Session 6.3

GRAM: Generative Reasoning and Augmented Modeling for Quantitative Macroeconomics



Redistribution prohibited without permission

Zhigang Feng

© Zhigang Feng

This deck describes a system that is work in progress by Zhigang Feng, Jianjun Miao, Haoxuan Xie, Yaolong Zhong, and Yingli Zhou (2026). Its evaluation is a design; no evaluation result is reported here.

Session 6.3

What We Do

What this paper is about, four ways to give a model context, a named basis, and the pilot



© Zhigang Feng · Redistribution prohibited without permission

What This Paper Is About

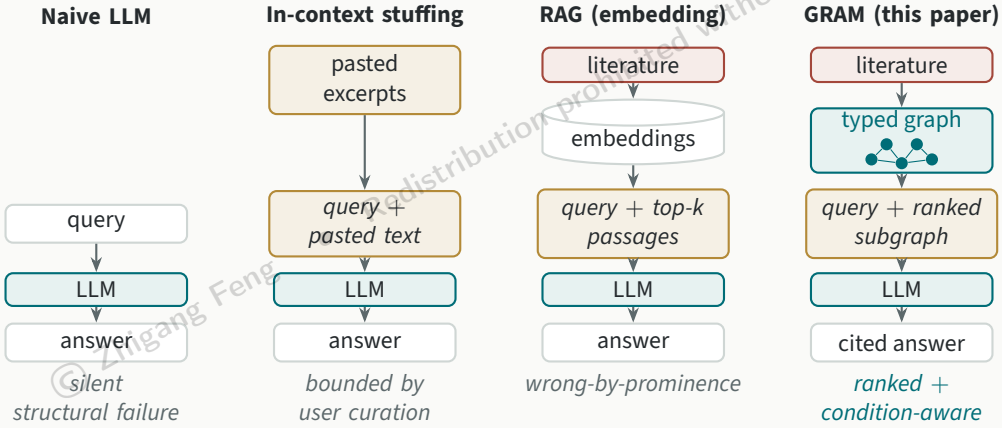
We build **GRAM**: it grounds an LLM's reasoning, answers, and code in the field's **own literature**, built **once** into a searchable knowledge base.

- **The core idea — PromptEcon**: read each paper against a **fixed basis**, a standard for describing any structural macro model. Every paper lands on the **same coordinates** and links to the rest; with a citation graph, this forms one searchable knowledge base. *(Built in 5 stages: parse → citation graph → model extraction → semantic citations → embed.)*
- **At query time the context helps two ways**: *(a)* steer what the model *already* knows; *(b)* add what it does *not* know (post-cutoff papers, exact identities). Because the basis is typed, retrieval is **condition-aware** — a method comes with the conditions under which it holds.
- **Pilot on OLG, but not specific to it**: a reproducible recipe for turning *any* structured economics literature into a knowledge base for AI-augmented research.

© Zhiheng Feng · Redistribution prohibited without permission

Same Query, Four Context Mechanisms

"How do we solve for the recursive competitive equilibrium of this OLG model?"



© Zingang Feng — Unauthorized distribution prohibited without permission

The Four Mechanisms, Side by Side

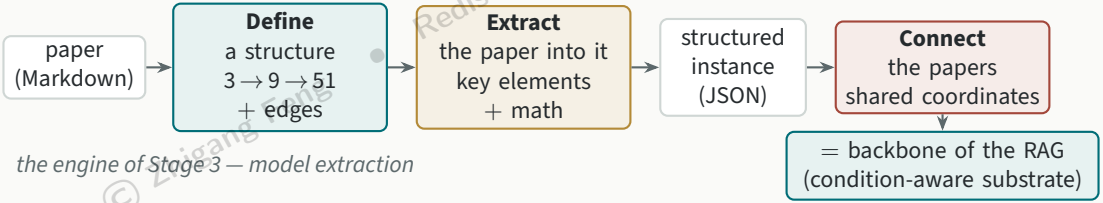
Property	Naive LLM	In-context stuffing	RAG (embedding)	GRAM (this paper)
Fact lookup	Partial	Partial	Yes	Yes
Compositional retrieval	No	Partial	Partial	Yes
Dependency-aware retrieval	No	User-supplied	No	Yes
Condition-aware retrieval	**No**	**No**	**No**	**Yes**
Scales to large corpora	—	No	Yes	Yes
Algorithm transfer	Weak	Moderate	Weak	Strong

© Zhigang Feng
Redistribution prohibited without permission

PromptEcon: What It Is, and Its Role

PromptEcon is the engine of the extraction stage (**Stage 3**). Reading each paper with an LLM, it does three things:

- **Defines a structure** — a fixed basis (**3 blocks** → **9 categories** → **51 axes** + 7 typed edges): the same coordinates for every paper.
- **Extracts each paper into it** — key elements (prose + inline math) become one structured instance.
- **Connects the papers** — shared coordinates let papers align by typed concept: the **backbone of the RAG**.



the engine of Stage 3 — model extraction

© Zhirang Zone
Redistribution prohibited without permission

*This basis is what makes retrieval **condition-aware** (the bold row earlier). Built once, in Stage 3.*

Why 51 Axes? Projecting a Paper onto a Fixed Basis

	an asset	a paper
named basis	Fama–French factors: market, size, value	the 51 axes: `preferences`, `technology`, …
statistical basis	PCA: components 1, 2, 3, …	the 4096-d embedding: $v_1, \ldots, v_{4096}$

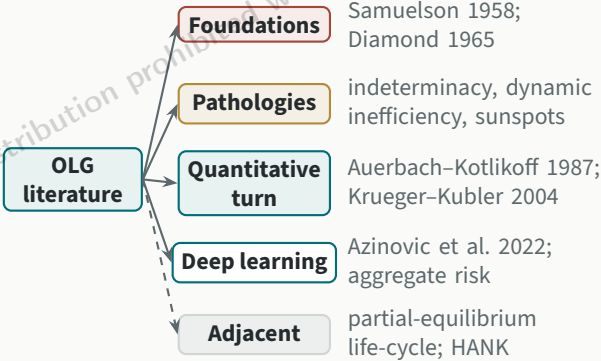
© Anyang Feng · Redistribution prohibited without permission

- The 51 axes are a **fixed basis**: extraction projects each paper onto the *same* named coordinates, so a whole paper becomes a short **coordinate vector**.
- An **empty axis** is a zero coordinate — still information: the model has no such ingredient.
- **Named, like Fama–French's factors: chosen by theory, so every coordinate is interpretable and auditable, and any two papers compare axis by axis.**

Our Pilot: Overlapping-Generations (OLG) Models

Why OLG?

- **Active frontier:** large OLG models with aggregate risk are still an open computational problem.
- **Crisp structure:** each ingredient is forced by a pathology that representative-agent models lack.
- **Author expertise:** we can validate every extraction; the corpus is curated, not scraped.



© Zhigang Feng · Redistribution prohibited without permission

What the Infrastructure Does: GRAM in Two Modes

The same knowledge base supports two modes of help — the two halves of GRAM:

Generative Reasoning
The LLM reasons over a *retrieved typed subgraph* to answer field and research questions — grounded and cited.

Augmented Modeling
The same knowledge base scaffolds *designing algorithms, writing, and validating* model code.

- Not a generic chatbot that paraphrases textbooks — an assistant that **respects the structure** of the field.
- Every paper is **math** (analytical scaffold) *and* **prose** (model anatomy, mechanisms); the assistant reads both.
- Operationalised by a **structured graph** over a **curated literature** — not a flat embedding store.
- **Pilot:** OLG in quant macro; the recipe generalises to other fields of economics and social science.

Nothing like this exists for economics today.

© Zixiang Feng · Redistribution prohibited without permission

Where This Sits

Two research lines retrieve over documents that are mostly *math* or mostly *prose*; economics papers are both.

Math information retrieval	Domain social-science RAG	This paper (econ)
Pure-math papers (Ju et al. 2026, …)	Qualitative prose (Legal: Chen et al. 2026, ACL; Medical: variants)	*Both at once.*
Unit: theorem / lemma / equation.	Unit: case / statute / clinical note.	Unit: typed economic axes, equations carried inside.
Structure: emergent from embedding similarity.	Structure: typed graph nodes (case, statute, …) inject domain knowledge.	Structure: typed graph; domain knowledge from textbook canon.
Strong on math, blind to prose.	*Strong on prose, no math.*	*Math is analysis; prose is mechanism.*

One extraction carries both — each populated axis holds the prose anatomy with its math inside. No direct analogue in economics today.

© Zhigang Feng Redistribution prohibited without permission

Session 6.3

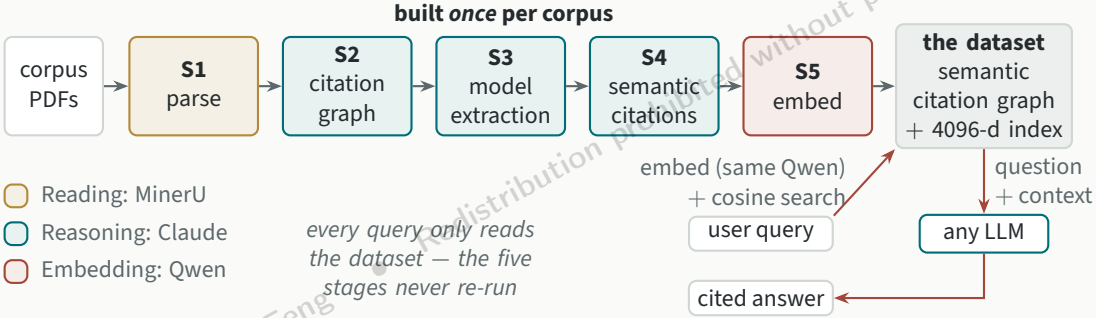
Infrastructure Development

A one-time build: three LLMs, a curated corpus, five stages, and a basis with names on its axes



© Zhigang Feng · Redistribution prohibited without permission

One-Time Build, Then Unlimited Queries



© Zhitang Feng · Redistribution prohibited without permission

The dataset is built **once** per corpus by the five-stage pipeline (S1–S5); thereafter every query only *reads* it — never re-running extraction.

Three LLMs do the building, each in a distinct role.

Three LLMs, Three Roles

The pipeline calls three different LLMs, each for what it is best at; they never substitute for one another.

Role	Model	Modality	Maps
Reading	MinerU	vision–language	page *images* → structured Markdown
Reasoning	Claude	generative (text→text)	text → structured knowledge / prose
Embedding	Qwen3-Embedding-8B	embedding (text→vector)	text → 4096-d vector

- **Reading** (S1): reconstructs each PDF page — OCR, layout, reading order, tables, equations → LaTeX.
- **Reasoning** (S2–S4): the judgment work — resolving citations (S2), **extracting each paper onto the 51 axes** (S3), and summarising how each paper uses what it cites (S4).
- **Embedding** (S5): maps every chunk — and, at query time, the question — into one shared 4096-d space.

(C) Zhigang Feng • Redistribution prohibited without permission

Curation: Tiered Selection of the OLG Corpus

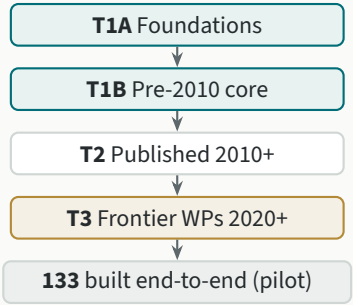
Three design pressures shape the list: **coverage vs. focus • pre- vs. post-cutoff • network connectivity**.

Tier 1A — Foundations. Pre-1980 load-bearing papers anchoring the theory.

Tier 1B — Pre-2010 core. Mechanism and computation pillars: pathologies, numerical methods, bequests, inequality.

Tier 2 — Published 2010–present. Top and field-journal papers spanning the full quant-macro stratum.

Tier 3 — Frontier WPs 2020+. NBER, SED / Summer Institute, arXiv, journal in-press, and job-market scans.



We treat corpus construction as engineering, not bibliography.

© Zheng Feng • Redistribution prohibited without permission

Stage 1: Document Parsing (the Reading LLM)

Economics PDFs are not plain text: two-column layouts, displayed equations, tables, and figures carry the model, and a naive text dump scrambles reading order and drops the math. So we *read* each page with a vision–language model. MinerU's `vlm-auto-engine` looks at the rendered *page image* and reconstructs it as machine-readable Markdown — OCR, layout and reading-order recovery, tables, and equation → LaTeX, in one pass. Output per paper:

- `<id>.md` — layout-aware Markdown, LaTeX math preserved **(used by Stages 3–4)**.
- `<id>_middle.json` — structured layout: pages → typed blocks, reading order, and the parsed reference list **(used by Stages 2–3)**.
- `_content_list_v2.json` / `_model.json` — intermediate VLM state, re-runnable; `_layout.pdf` for visual QA; `*.jpg` images for extracted figures.

© Zhigang Feng · Redistribution prohibited without permission

Stage 2: Citation Graph (the Reasoning LLM)

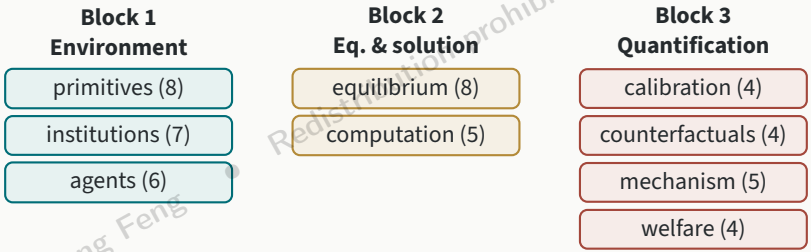
Who builds on whom, across the corpus. For every paper we take its reference list and ask: which of these references are themselves *in our corpus*? An LLM matches each messy reference string to a corpus paper (or to none), giving a clean "*X* cites *Y*" graph over the corpus:

1. One Claude subagent per citing paper — fresh context, parallelisable.
2. Each resolves every listed reference to an in-corpus paper id (or none): bibliographic **0/1 edges**, bare ids only.
3. The script assembles the responses into `citation_graph.json` — a directed adjacency over the N papers, i.e. an $N \times N$ matrix.

© Zhigang Feng • Redistribution prohibited without permission

Stage 3: A Standard Basis for the Field

Stage 3 runs **PromptEcon**: it reads each paper against a **fixed basis** — a standard for describing any structural macro model. Its $m = 51$ **axes** are named coordinates; extraction *projects the paper into this m-dimensional space*, populating only the axes it has:

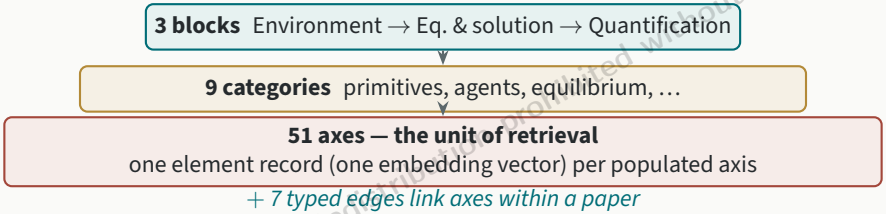


© Zhigang Feng Redistribution prohibited without permission

9 categories group the 51 axes across 3 blocks. A paper fills each axis at most once — one short titled statement, with its equations written inline — and an axis left empty is itself informative (the model simply has no such ingredient).

The Framework: Its Building Blocks

Stage 3 reads each paper against this fixed structure:



3 blocks Environment → Eq. & solution → Quantification

↓

9 categories primitives, agents, equilibrium, …

↓

51 axes — the unit of retrieval
one element record (one embedding vector) per populated axis

+ 7 typed edges link axes within a paper

Term	#	Definition
Block	3	Environment → Equilibrium & solution → Quantification.
Category	9	finer grouping inside the blocks (primitives, agents, …).
Axis	**51**	the **unit of retrieval**: one `element` record per populated axis (one embedding vector each).
Typed edge	7	within-paper dependency between axes (`defines`, `parameterizes`, `clears_in`, …).

© Zhigang Feng • Redistribution prohibited without permission

The Basis, All 51 Axes: Block 1 (Environment, 21)

Category	Axes
primitives (8)	`time`, `demography`, `type_space`, `preferences`, `endowments`, `technology`, `exogenous_shock_process`, `state_space`
institutions (7)	`market_structure`, `friction`, `borrowing_constraint`, `government_rule`, `monetary_regime`, `contract`, `corporate_structure`
agents (6)	`household_problem`, `firm_problem`, `government_problem`, `intermediary_problem`, `planner_problem`, `information_structure`

*Block 1 captures **what the world contains** (primitives), **the rules of the game** (institutions), and **who optimises** (agents) — the Hurwicz–Myerson three-way split.*

© Zhigang Feng · Redistribution prohibited without permission

The Basis, All 51 Axes: Blocks 2 and 3 (30)

Block	Category	Axes
Equilibrium & solution	equilibrium (8)	`state_vector`, `policy_function`, `price_function`, `optimality_condition`, `market_clearing`, `distribution_evolution`, `transversality_condition`, `equilibrium_concept`
Equilibrium & solution	computation (5)	`solution_method`, `discretization`, `distribution_approximation`, `accuracy_diagnostic`, `numerical_implementation`
Quantification	calibration (4)	`parameter`, `target_moment`, `estimation_procedure`, `data_source`
Quantification	counterfactuals (4)	`benchmark_economy`, `alternative_regime`, `comparison_metric`, `transition_path`
Quantification	mechanism (5)	`causal_channel`, `structural_argument`, `comparative_static`, `theoretical_result`, `intuition_narrative`
Quantification	welfare (4)	`welfare_metric`, `welfare_evaluation_point`, `welfare_decomposition`, `welfare_result`

© Zhigang Feng
Redistribution prohibited without permission

Basis Design: How Big, and Where It Comes From

- **How big?** $m = 51$ is a deliberate *dial*, not a limit — a larger basis gives finer resolution at higher curation cost; the precedent is ImageNet (Deng et al. 2009), whose finite, human-curated label inventory was enough to transform computer vision.
- **Where from?** The 9-category split is assembled from five canonical treatments — Arrow–Debreu GE, recursive macro (SLP), MDPs (Bertsekas), mechanism design (Hurwicz–Myerson), and the quantification canon — so every category and axis is auditable by reference.

© Zhigang Feng — Redistribution prohibited without permission

Per-Paper Instance: One JSON, Fully Provenanced

Each paper produces one `<id>.json`: a list of `element` records (one per populated axis) plus an `edges` array; each `element` becomes one embedding chunk in Stage 5. Example below from the OLG run (`abel_panageas_2023`, abridged).

```
{ "model_id": "abel_panageas_2023",
  "elements": [
    { "kind":           "market_clearing",
      "category":       "equilibrium",
      "canonical_name": "capital-market clearing (saving = capital + bonds)",
      "description":    "Young saving finances all assets, so capital is
                         saving net of bonds: $K_{t+1} = S_t - B_{t+1}$.",
      "source_lines":   [[612, 619]] } ],
  "edges": [
    { "from": "government_rule_001", "to": "market_clearing_001",
      "type": "enters_constraint" } ] }
```

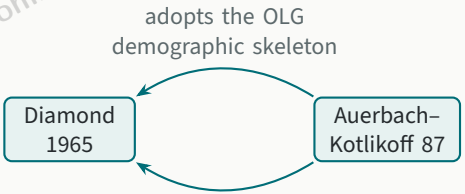
© Zhiqiang Feng • Redistribution prohibited without permission

`kind` names the **axis** (1 of 51); `category` its group (1 of 9); `description` (prose + inline math) is the **only field embedded**; `source_lines` give provenance; `edges` link the paper's axes by one of the 7 types.

Stage 4: Semantic Citations, How *X* Uses *Y*

Stage 2 gives only the bare "*X* cites *Y*" graph. Stage 4 fills in *how* each citation is used — adding no new edges.

- **What:** for each citation, a one- to two-sentence note — does *X* adopt *Y*'s model, borrow its method, calibrate to it, or benchmark against it?
- **How:** an LLM reads the sentences around the citation in *X* and writes that note; one pair can get several notes.
- **Why natural language** (not a fixed label): the note becomes a search vector *as-is*, so a citation is retrievable by meaning — a one-word label like "extends" cannot be, and hides *what* is extended.



multi-edges per pair, each with its own natural-language summary.

© Zhigang Feng · Redistribution prohibited without permission

Stage 5: Embedding (the Embedding LLM)

Goal: make the extracted pieces searchable. We turn each `element` record and each citation note into one vector — so the corpus becomes **machine-readable and indexed**; similar economics lands at nearby vectors, and a query then retrieves **by meaning, not keywords**. The unit we embed is the axis (one `element` record), not the whole paper or section.

- **Model:** Qwen3-Embedding-8B turns each chunk into one **4096-d** vector, length-normalised so a similarity score is just a cosine (a dot product).
- **What a chunk is:** one per `element` record and one per citation note — the same units we later retrieve.
- **Output:** a vector matrix plus aligned metadata (paper, axis, category, provenance); new papers are appended without re-embedding the rest.

*The **same model** embeds the query at retrieval time — so question and corpus share one space.*

© Zhigang Feng · Redistribution prohibited without permission

What Each Stage Produces: From Papers to a Searchable Index

Input: N selected papers. Each stage adds structure:

Stage	Produces	Shape
S1	machine-readable documents	N Markdown / JSON files
S2	citation topology	$A \in \{0,1\}^{N \times N}$
S3	each paper projected onto the basis	up to m element records per paper ($m = 51$)
S4	usage text on every citation	refined $N \times N$ (edges carry summaries)
S5	everything embedded	each element & usage in $\mathbb{R}^{4096}$

© Zhigang Feng • Redistribution prohibited without permission

Where Each Stage Runs

Stage / job	Hardware	Software
S1 parse	T4 GPU	MinerU (vision–language model).
S2–S4 reason	— (API)	Claude.
S5 embed	T4 GPU	Qwen3-Embedding-8B.
Curation / glue	CPU	Python (`llm_econ`).

© Zhigang Feng Redistribution prohibited without permission

Session 6.3

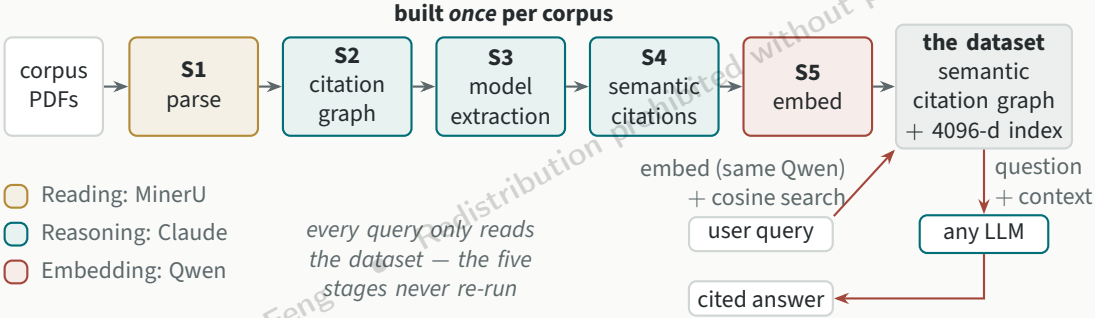
Generative Reasoning and Augmented Modeling

The read-only query loop, two demonstrations, and the recipe in other fields



© Zhigang Feng · Redistribution prohibited without permission

Recap: We Built It Once; Now We Query It



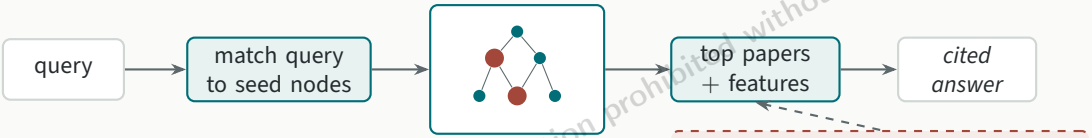
© Yueqing Feng · Redistribution prohibited without permission

The five stages (S1–S5) ran **once** to produce the dataset — a semantic citation graph plus a 4096-d index. Everything that follows is the **read-only query loop** (bottom row): no stage ever re-runs.

Two ways to use it — ask the corpus directly, or ask any research question and let retrieval ground the answer.

At Query Time (1/3): The Intuition

What actually happens when you ask a question:

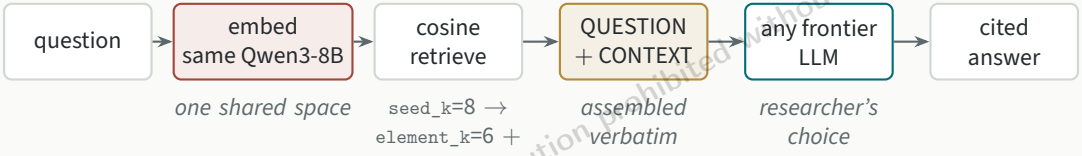


© Zhigang Feng Redistribution prohibited without permission

- The **same** embedder that built the index maps your question into the *one shared space* as the corpus.
- The nearest *typed pieces* — model elements and citation notes — are pulled **by meaning**, not keyword overlap.
- The model answers **only from those pieces**, with citations; off-topic or inadmissible material sits far away and is filtered out *before* the model reads anything.

At Query Time (2/3): The Mechanics

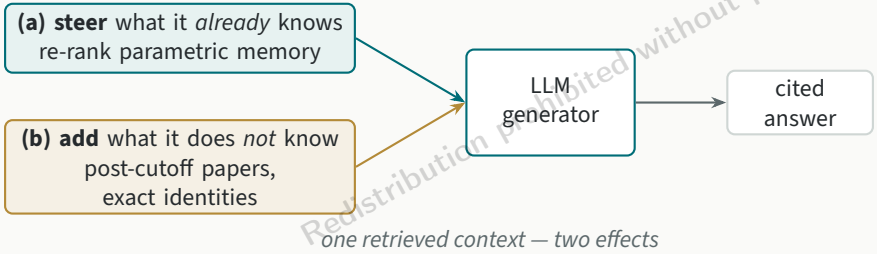
After the one-time build, every query is read-only and deterministic:



Redistribution prohibited without permission
© Zhigang Peng

1. The researcher types a question (`/promptecon "..."`) — a field question, an algorithm to design, eventually code to write.
2. The **same Qwen3-Embedding-8B** embeds it, so query and corpus live in **one semantic space**.
3. Cosine retrieval: the `seed_k`=8 nearest hits seed the search; the context keeps its top `element_k`=6 element + `citation_k`=4 citation chunks (10 in total).
4. A verbatim QUESTION + CONTEXT block goes to **whatever frontier LLM the researcher picks**; the answer is written *from that block alone* — cite-or-decline, with provenance.

At Query Time (3/3): Two Ways the Context Helps



Redistribution prohibited without permission

(a) Steer what the model knows
Targeted context re-ranks the generator's *own* parametric knowledge toward the right region of the field.

(b) Add what it does not know
The same context injects post-cutoff papers, exact accounting identities, and quantitative results absent from its weights.

© Yiqing Xing

Demo 1: Corpus-Targeted Query (the Prompt)

A question that names the corpus directly:

Question. What computational or solution methods do these papers use to solve OLG models with aggregate risk?

Retrieval builds the literal augmented prompt (abridged):

```
CONTEXT (retrieved from the OLG GraphRAG):
[1] nishiyama_smetters_2013 -> hasanhodzic_kotlikoff_2013 (citation)
    state-of-the-art: simulation-projection on an 80-period OLG with aggregate risk.
[3] hasanhodzic_kotlikoff_2013 -> rios_rull_1996 (citation)
    55-period OLG with aggregate shocks solved by local perturbation.
[6] azinovic_yang_cole_kubler_2023 :: solution_method
    simulation-based Krusell-Smith / Kubler-Schmedders with endogenous grid method;
    Euler equations inverted along simulated aggregate paths, then damped iteration.
[10] krueger_mitman_perri_2016 :: solution_method
    Krusell-Smith bounded-rationality Euler solver; occasionally binding constraints.
...  (10 chunks: 6 element + 4 citation, across 9 papers)
```

rank=cosine, seed_k=8, element_k=6, citation_k=4.

© Zhigang Feng • Redistribution prohibited without permission

Demo 1: The LLM's Answer (Verbatim, Abridged)

Answer — written from the augmented prompt alone

The retrieved evidence shows these papers solve aggregate-risk OLG models with a handful of method families:

- **Krusell–Smith / simulation-based Euler** (the dominant approach): `azinovic_yang_cole_kubler_2023` solves the household / firm / market-clearing system *only along simulated paths*, with the **endogenous grid method** inverting the consumption Euler equations; `krueger_mitman_perri_2016` uses the same idea with occasionally-binding constraints.
- **Simulation–projection** and **local perturbation** for large life-cycle models — surfaced *via citations* (`nishiyama_smetters` → `hasanhodzic_kotlikoff`; `hasanhodzic_kotlikoff` → `rios_rull`).
- The shared challenge is carrying the **cross-sectional distribution as part of the aggregate state** (`harenberg_ludwig_2018`: $\Phi' = H(\Phi, z, z')$).
- One paper avoids GE entirely: `pashchenko_porapakkarm_2025` solves a **partial-equilibrium** problem and aggregates by Monte-Carlo simulation rather than a price fixed point.

© Lorenz Feng Redistribution prohibited without permission

Demo 1: Scored Retrieval and What It Shows

Top retrieved chunks (cosine):

#	score	type	source
1	0.827	citation	`nishiyama_smetters` → `hasanhodzic_kotlikoff`
3	0.767	citation	`hasanhodzic_kotlikoff` → `rios_rull`
6	0.646	element	`azinovic_yang_cole_kubler_2023 :: solution_method`
10	0.623	element	`krueger_mitman_perri_2016 :: solution_method`

© Zheng Feng · Redistribution prohibited without permission

What this shows

Citation edges top the ranking (0.83–0.77) above every element chunk (0.65–0.62) — the citation graph surfaced the best evidence; `auerbach_kotlikoff/storesletten_…` are citation-only, outside the 133 papers.

Demo 2: A Plain Research Question (the Prompt)

A question you would put to any macroeconomist — with *no mention of the corpus*:

Question. If a government runs persistent deficits and issues more bonds, does that reduce the economy's capital stock?

Retrieval silently supplies the exact accounting and mechanisms (abridged):

```
CONTEXT (retrieved from the OLG GraphRAG):
[10] abel_panageas_2023 :: market_clearing
     capital = saving net of bonds:   K_{t+1} = S_t - B_{t+1}
[8]  diamond_1965 :: causal_channel
     debt-service taxes cut young saving; internal debt competes for portfolios
     -> capital stock falls, interest rate rises.
[1]  azinovic_yang_cole_kubler_2023 :: welfare_result
     calibrated economy: government debt is never Pareto-improving; crowds out capital.
...  (10 chunks: 6 element + 4 citation, across 3 papers)
```

© Learning ● Redistribution prohibited without permission

Demo 2: The LLM's Answer (Verbatim, Abridged)

Answer — written from the augmented prompt alone

Yes — in these OLG models government debt mechanically **crowds out capital**, and the retrieved evidence pins down both the accounting and the economics:

- **The accounting is explicit.** `abel_panageas_2023` states capital-market clearing as $K_{t+1} = S_t - B_{t+1}$: for given saving, more bonds mean less capital, one-for-one.
- **The channels** (`diamond_1965`): debt service needs *taxes* that cut the young's saving (supply side); internal debt competes on the *demand* side — interest rate up, capital down.
- **Whether that is *bad* hinges on dynamic efficiency.** `samuelson_1958` / `abel_mankiw_summers_zeckhauser_1989` (cited) establish debt can *raise* welfare in dynamically inefficient economies.
- **Quantitatively**, `azinovic_yang_cole_kubler_2023` finds debt is **never Pareto-improving** in its calibration.

(c) Zhigang Feng. Redistribution prohibited without permission

Demo 2: What This Shows (Retrieval Honesty)

The answer is grounded *solely* in what retrieval pulled — so it is only as balanced as the retrieved prompt.

A one-sided pull

The **Ricardian-equivalence** counterview (debt is neutral; Barro) did *not* surface — though `barro_1974` is in the corpus, this phrasing pulled the OLG crowding-out tradition instead. That is why `--show-prompt` matters: you can see *what* grounded the answer.

© Zhigang Feng Redistribution prohibited without permission

Generalising the Recipe to Other Fields

The pilot is OLG, but the workflow is not OLG-specific. In quant-macro the field's *paradigm* is:

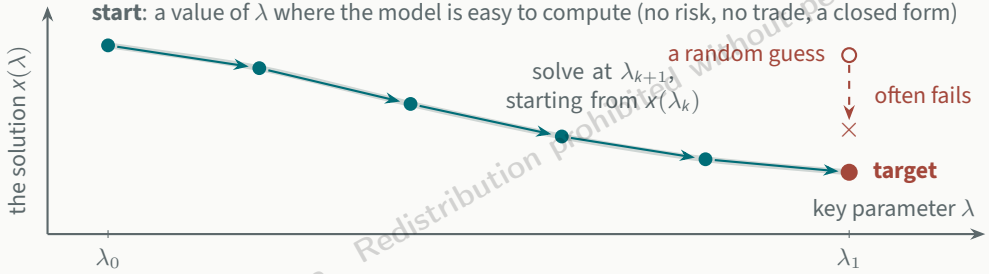
$$\textit{model} \;\rightarrow\; \textit{equilibrium definition} \;\rightarrow\; \textit{propositions \& theorems} \;\rightarrow\; \textit{calibration / estimation}$$

The $9 \rightarrow 51$ PromptEcon framework is the machine-readable form of *that* paradigm: the categories encode the universal recipe, the 51 axes are the populated slots per paper, and the typed edges encode how slots depend on one another. Reusing the recipe in another field requires re-distilling its paradigm first — the category/axis table is the deliverable.

- Other quant-macro subfields (HANK, RBC, Bewley–Aiyagari): same paradigm, different slot values — recipe carries over.
- Other social-science fields: distil that field's paradigm from its textbook synthesis, then specialise the schema.
- Empirical work: paradigm is structurally different (data design, identification, robustness) — re-derive before reuse.

© Zhigang Feng Redistribution prohibited without permission

Homotopy: Compute the Easy Model, Then Move the Key Parameters



Redistribution prohibited without permission

© Zhiyang Feng

- **Start easy:** the value of the key parameter at which the model is easy to compute.
- **Walk the parameter** to its target in small steps: each solve starts from the last solution, and a step that fails is halved and retried.
- **Session 4's OLG lowered the borrowing limit from 0 to -0.05 this way.**

When You Use AI: Five Things to Remember

plan mode
think the problem through first; start simple, then step toward the full one (homotopy)

reflections
make sure the AI has understood the problem, and that its results are verified

gates
a check fixed in advance at every step: verification that stops a mistake early

orchestration
break a complex problem into smaller tasks, and make sure they scale up

gut feeling: domain knowledge still matters — you need to understand the theory and its principles, to know when an answer is wrong

© Zhigang Feng Redistribution prohibited without permission

- **Start simple, check every step, and keep the economics in your own head.**