

# Retrieving and Generating data using LLMs in Python

**Prashant Garg**

(Imperial College London)

# Prashant Garg



Hi! I'm Prashant, a PhD student at Economics and Public Policy Department of Imperial College Business School. I am also currently a visiting researcher at International Finance Corporation (IFC) and University of Cambridge.

Research Areas:

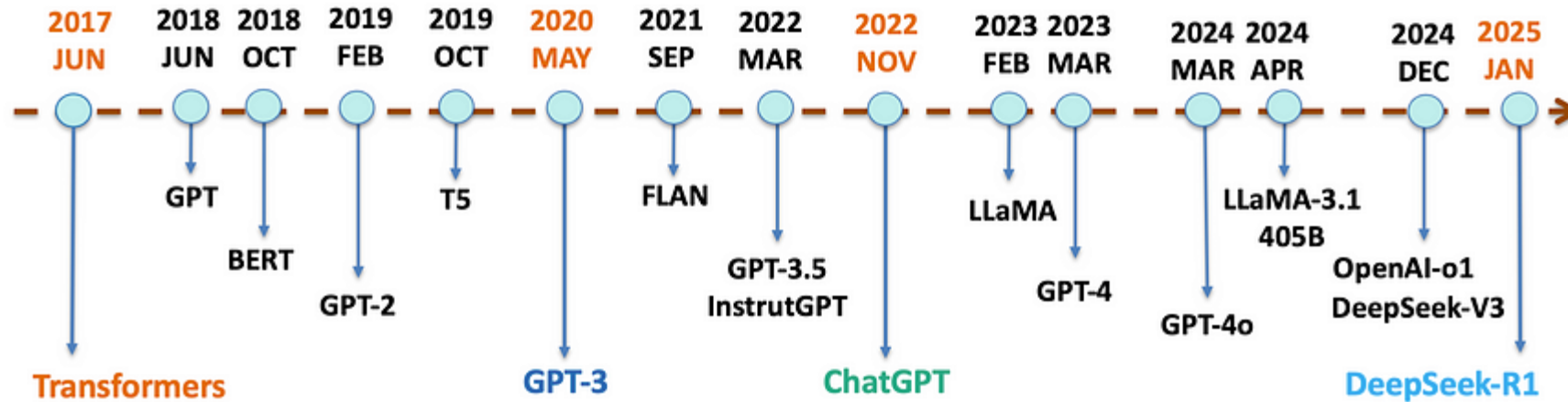
- **AI and Big Data**
- **Economics of Networks**
- **Science of Science**
- **Media and Political Economy**

Find my research papers [here](#)

Please feel free to contact me at [prashant.garg@imperial.ac.uk](mailto:prashant.garg@imperial.ac.uk)

Find my research at [www.prashantgarg.org](http://www.prashantgarg.org)

# A Brief History of LLMs



[source](#)

## Why-now?

- Cost per 1K tokens ↓ 97x (2020 → 2025).
- Context window ↑ from 2K → 128K tokens (64x).
- Open-weight models (Mistral, Llama 3) close the gap.

## Practical uses

- Retrieval & extraction
- Structured generation
- Code assist
- Semantic search
- Data cleaning
- Synthetic data / anonymisation

# Model providers at a glance

API / Platform	Example models	Max context	Price /1K in-tokens	Licence
<b>OpenAI</b>	GPT-3.5-Turbo, GPT-4o-mini	128 K	\$ 0.50	Closed
<b>Anthropic</b>	Claude 3 Sonnet	200 K	\$ 0.25	Closed
<b>Google</b>	Gemini 1.5 Flash	128 K	\$ 0.35	Closed
<b>Open-weights</b>	Llama 3 70B, Mistral 8×22B	8–32 K	\$ 0 (run yourself)	Apache-2 / CC

## Which one to pick?

- “Closed” vs “Open”
- Features / parameters
- Costing
- Leaderboard stability
- Off the shelf / fine-tunable
- Use case

✓ Same workflow works across providers

# Five Workflow Axes

- **Retrieval** (pull facts from documents)
- **Generation** (create new structured objects)
- **Classification** (label things, e.g., stance, gender, race)
- **Pruning** (filter bad or vague items)
- **Structured Output** (enforce JSON / schema)

# API

```
from openai import OpenAI
client = OpenAI()

resp = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user",
                "content": "One-sentence why data cleaning matters"}],
    temperature=0
)
print(resp.choices[0].message.content)
```

# API

```
from openai import OpenAI
client = OpenAI()

resp = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user",
                "content": "One-sentence why data cleaning matters"}],
    temperature=0
)
print(resp.choices[0].message.content)
```

Useful features:	What it does & typical use
<b>temperature 0 → 2</b> default = 0.7	Controls randomness. <ul style="list-style-type: none"><li>• 0 = deterministic (exact repeat)</li><li>• 0.3 – 0.7 = factual extraction</li><li>• 0.7 &gt; 1 = brain-storm, diverse, creative output</li></ul>
<b>response_format</b> { "type": "json_schema", ... }	Enforces valid JSON. Model must emit a string that <b>passes schema validation</b> or you get a 400-error. <i>Must-have for tables, edge lists, MCQs, multiple questions</i>
<b>Batch endpoint</b> client.batches.create(...)	<ul style="list-style-type: none"><li>• Upload a JSONL with <b>≤ 100 MB</b> lines.</li><li>• Up to <b>50,000</b> requests processed async (24 h window).</li><li>• Halves the cost</li></ul>

# Hello API Demo

- Open **00\_api\_smoke\_test.ipynb**
- Test key (in environment) with basic prompt
- Note output items
- If everything fine, move to any of the next notebooks.



# Notebooks overview

Pattern	Demo notebooks
Delimited context + extraction	10, 20
Schema-guided generation	30, 50, 80
Vote-based aggregation	60, 70
Embeddings for canonical mapping	40
Heuristic pruning	50

# Retrieval

When?

- You have large data
- Keep things grounded
- Quickly clean, tidy and do small tasks
- Best if there is a large corpus and the task can be done by humans

# 1. Causal Claims in Economics ([Garg & Fetzer, 2025](#))

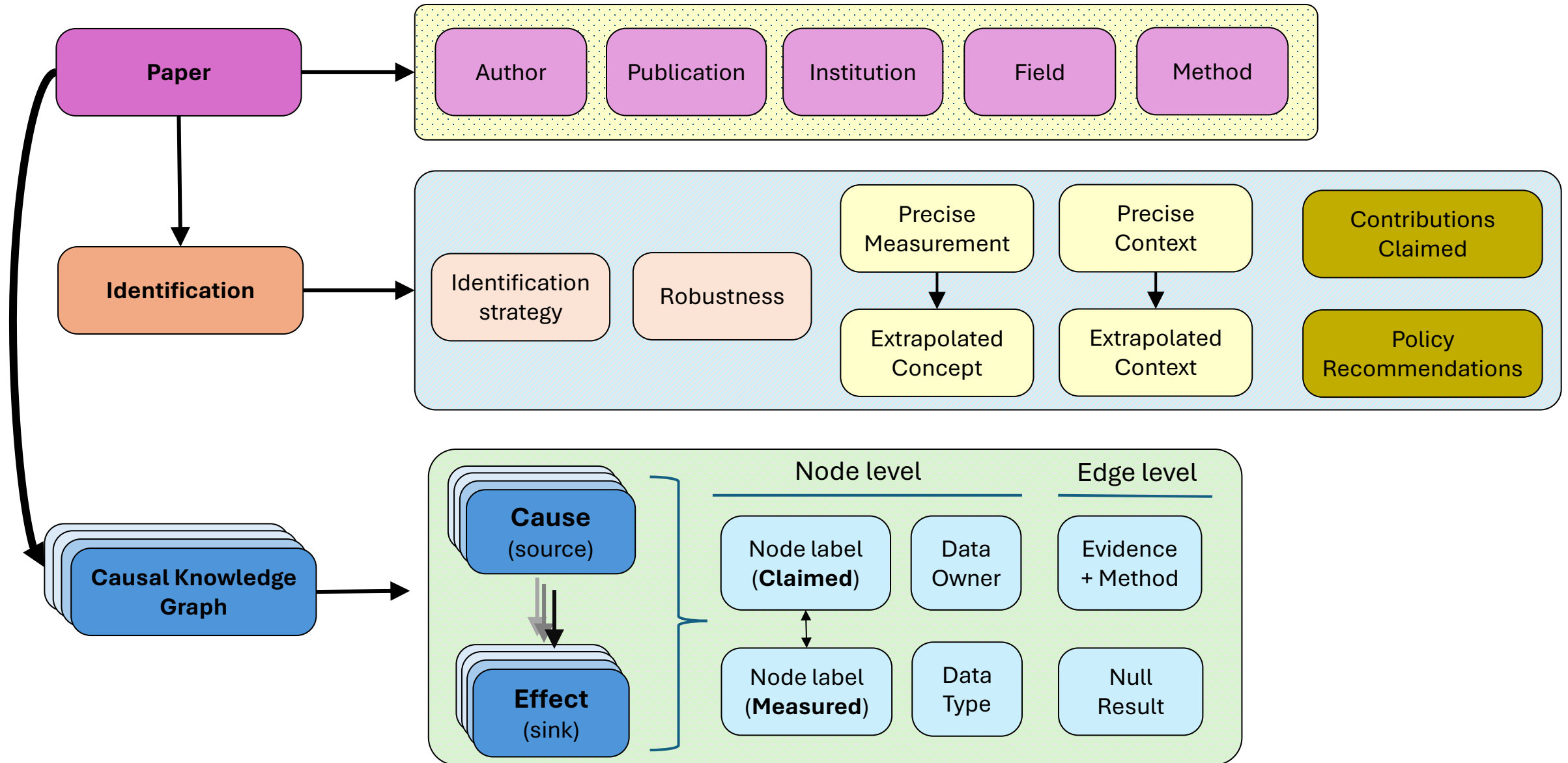
- **10\_retrieval\_edges.ipynb**
- Goal: cause-effect extraction
  - retrieving causal relationships from an unstructured document
- Input: academic text

*“Studies show lack of sleep causes decreased productivity in workers... High stress levels can lead to sleep deprivation...”*
- Prompt:

*“Extract all statements of the form X causes Y...”*
- Output format: a network, as edge-list.

```
[{"cause": "lack of sleep", "effect": "decreased productivity"}, {"cause": "high stress levels", "effect": "sleep deprivation"}]
```

# Retrieval of range of concepts using AI



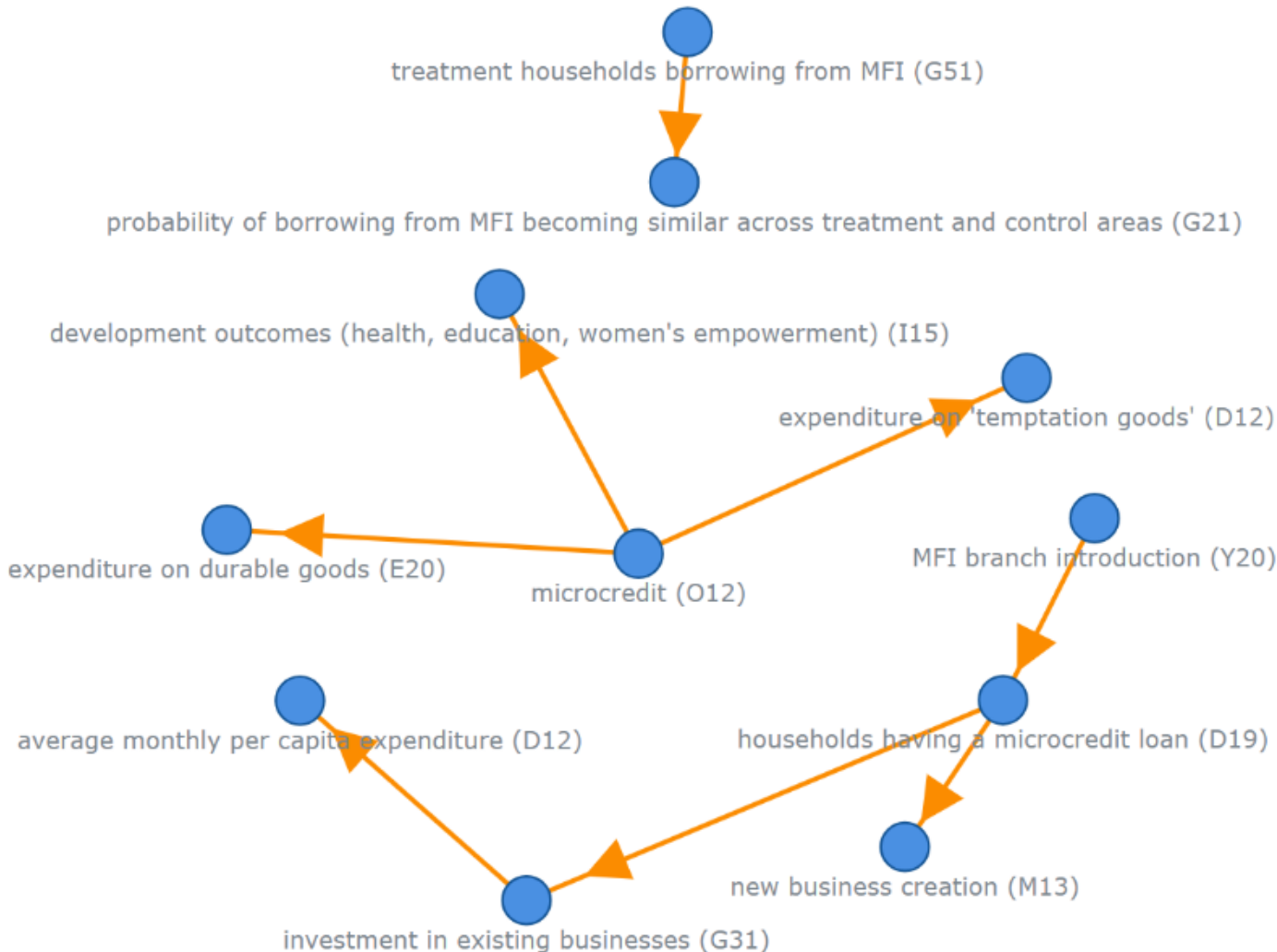
# An example

## The Miracle of Microfinance: Evidence from a Randomized Evaluation (2015, AEJ)

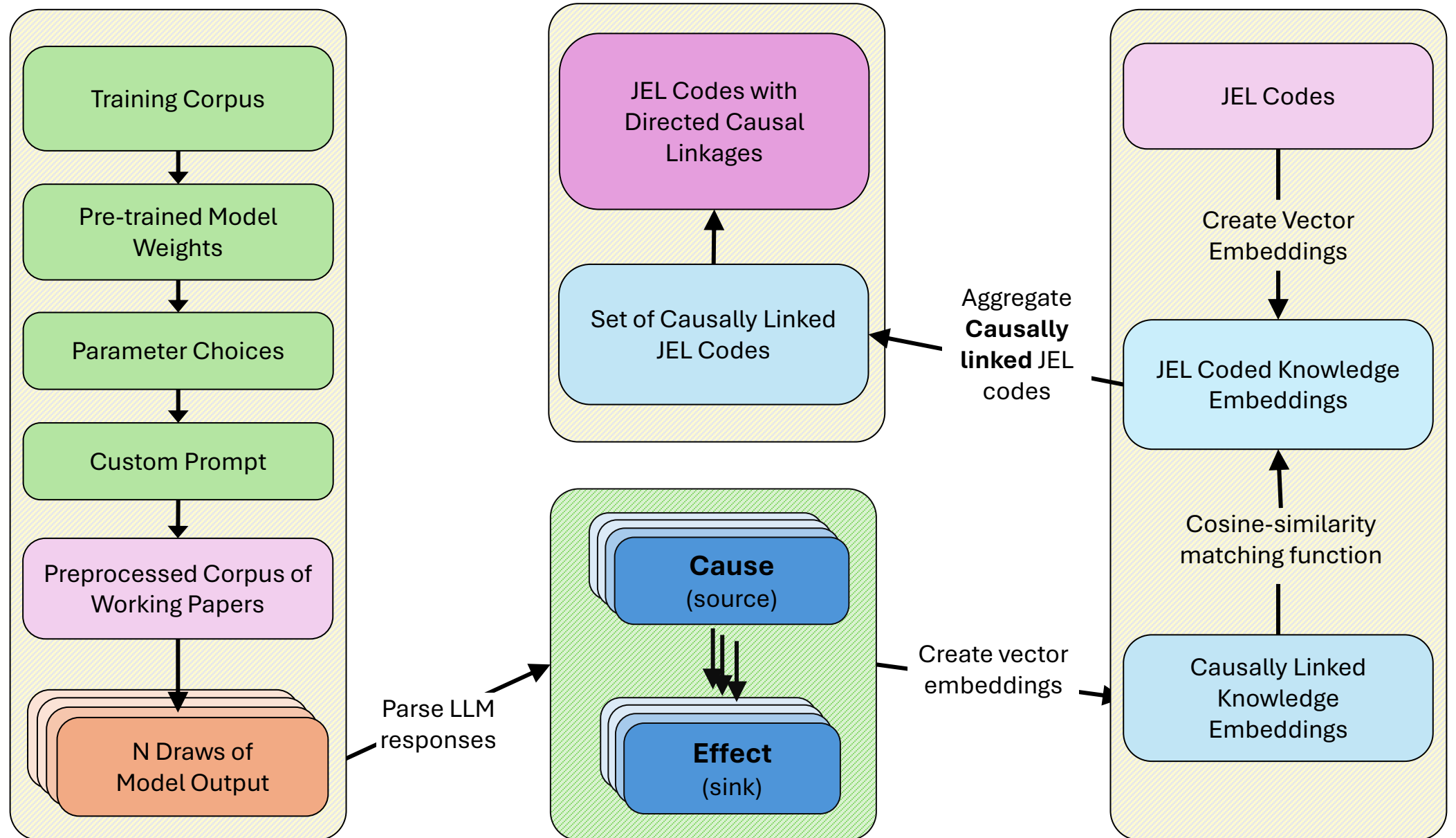
*Esther Duflo; Abhijit Banerjee; Rachel Glennerster; Cynthia G. Kinnan*

## Summary:

Evaluates the impact of introducing microfinance in India, finding increased borrowing and investment but limited effects on consumption and development outcomes.



# Mapping Causal Linkages Between JEL Codes Using AI



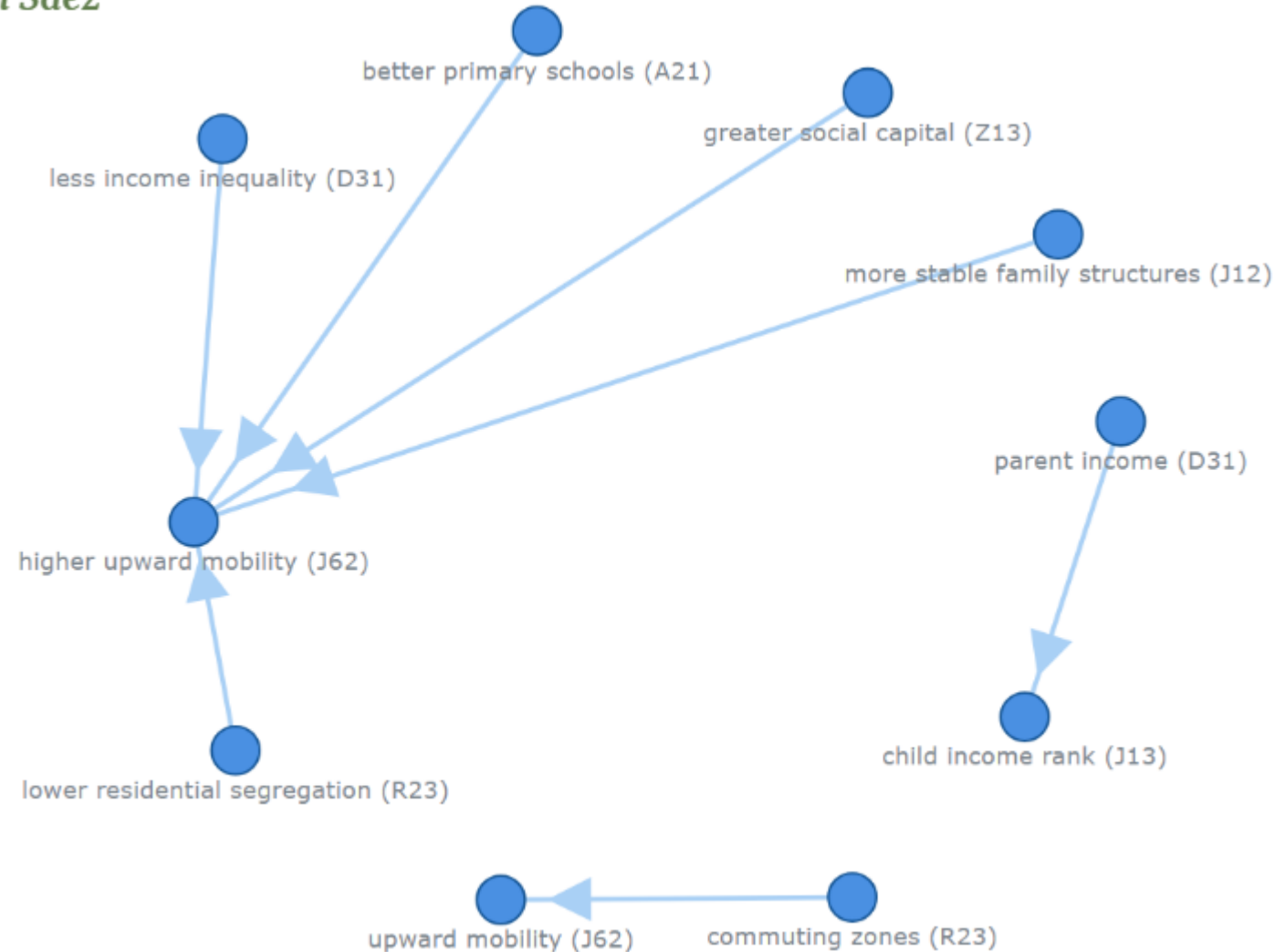
# Where is the Land of Opportunity? The Geography of Intergenerational Mobility in the United States (2014, QJE)

## Another example

*Raj Chetty, Nathaniel Hendren, Patrick Kline, Emmanuel Saez*

### Summary:

Analyzes U.S. intergenerational income mobility, identifying factors like less segregation and better schools that correlate with higher upward mobility.



# Advanced Retrieval: Two-Stage Approach

- **20\_two\_stage\_retrieval.ipynb**
- In reality, documents are large and complex.
- To ease the task for LLM, let's break task into parts.
- a.k.a. prompt-chaining or chain-of-thought – pre-cursor to “reasoning” models
- Stage 1: curate or summarise the large and complex document into a few lines or paragraphs of content for stage 2.
- Stage 2: extract structured output from this smaller, curated input.



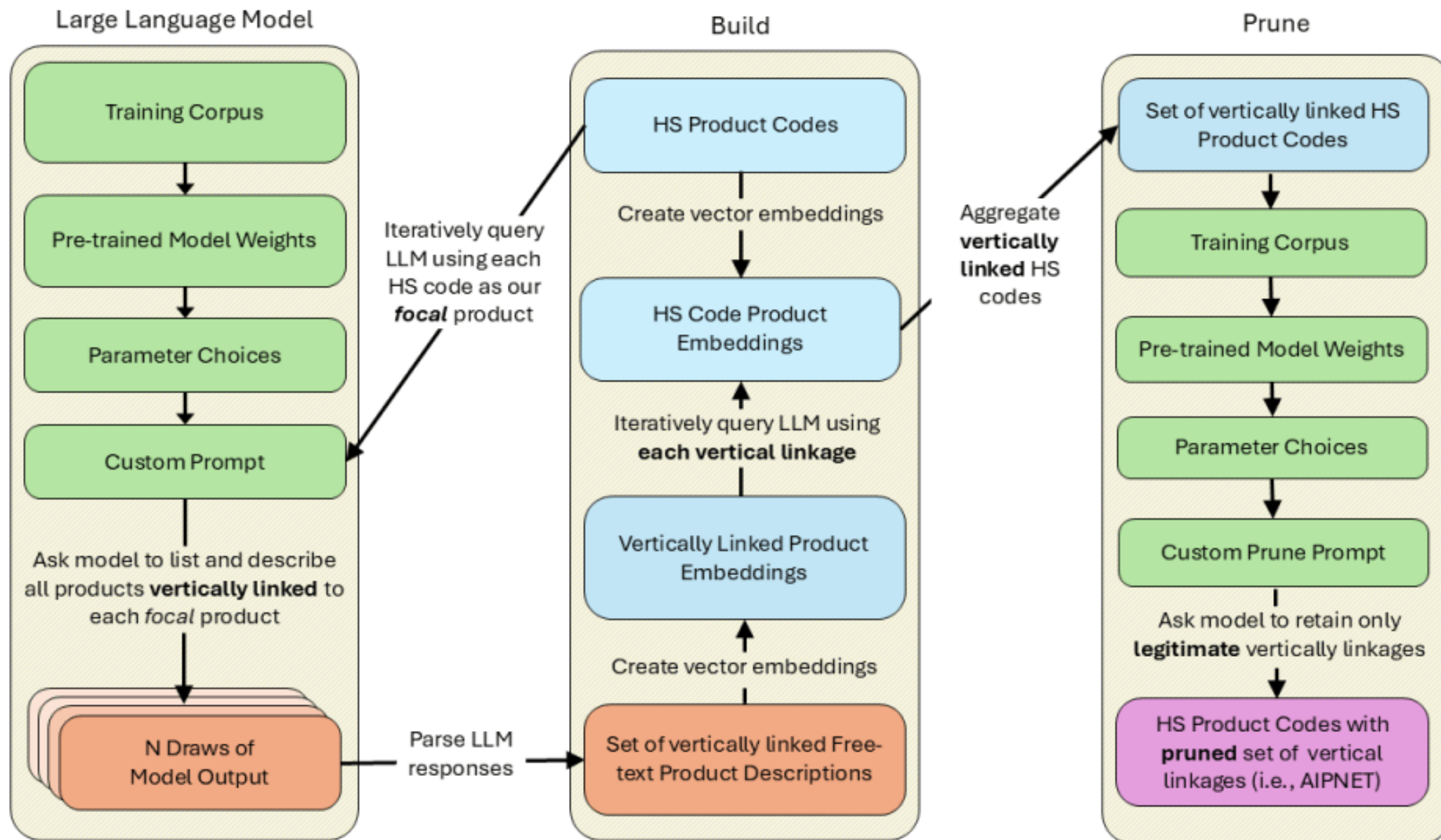
# Generation

- Take advantage of the large training corpus of LLMs
- Has access to common crawl, policy and company documents, and much more
- Your creativity is the limit
- Validation becomes paramount as there are reasons to be sceptical

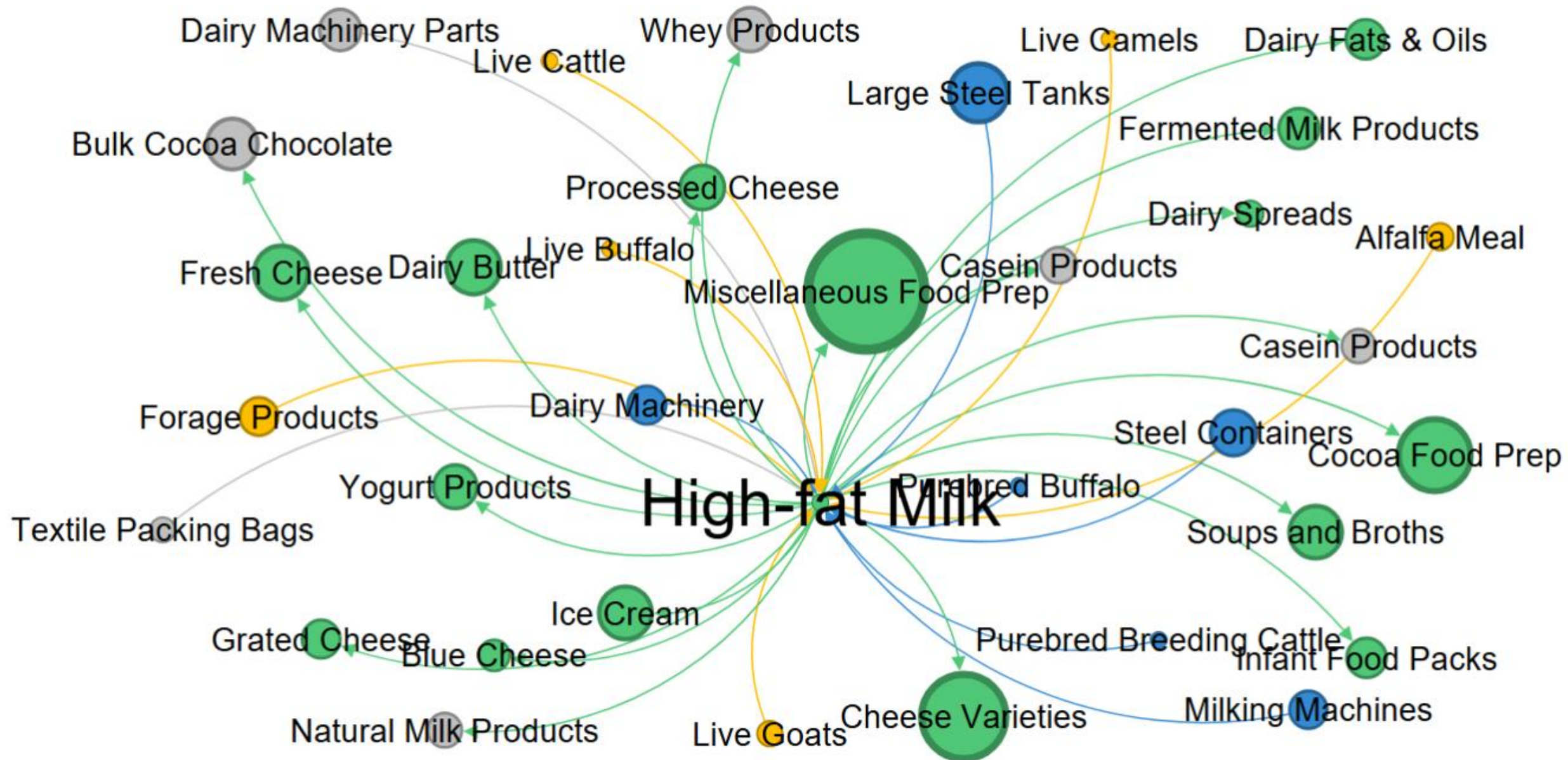
## 2. AI-Generated Production Network ([Fetzer et al., 2024](#))

- **30\_supply\_chain\_generation.ipynb**
- Task: We'll ask the model to outline a production process (a mini supply chain) for a specific product, using only the model's knowledge.
- Input: focal product, e.g. an electric car
- Prompt:
  - Generate a step-by-step production network for an electric car, from raw materials to final assembly, as a numbered list
- Output:
  - can be asked free form (for a stage 2) or directly as structured edge-list.

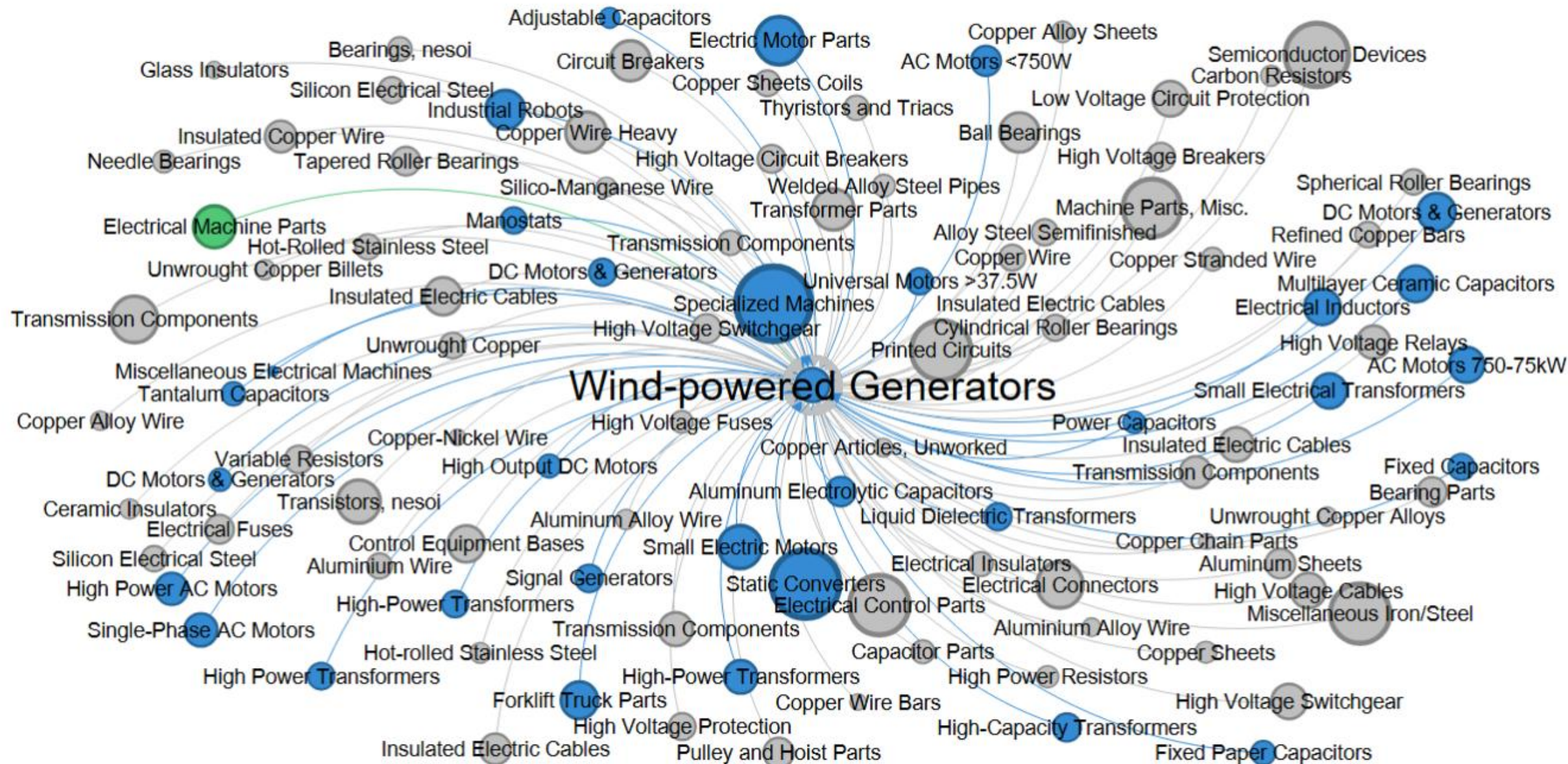
# The full pipeline



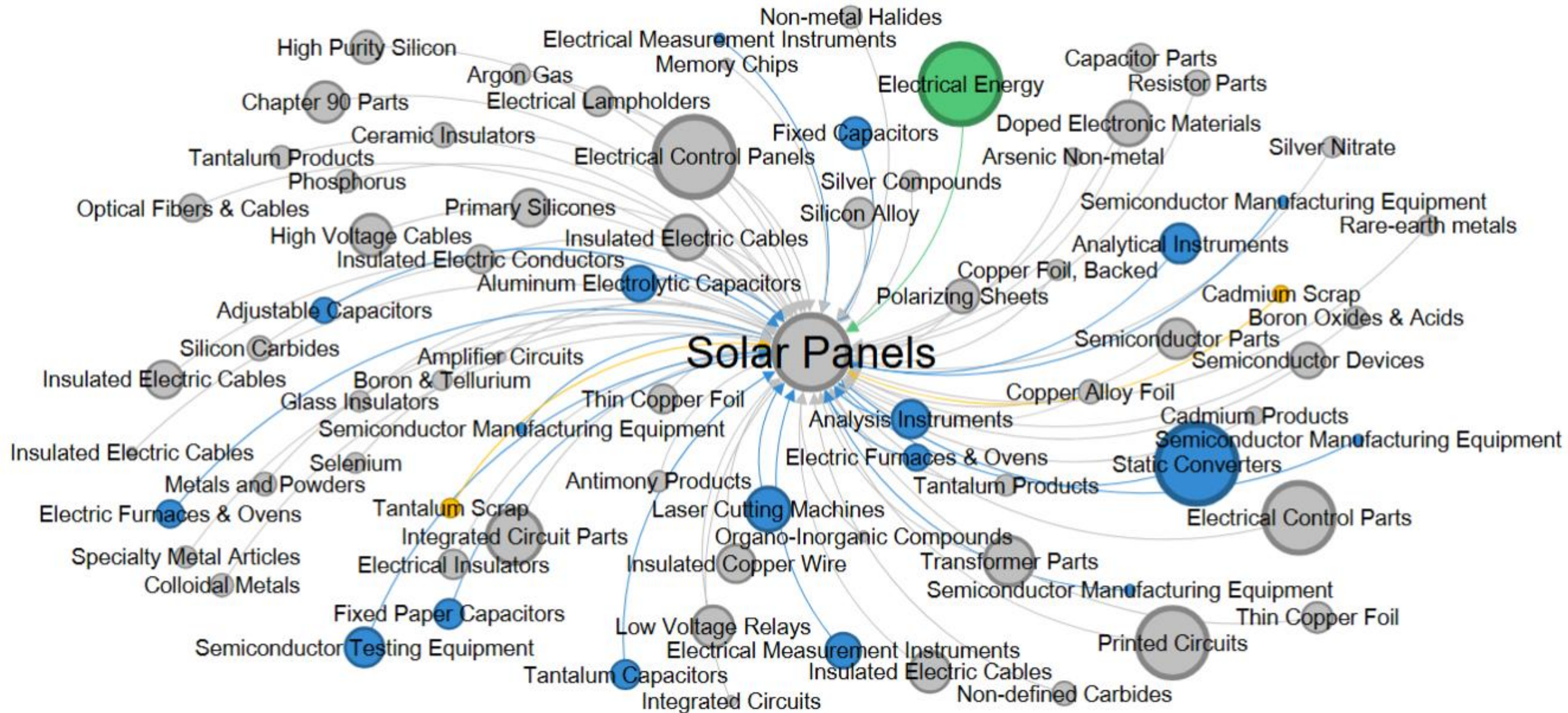
# Example: High-fat Milk











# On generation...

- Model's answer may be **plausible** (it matches common knowledge) but might not be exhaustive.
- What if we ask the same prompt  $N$  ( $=10$ ) times.
- This may theoretically increase recall, but perhaps loose precision?
- We could consider a build-prune approach (as in AIPNET)
  - Collate all  $N$  response
  - Ask LLM in second stage if a given input-output relationship is true or false. Repeat for all edges.
  - This is basically a “retrieval” step as we give our own content to LLM, but more accurately call it “pruning” or more generally, “classification”.

# 3. Embeddings & Cosine Matching

- **40\_embeddings\_mapping.ipynb**
- Why embed?
  - Map free-text nodes to controlled vocabularies (JEL, HS6).
  - Enables aggregation, visualization, and cross-document linking.
- Workflow
  - OpenAIEmbeddings → up to 3072-dimension semantic vectors
  - Apply to both “sides”, i.e., the LLM free-text and controlled vocabulary.
  - Cosine similarity matrix
  - argmax or top-k threshold
  - Manual spot-check → iterate



# Generation #2: keyword dictionary

- **50\_dictionary\_generation\_pruning.ipynb**
- Political Expression of Academics on Twitter ([Garg & Fetzer, 2025b](#))
- Now we'll have the model act as a subject matter expert that generates important keywords for specific topics
- Prompt

*Generate 5 key phrases for each of the following topics... Format as JSON with the topic as keys*
- Output

*Climate Change: Carbon Emissions, Paris 2015, COP24, Global Warming*

# Pruning once more

- Caution:
  - that while the keywords are good suggestions, they might not be exhaustive or might include common terms
- Consider adding a stage 2 “pruning” step:
  - “Does this term always refer to this topic when seen in a piece of text?”*
- Ask N times, then pick majority vote
- Post-process, e.g., substring pruning → Final dictionary

## 4. Classification (Notebook 60)

- **60\_tweet\_stance\_classification.ipynb**
- Simple and popular
- Once more: Political Expression of Academics on Twitter ([Garg & Fetzer, 2025b](#))
- Prompt:  
*Classify the following tweet as pro / anti / neutral / unrelated towards given topic.  
Topic is '{TOPIC}'. Tweet is '{TWEET}'*
- 3 independent calls → pick majority vote
- Agreement metric (1 = full, 0 = none) to flag noisy tweets
- Generation then Retrieval: You can pre-filter tweets by keyword dictionary, then classify stance.

## 4. Demographic Attribute Classification

- **70\_gender\_name\_classification.ipynb**

Task	Labels	Iterations	Agreement Heuristic
<b>Gender</b>	Male / Female / Unclear	3	modal vote
<b>Race/Ethnicity</b>	White / Non-White / Unclear	3	modal vote

- Exactly the same API pattern; swap schema & prompt
- Be aware of ethical implications and limitations. N
  - Names can be ambiguous; cultural bias; always mark ‘Unclear’ option when substantial disagreement.

## 5. More generation, sky is the limit

- **80\_company\_innovation\_generation.ipynb**
- Give a company name and geography
- Ask for large set of info, e.g., patents, green tech, digital adoption, new products, etc.
- Main use case: to get alternative data when official data is scarce or not collected due to resource constraints.
- It is work-in-progress. Requires more validation

# Best Practices on Retrieval and Generation

- Prompt clarity
- Specifying response\_format
- Controlling randomness: temperature, multiple-iterations (majority voting)
- Pipeline iteration: ask for justifications, examples, etc. in early iterations to understand what LLM understands

# Scaling & Cost Considerations

- Batch API: 50 MB / 45 k lines caps → chunking helpers in notebooks
- Minimize tokens, while keeping information same:
  - Information dense input, structured efficient output
- Pilot: start with cheapest model (e.g., gpt-4o-mini) before scaling
- Parallelism: Async calls vs. waiting loop
- Caching: store embeddings & LLM outputs; rerun only on misses

# Batching LLMs

- **90\_batching\_and\_translation.ipynb**
- Benefits: parallel, 50% cheaper, no rate-limit juggling
- Cons:
  - Slightly more complex code.
  - have to wait up to 24 hours.
    - In practice, it can be faster as it is asynchronous
    - Often runs in minutes, or few hours.
    - Faster on weekends / off-peak times.



Any questions about what we just did?

# Validation

- Validate at prompt level
- Validate at measurement level
- Eye-ball validation
- Automated and/or human validation
- Within-model stability: Modal agreement (e.g., stance, gender, race)
- Across-models overlap

# Validate at prompt level + Human Validation

(e.g. [Garg and Fetzer, 2025, Nature Human Behaviour](#))

- ✓ Most convincing
- ✓ Use gold-standard labels, ideally human labelled.

Table 11: Evaluation Metrics for Stance Detection

Task	Target	GPT 3.5 Turbo ( $F_{avg}$ )	GPT 4 ( $F_{avg}$ )
A	Feminism	92.44	81.89
A	Hillary Clinton	89.57	87.53
A	Abortion	79.52	84.36
B	Donald Trump	84.18	80.00

This table reports the  $F_{avg}$  scores (Supplementary Equation 1) for stance detection tasks using GPT-3.5 Turbo and GPT-4. Validation was conducted using 40,317 hand-labeled tweets from the ACM SemEval-2016 Task 6 dataset.<sup>1</sup> Each task involves predicting stances (pro, anti, neutral) toward targets such as Feminism, Hillary Clinton, and Abortion. Results indicate strong performance, with  $F_{avg}$  scores ranging from 79.52 to 92.44 for GPT-3.5 Turbo and 80.00 to 87.53 for GPT-4. Higher  $F_{avg}$  scores denote better alignment with human-labeled stances, highlighting the accuracy of the GPT-based stance detection methodology.

# Validate at prompt level + Automated Validation

(e.g. [Garg and Fetzer, 2025, Nature Human Behaviour](#))

- ✓ Complementary to human validation
- ✓ Cheap to do, and provides inter-LLM stability

Table 12: Comparison of Agreement and F1 Scores Across GPT Models

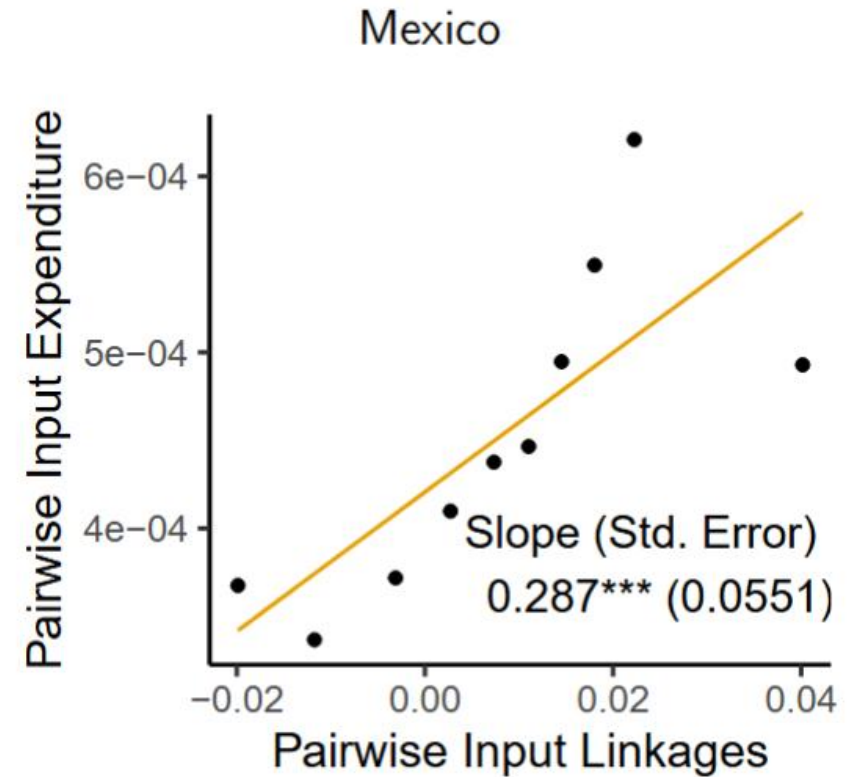
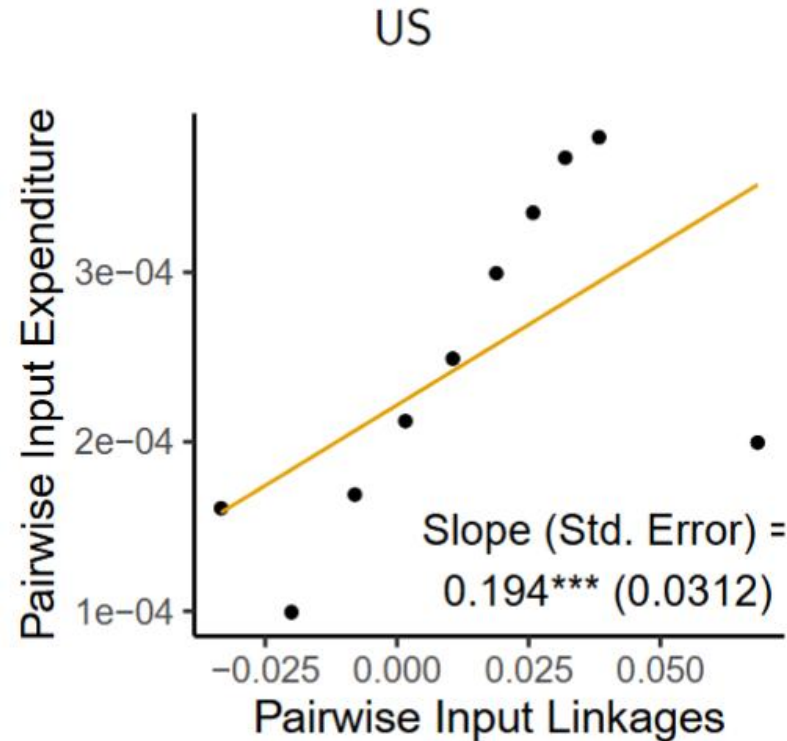
Comparison	Agreement (Modal)	Agreement (Iterations)	F1 (Modal)	F1 (Iterations)
GPT-3.5-turbo vs GPT-4o	0.781	0.772	0.806	0.795
GPT-3.5-turbo vs GPT-4	0.750	0.738	0.772	0.756
GPT-3.5-turbo vs GPT-4o-mini	0.684	0.681	0.696	0.691

This table compares agreement rates and average  $F_{avg}$  scores (Supplementary Equation 1) between different GPT models for stance detection. Agreement metrics include: **Modal Agreement**, the proportion of identical stance predictions when using the modal stance across 10 iterations per tweet, and **Iteration Agreement**, which measures agreement across individual iterations.  $F_{avg}$  scores assess precision and recall consistency, with higher values indicating better model alignment. Results highlight strong consistency across models, particularly between GPT-3.5 Turbo and GPT-4o ( $F_{avg} = 0.806$  for Modal Agreement). These findings demonstrate the robustness of GPT-based stance detection, even across variations in model architecture.

# Validate at measurement level (e.g. [Fetzer et al., 2024](#))

- ✓ Compare your measurement with next-best available alternative
- ✓ Ideally, this alternative is well-used and well-known
- ✓ In our case, it is the industry level, rather than product level, input/output table.

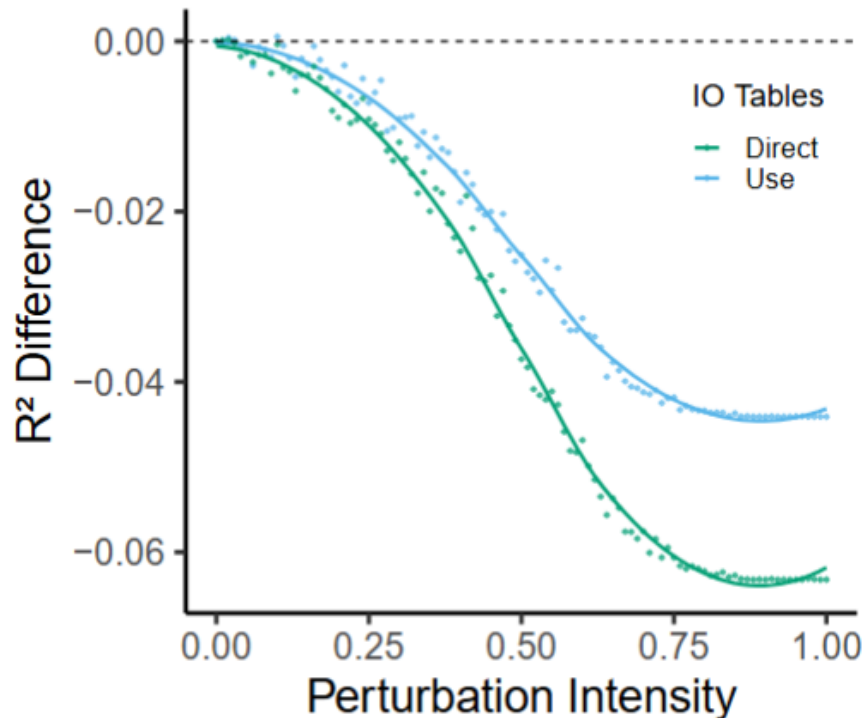
## AIPNET strongly correlates with Input-Output Tables



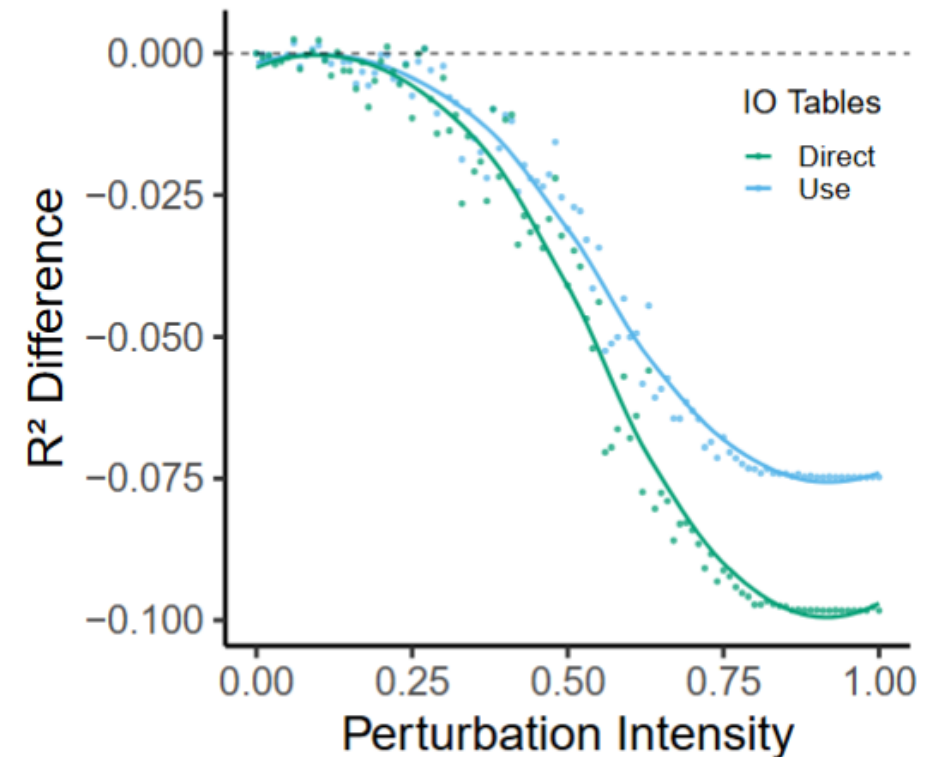
# Validate at measurement level (e.g. [Fetzer et al., 2024](#))

## Perturbing network attenuated validation estimates

US



Mexico



- ✓ If you add noise to your measurement, does it lose signal?
- ✓ If yes, then your new measurement is not totally random!
- ✓ It is a lower bar to clear but still adds some confidence.

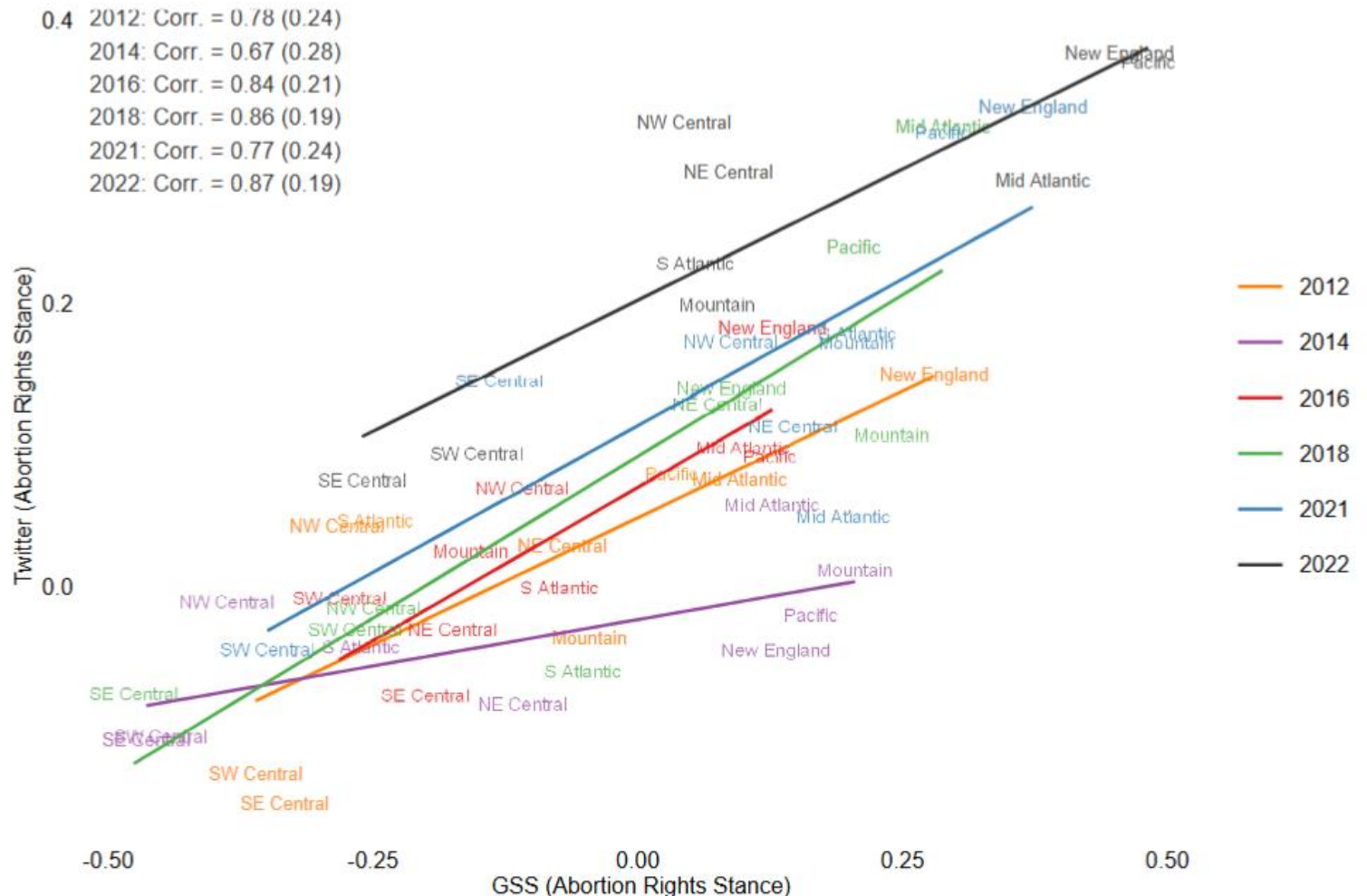


# Validate at Measurement Level

(e.g. [Garg and Fetzer, 2025, Nature Human Behaviour](#))

- ✓ Generally, next best available alternative is easy to access
- ✓ However, it has limitations, e.g. it is available for a given time period or region.
- ✓ You might have to subset your LLM output to that context
- ✓ You might have to do a bespoke LLM exercise for it

Spatio-temporal correlation between Twitter and GSS stance on Abortion Rights

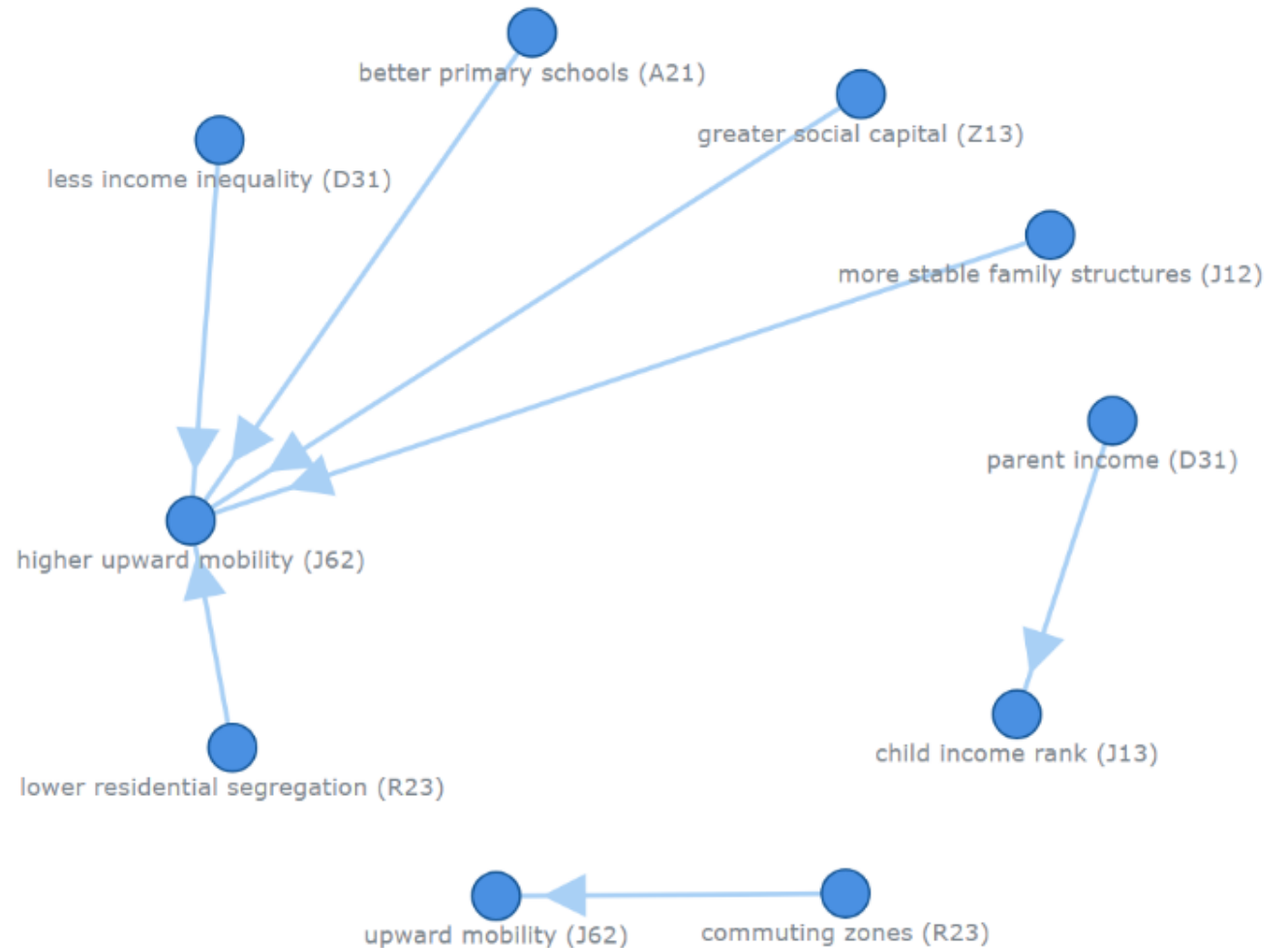


# Eye-ball validation (e.g. [Garg and Fetzer, 2025](#))

*Where is the Land of Opportunity? The Geography of Intergenerational Mobility in the United States (2014, QJE)*

*Raj Chetty, Nathaniel Hendren, Patrick Kline, Emmanuel Saez*

- ✓ Look at (and show) some well-known examples to see if they make sense





# Conclusion

1. Same API, many paradigms (generation, retrieval, classification).
2. Small-scale demos scale linearly with batching + schema enforcement.
3. Embeddings unlock alignment to standardized codes.
4. Simple aggregation heuristics (majority vote, overlap) go a long way.
5. Always build validation and ethics checks into the pipeline.

# Questions?

- Your use cases?
- Any others ideas for application?
- Please share your experiences

Get in touch: [prashant.garg@imperial.ac.uk](mailto:prashant.garg@imperial.ac.uk)

[www.prashantgarg.org](http://www.prashantgarg.org)

# Links to my LLM works:

- Retrieval/Classification:

1. [Leveraging large language models for large-scale information retrieval in economics](#) (a VoxEU article)
2. [Causal Claims in Economics](#). (Non-technical summary)
3. [On Bob Dylan: A Computational Perspective](#)
4. [Artificial Intelligence health advice accuracy varies across languages and contexts](#)

- Generation:

1. [AI-Generated Production Networks](#) (Non-technical summary)

- Retrieval/Classification + Generation:

1. [Political Expression of Academics on Social Media](#) (Non-technical summary)
2. [Politicized Scientists: Credibility Cost of Political Expression on Twitter](#)

# Some resources on use of LLMs in Economics

- Stephen Hansen's Github (code and review article): [Text Algorithms in Economics](#)
- Melissa Dell's JEL article: [Deep learning for Economists](#)
- Anton Korinek's JEL article: [Generative AI for Economic Research: Use Cases and Implications for Economists](#)
- Experimenting with LLM: [Automated Social Science: Language Models as Scientist and Subjects](#) and other works by [John Horton](#)