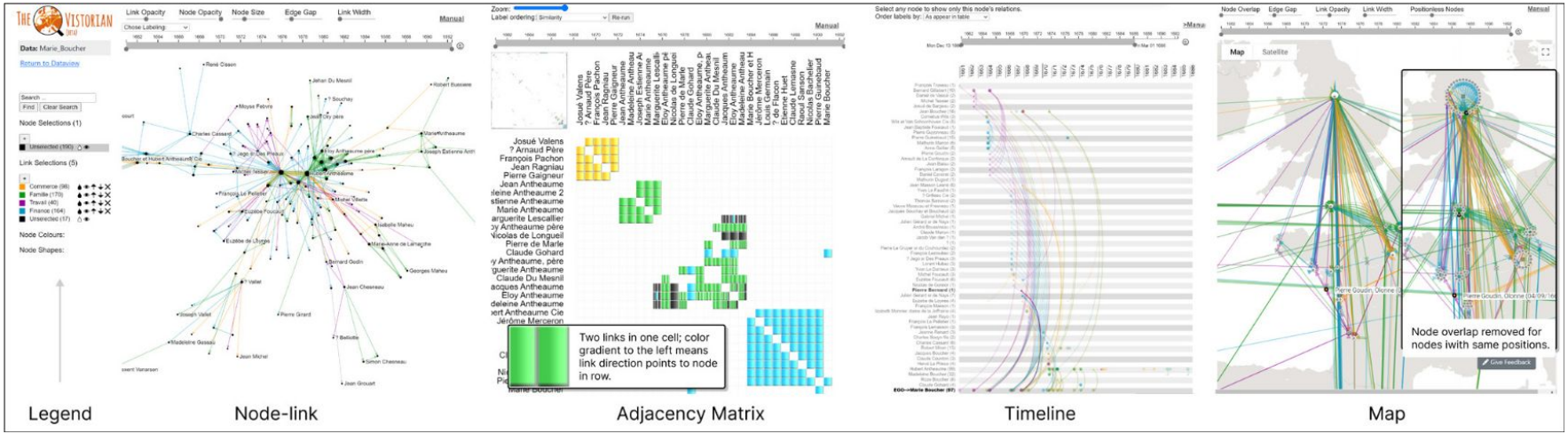


Fig. 1: Barriers (red) that may be encountered during visual network exploration process, while translating domain concepts (ideas) into network structures and visualizations (forms) and back into findings.

Abstract—This article reports on an in-depth study that investigates barriers to network exploration with visualizations. Network visualization tools are becoming increasingly popular, but little is known about how analysts plan and engage in the visual exploration

Network visualization tools are becoming increasingly popular. How do users engage in the visual exploration of network data, which exploration strategies they employ, and how they prepare their data, define questions, and decide on visual mappings?

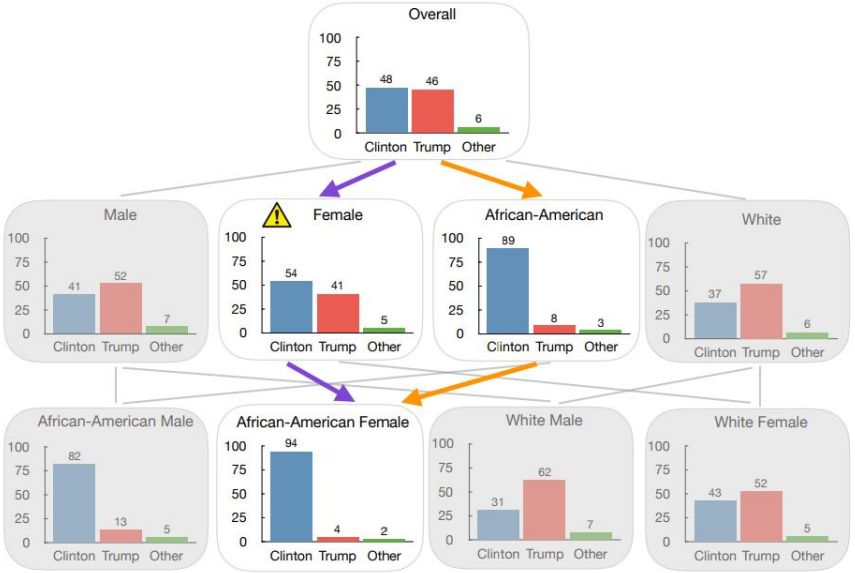


Study 1 - Researchers tracked users of the Vistorian logging 534 sessions to understand how it is being used.

Study 2 - Researchers collected qualitative during during a 6-week network exploration course by monitoring 36 participants. 50% without experience in network visualization.

- Missing Goals & Questions
  - ✗ Choosing schemas and visualizations is difficult without specific goals and can lead to drill down fallacy.
  - ✓ Sketching, and examples improved results.

- Preconceived Ideas & Mental Images / Deciding on a Network Structure
  - ✗ Wrong preconceived about what a graph visualization is (e.g. exclusively a social network graph)
  - ✓ Guided process to construct schemas and fast sketching.



Lee, Doris Jung-Lin, et al. "Avoiding drill-down fallacies with vispilot: Assisted exploration of data subsets." *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 2019.

- Choosing The Right Level of Abstraction
  - ✗ Choosing the wrong level of abstraction can lead to cluttered graphs.
  - ✓ Transformation and aggregation strategies.
- Interpreting Visual Patterns in Visualization
  - ✗ Understanding patterns in data is complex specially when interaction can lead to changes of visual patterns on-the-fly.
  - ✓ Support multiple coordinated views, support for examples.
- Establish Trust in a Network Visualization
  - ✗ Unfamiliar visualization (e.g. adjacency matrix) and misunderstanding provenance.
  - ✓ Showing examples of use in credible sources (e.g.) journalism, explaining algorithms, explaining anti-patterns.

Fatemeh Gholamzadeh

G1: Teaching LLMs to Reason on Graphs with Reinforcement Learning

Xiaojun Guo[*] Peking University
Ang Li[*] Peking University
Yifei Wang[*] MIT
Stefanie Jegelka TUM[†] and MIT[‡]
Yisen Wang[§] Peking University

Abstract

Although Large Language Models (LLMs) have demonstrated remarkable progress, their proficiency in graph-related tasks remains notably limited, hindering the development of truly general-purpose models. Previous attempts, including pre-training graph foundation models or employing supervised fine-tuning, often face challenges such as the scarcity of large-scale, universally represented graph data. We introduce G1, a simple yet effective approach demonstrating that Reinforcement Learning (RL) on synthetic graph-theoretic tasks can significantly scale LLMs' graph reasoning abilities. To enable RL training, we curate Erdős, the largest graph reasoning dataset to date comprising 50 diverse graph-theoretic tasks of varying difficulty levels, 100k training data and 5k test data, all drived from real-world graphs. With RL on Erdős, G1 obtains substantial improvements in graph reasoning, where our finetuned 3B model even outperforms Qwen2.5-72B-Instruct (24x size). RL-trained models also show strong zero-shot generalization to unseen tasks, domains, and graph encoding schemes, including other graph-theoretic benchmarks as well as real-world node classification and link prediction tasks, without compromising general reasoning abilities. Our findings offer an efficient, scalable path for building strong graph reasoners by finetuning LLMs with RL on graph-theoretic tasks, which combines the strengths of pretrained LLM capabilities with abundant, automatically generated synthetic data, suggesting that LLMs possess graph understanding abilities that RL can elicit successfully. Our implementation is open-sourced at https://github.com/PKU-ML/G1, with models and datasets hosted on Hugging Face collections PKU-ML/G1 for broader accessibility.

Motivation

LLMs excel in text reasoning, but **struggle with graph reasoning**

Gap:

- **Limited data for graph reasoning,**
- **graph-structured problems** (like connectivity, shortest paths, cycles, centrality, NP-hard problems) are not well represented in internet text.

Problem Statement

- Goal: teach LLMs to **solve graph-theoretic tasks**

Problem Statement

- Goal: teach LLMs to **solve graph-theoretic tasks**
- Key challenges:
  - Complex & diverse graph structures
  - Encoding graphs into text for LLMs
  - Zero-shot generalization to **new tasks and domains**

G1 Approach

Core idea: use *Reinforcement Learning* (RL) on synthetic graph tasks

LLMs trained on Internet-scale data already have some graph reasoning ability, and we can bring it out using their own trial and error without relying on human data.

RL reward: correctness of answers → no manual labels needed

Training pipeline:

1. Generate graph tasks (synthetic)
2. Fine-tune base LLM with RL
3. Evaluate on diverse benchmarks

The Erdős Dataset

- New benchmark for graph reasoning
- ~50 **graph-theoretic tasks**
- 100K training, 5K test examples
- Derived from **real-world graphs**

Difficulty	Tasks	Ratio	Base Model Acc	G1 Acc
Easy	Node Number, Dominating Set, Common Neighbor, Edge Number, Neighbor, BFS, Has Cycle, DFS, Minimum Spanning Tree, Edge Existence, Is Regular, Degree, Is Tournament, Density	29.16%	57.16%	**95.07%**
Medium	Adamic Adar Index, Clustering Coefficient, Connected Component Number, Bipartite Maximum Matching, Local Connectivity, Jaccard Coefficient, Min Edge Covering, Is Eularian, Degree Centrality, Is Bipartite, Resource Allocation Index	22.91%	42.55%	**88.91%**
Hard	Max Weight Matching, Closeness Centrality, Traveling Salesman Problem, Strongly Connected Number, Shortest Path, Center, Diameter, Barycenter, Radius, Topological Sort, Periphery, Betweenness Centrality, Triangles, Average Neighbor Degree, Harmonic Centrality, Bridges	33.33%	18.87%	**50.44%**
Challenging	Isomophic Mapping, Global Efficiency, Maximal Independent Set, Maximum Flow, Wiener Index, Hamiltonian Path, Min Vertex Cover	14.58%	3.29%	**23.57%**

Difficulty	Tasks	Ratio	Base Model Acc	G1 Acc
Easy	Node Number, Dominating Set, Common Neighbor, Edge Number, Neighbor, BFS, Has Cycle, DFS, Minimum Spanning Tree, Edge Existence, Is Regular, Degree, Is Tournament, Density	29.16%	57.16%	**95.07%**
Medium	Adamic Adar Index, Clustering Coefficient, Connected Component Number, Bipartite Maximum Matching, Local Connectivity, Jaccard Coefficient, Min Edge Covering, Is Eularian, Degree Centrality, Is Bipartite, Resource Allocation Index	22.91%	42.55%	**88.91%**
Hard	Max Weight Matching, Closeness Centrality, Traveling Salesman Problem, Strongly Connected Number, Shortest Path, Center, Diameter, Barycenter, Radius, Topological Sort, Periphery, Betweenness Centrality, Triangles, Average Neighbor Degree, Harmonic Centrality, Bridges	33.33%	18.87%	**50.44%**
Challenging	Isomophic Mapping, Global Efficiency, Maximal Independent Set, Maximum Flow, Wiener Index, Hamiltonian Path, Min Vertex Cover	14.58%	3.29%	**23.57%**

Rule-based Rewards on Graphs.

- Strict value matching.For tasks that have a unique ground truth value, e.g., node counting. a reward of +1, otherwise 0
- Jaccard Index for set matching. For problems whose answer is not a single value sˆ but an unordered e.g., common neighbors of two nodes
- Algorithmic verification. Lastly, for problems that have multiple correct solutions (e.g., shortest paths). algorithmic verifiers to check correctness of the proposed solutions.

Test accuracy (%) comparison of different LLMs of varying sizes on our Erdős benchmark tasks. In all experiments Qwen2.5-Instruct is used as as base model.

Model	Easy	Medium	Hard	Challenging	Average
Proprietary (Unknown Parameters)					
GPT-4o-mini	76.20	72.07	28.81	3.34	47.60
OpenAI o3-mini (w/ tool use)	74.83	83.49	59.28	43.22	64.90
3B Parameters					
Llama-3.2-3B-Instruct	36.50	21.45	6.81	1.14	17.32
Qwen2.5-3B-Instruct (base model)	45.71	30.18	9.44	1.29	22.72
Direct-SFT-3B (Ours)	74.43	75.27	**43.69**	**14.43**	53.78
CoT-SFT-3B (Ours)	65.57	67.64	29.44	4.57	43.56
G1-3B (Ours)	**94.86**	**84.64**	41.25	7.57	**59.76** (+37.04)
7B Parameters					
Llama-3.1-8B-Instruct	49.21	30.45	13.69	1.43	25.10
Qwen2.5-7B-Instruct (base model)	57.36	42.55	18.87	3.29	32.06
Qwen2.5-Math-7B-Instruct	52.79	39.64	14.82	2.46	28.94
DeepSeek-R1-Distill-Qwen-7B	71.79	73.73	39.12	16.57	51.64
GraphWiz-7B-RFT	14.57	13.73	1.38	0.47	7.70
GraphWiz-7B-DPO	20.36	19.09	1.44	0.78	10.59
Direct-SFT-7B (Ours)	73.57	75.91	39.12	10.71	51.76
CoT-SFT-7B (Ours)	72.57	75.73	38.50	11.00	51.34
G1-7B (Ours)	**95.07**	**88.91**	**50.44**	**23.57**	**66.16** (+34.10)
70B Parameters					
Llama-3.1-70B-Instruct	68.07	55.45	31.87	4.44	42.28
Qwen2.5-72B-Instruct	71.71	67.81	33.37	8.22	47.16

Transferability to Other Graph Reasoning Benchmarks

Table 3: Test accuracy (%) by computational complexity on the GraphWiz benchmark.

Model	Linear	Poly	NP-Complete	Avg.
Llama-3.2-3B-Instruct	29.80	3.00	2.50	19.80
Qwen2.5-3B-Instruct (base)	40.25	9.58	**69.12**	36.44
G1-3B	**58.06**	**26.75**	**69.12**	**50.08**
Llama-3.1-8B-Instruct	54.00	5.67	32.12	33.03
DeepSeek-R1-Distill-Qwen-7B	57.69	31.42	70.88	51.86
GraphWiz-7B-RFT	67.56	29.83	43.38	49.61
GraphWiz-7B-DPO	63.88	**36.25**	39.50	49.25
Qwen2.5-7B-Instruct (base)	49.06	17.92	**76.12**	44.69
G1-7B	**68.00**	32.25	72.62	**57.11**

Table 4: Test accuracy (%) by computational complexity on the GraphArena benchmark.

Model	Poly-Time		NP-Complete		Avg.
	Easy	Hard	Easy	Hard	
Llama-3.2-3B-Instruct	22.25	6.75	8.00	0.66	8.40
Qwen2.5-3B-Instruct (base)	31.50	14.50	17.33	1.50	14.85
G1-3B	**57.50**	**26.75**	**24.66**	**1.83**	**24.80**
Llama-3.1-8B-Instruct	47.00	21.25	22.00	2.16	20.90
DeepSeek-R1-Distill-Qwen-7B	66.0	22.75	34.83	1.50	28.65
GraphWiz-7B-RFT	2.25	0.75	0.83	0.00	0.85
GraphWiz-7B-DPO	0.25	1.00	0.66	0.16	0.49
Qwen2.5-7B-Instruct (base)	62.00	35.75	28.83	2.16	28.84
G1-7B	**77.50**	**44.25**	**47.33**	**8.50**	**41.10**

These results demonstrate that G1 has strong zero-shot generalization ability to unseen graph encoding methods, graph distributions, and graph tasks.

G1 on Real-world, Non-graph-theoretic Graph-reasoning Tasks

node classification and link prediction

Cora and PubMed citation graphs

Each instance includes a description of the target node (or node pair) containing the paper ID and title, along with the textual and structural information of neighboring nodes.

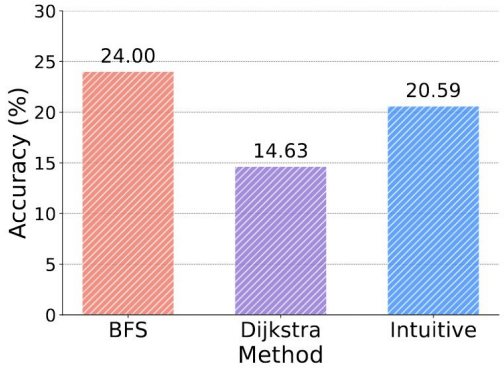
Table 5: Test accuracy (%) on Node Classification and Link Prediction benchmarks.

Model	Node		Link		Avg.
	Cora	PubMed	Cora	PubMed	
Llama-3.2-3B-Instruct	68.77	75.20	60.40	57.60	64.79
Qwen2.5-3B-Instruct (base)	70.83	75.08	62.15	58.38	65.66
CoT-SFT-3B	75.97	81.47	75.70	**71.52**	75.12
G1-3B	**77.25**	**83.88**	**78.97**	69.75	**75.16**
Llama-3.1-8B-Instruct	70.90	75.00	50.60	46.10	59.53
DeepSeek-R1-Distill-Qwen-7B	76.50	81.25	68.03	78.72	78.80
Qwen2.5-7B-Instruct (base)	**79.30**	85.35	**88.22**	88.67	85.50
CoT-SFT-7B	73.20	83.25	64.70	68.12	73.17
G1-7B	79.20	**86.20**	87.98	**91.88**	**87.29**

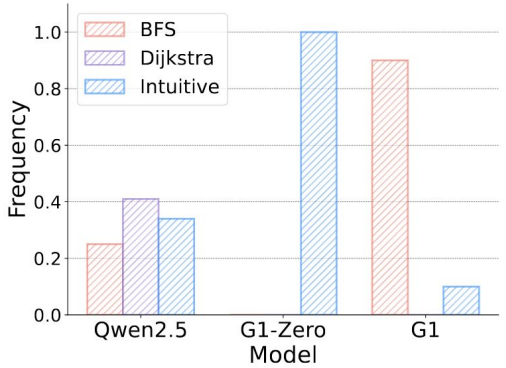
Understanding the Benefits of RL Training for Graph Reasoning

Qwen2.5-3B-Instruct (base),
G1-Zero-3B (RL only),
and G1-3B (SFT & RL).

1) Breadth-First Search (BFS),
2) Dijkstra's algorithm,
3) Intuitive deductions



(a) The accuracy of different graph reasoning patterns for shortest path on Qwen2.5-3B-Instruct.



(b) Frequency of different graph reasoning patterns for Qwen2.5-3B-Instruct, G1-Zero-3B and G1-3B.

Yahia Dalbah

Fugatto 1

Foundational Generative Audio Transformer Opus 1

NVIDIA
Rafael Valle, Rohan Badlani, Zhifeng Kong, Sang-gil Lee, Arushi Goel, Sungwon Kim, João Felipe Santos, Shuqi Dai, Siddharth Gururani, Aya AlJa'fari, Alexander H. Liu, Kevin Shih, Ryan Prenger, Wei Ping, Chao-Han Huck Yang, Bryan Catanzaro
rafaelvalle@nvidia.com

ABSTRACT

Fugatto is a versatile audio synthesis and transformation model capable of following free-form text instructions with optional audio inputs. While large language models (LLMs) trained with text on a simple next-token prediction objective can learn to infer instructions directly from the data, models trained solely on audio data lack this capacity. This is because audio data does not inherently contain the instructions that were used to generate it. To overcome this challenge, we introduce a specialized dataset generation approach optimized for producing a wide range of audio generation and transformation tasks, ensuring the data reveals meaningful relationships between audio and language. Another challenge lies in achieving compositional abilities – such as combining, interpolating between, or negating instructions – using data alone. To address it, we propose *ComposableART*, an inference-time technique that extends classifier-free guidance to compositional guidance. It enables the seamless and flexible composition of instructions, leading to highly customizable audio outputs outside the training distribution. Our evaluations across a diverse set of tasks demonstrate that *Fugatto* performs competitively with specialized models, while *ComposableART* enhances its sonic palette and control over synthesis. Most notably, we highlight emergent tasks and properties that surface in our framework's – sonic phenomena that transcend conventional audio generation – unlocking new creative possibilities. Demo Website.

Dataset generation

1- Free-Form Instruction Synthesis via pre-defined python generators

2- relative instruction generation (happy voice => happier voice)

3- use classifiers & LLM to generate descriptions

4- datasets that have explicit isolated factors

5- use Praat and Spotify’s Pedalboard to edit speech and music

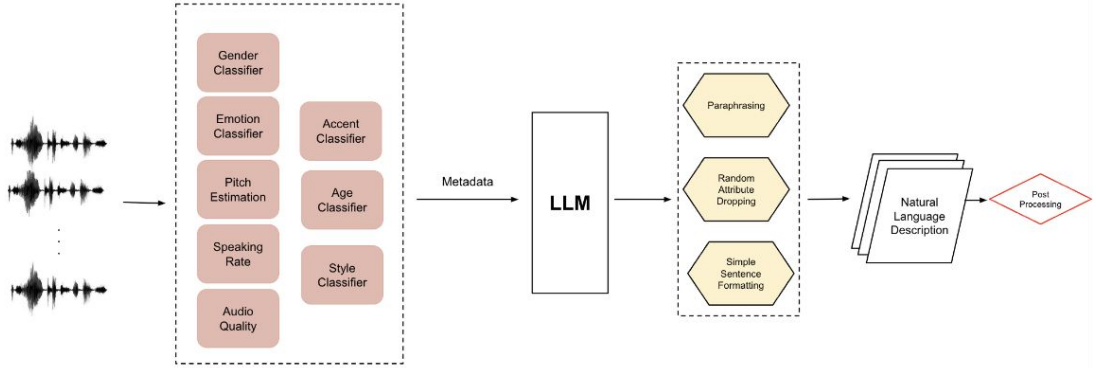


Figure 4: Synthetic caption generation pipeline for Prompt-to-Voice (P2V).

Model & Operation

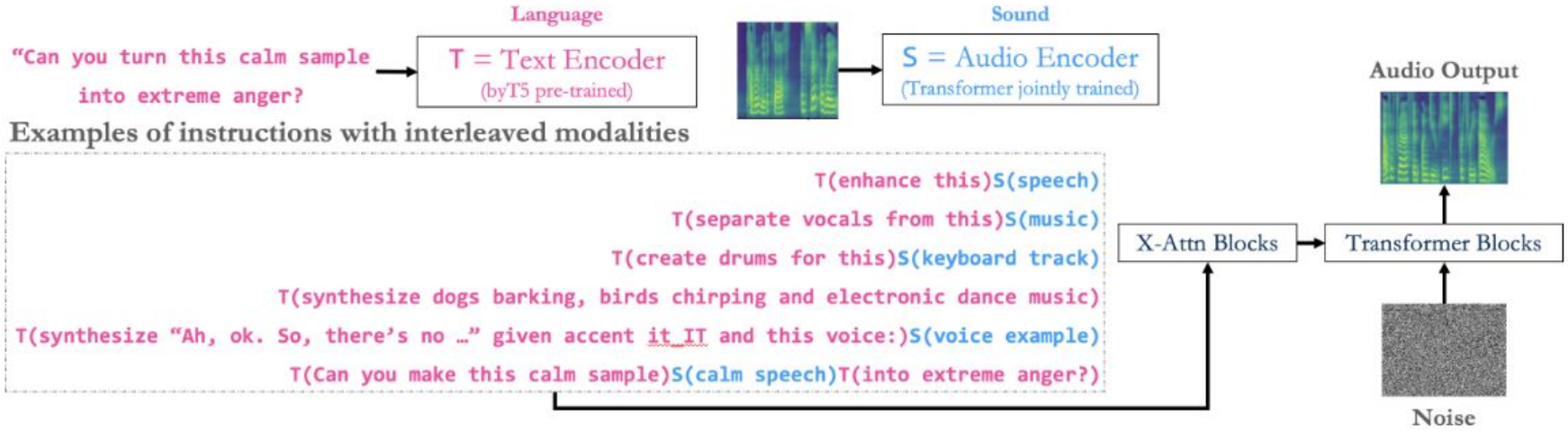


Figure 5: A description of *Fugatto*'s architecture and input handling.

Emergent sounds & tasks

The model ‘can’ generate sounds that were not present in the dataset, and do tasks that it was not explicitly trained on doing.

https://fugatto.github.io/