How Large Language Models Answer Questions About Economic Elasticities

A repeated-elicitation study of prompt-conditioned response distributions

Table of contents

This paper measures the prompt-conditioned response distributions that frontier large language models produce when directly asked about economic elasticities: under a fixed elicitation protocol, what point estimates and uncertainty distributions do the models return? The main dataset contains 11,310 successful runs across 29 models from nine organizations — 11 elicited in April 2026 and 18 in July 2026, in four waves: six frontier updates, the five-model Chinese-lab wave, the GPT-5.6 family, and four late additions (Grok 4.5, Kimi K3, Gemini 3.6 Flash, Claude Opus 5) — over 26 U.S.-scoped quantities (including a capital-gains convention sibling), with 15 runs per model-quantity cell, all under a direction-first v4 prompt that embeds sign-convention clarifiers for the two quantities with documented ambiguity (the clarifier wording was revised to a symmetric if-and-only-if form two days into the April wave: 22 models carry the revised text and seven April models the original, a split the Design section documents; the other 23 quantities are byte-identical across all 29 models). The headline analysis focuses on a nine-elasticity subset of the 13-quantity canonical panel, grouped into labor-and-tax and macro-and-trade subpanels; the panel's four calibration-style parameters enter the stability and robustness checks, and simulation-facing PolicyEngine response coefficients and the capital-gains convention audit appear in separate appendix tables.

Four descriptive findings stand out. First, the model ordering is domain-specific. On the labor-and-tax subpanel, the most elastic models by average within-quantity absolute-value rank are Claude Sonnet 4.6 and Grok 4.5; on the macro-and-trade subpanel, that ordering changes sharply, with Grok 4.3 and Grok 4.20 moving to the top. Second, most labor-and-tax pooled centers fall inside rough review-based benchmark ranges, though the capital-gains realizations elasticity remains the most cross-model-dispersed object within the labor-and-tax subpanel. Third, the capital-gains convention audit — eliciting both parameterizations of the same economic object in parallel under the v4 clarifier — confirms the sign-convention story

largely holds up: 28 of 29 models return a negative w.r.t.-tax-rate elasticity paired with a positive w.r.t.-net-of-tax-rate elasticity, consistent with the identity $\varepsilon_\tau = -\tau/(1-\tau) \cdot \varepsilon_{1-\tau}$. Fourth, two exploratory cross-model cuts: models scoring higher on the PolicyBench policy-calculation benchmark elicit lower taxable-income elasticities (Spearman $\rho \approx -0.5$, raw $p \leq 0.01$, significant under Benjamini-Hochberg at the 0.05 level though narrowly missing familywise Holm significance), and the six Chinese-lab models imply lower optimal top tax rates than the twenty-three US-lab models (medians 32.6% versus 35.9%, exact permutation $p = 0.062$, not surviving family correction), with lab country confounded with serving path, completion budget, and wave. The paper's contribution is methodological and descriptive: it offers a reproducible protocol for eliciting economic parameter distributions from LLMs and documents how those prompt-conditioned distributions vary across models and domains.

1 Introduction

Economists use elasticities constantly. They enter sufficient-statistics formulas, optimal tax calculations, quantitative macro calibrations, and applied policy debates. Yet many of the elasticities that matter most are uncertain, interpretation-sensitive, and contested. If large language models are going to be used as policy assistants, research aids, or informal synthetic experts, it matters what they say when asked for those parameters.

The key object in this paper is the prompt-conditioned elicited response distribution — what a model returns under a fixed protocol, as distinct from how well it forecasts or simulates policy behavior. I ask a model for a quantity such as the Frisch elasticity of labor supply, the elasticity of taxable income, or the Armington elasticity. I require a fixed interpretation, a point estimate, and a distributional summary. Repeating that elicitation many times lets me measure three distinct objects:

- the central estimate the model tends to report
- the uncertainty the model states within a run
- the variation the model exhibits across repeated runs

This is expert elicitation with machine respondents. It differs from LLM uncertainty elicitation on quiz-style benchmarks in its target: a contested economic quantity with no single ground-truth answer, where the disagreement itself is often part of the result.

The initial empirical question is narrow: what distributions do leading models return for a common panel of economic elasticities? The elasticity panel is useful because it includes canonical parameters with clear policy relevance and well-known disagreements in the literature. For the labor-supply and tax subset, the policy reading is straightforward: holding welfare weights fixed, lower behavioral elasticities generally imply more room for redistribution, while higher elasticities generally imply larger efficiency costs of redistribution. That monotone policy mapping holds only for the labor-and-tax subset, so the paper reports labor-and-tax results separately from macro-and-trade results.

This paper therefore contributes on three margins. First, it provides a reproducible design for eliciting probabilistic response distributions from LLMs over economic quantities. Second, it documents systematic cross-model differences in central estimates and pooled predictive uncertainty. Third, it shows that these differences are domain-specific: models rank differently on labor-and-tax elasticities than on macro-and-trade elasticities.

2 Related Literature

This project sits at the intersection of four literatures.

First, it is closest in spirit to expert elicitation over uncertain parameters. In economics, the most directly relevant precedent is the ETI elicitation exercise in Designing, Not Checking, for Policy Robustness (2020), which constructs a subjective distribution over the long-run elasticity of taxable income by combining the empirical literature with a survey of economists. More generally, the decision-analysis literature on quantile-based elicitation and "Quantile-Parameterized Distributions for Expert Knowledge Elicitation" (2024) provides a natural methodological precedent for asking directly for quantiles and then fitting a distribution that preserves them.

Second, it relates to the recent literature on uncertainty elicitation from black-box LLMs. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs (2023) is the clearest baseline for this paper's elicitation design because it explicitly decomposes uncertainty elicitation into prompting, sampling, and aggregation choices. Generating with Confidence (2023) is similarly important because it treats repeated generations as an uncertainty object in their own right, not merely alternative answers.

Third, the paper is adjacent to the growing literature on calibration and overconfidence in LLM uncertainty reports. LLMs Are Overconfident (2025) and Bayesian Elicitation with LLMs (2026) are especially close to the current design because they ask models for numerical intervals and then evaluate coverage and recalibration. Those papers target out-of-sample predictive reliability against realized truths; this paper measures the distribution a model manifests over contested economic parameters, which may admit no single clean benchmark truth.

Fourth, the paper borrows from the distribution-aggregation logic used in forecasting platforms. The closest practical analogy is Metaculus, whose FAQ states that for numeric and date questions the community prediction is a weighted average of individual forecaster distributions (Metaculus 2026). A community prediction differs from "model belief," but it is a useful precedent for aggregating run-level distributions instead of averaging quantiles or endpoints directly.

Finally, review and handbook-style sources anchor the economics side of the project. For labor supply, McClelland and Mok (2012) is the key benchmark source because it synthesizes both income and wage elasticities across groups and margins. Similar review-style anchors appear throughout the registry wherever possible.

3 Design

3.1 Prompting target

The main prompt is a memory-only elicitation prompt. It instructs the model to answer from background knowledge alone — no tools, no external resources, no reconstructing a consensus estimate through a literature review. The prompt fixes the target interpretation of the quantity and requests JSON with:

- interpretation
- point_estimate
- quantiles.p05
- quantiles.p25
- quantiles.p50
- quantiles.p75
- quantiles.p95
- citations
- reasoning_summary

The run artifacts record the verbatim prompt text and a version label for every run, and the analysis never pools materially different prompt families. The current results therefore combine a main memory-only panel with separately labeled robustness reruns when a quantity needed a targeted clarification.

Two quantities in the registry have well-documented convention ambiguity — the prime-age income elasticity (whether an increase in non-labor income raises or reduces hours is a sign, not a magnitude, question) and the capital-gains realizations elasticity (whether the elasticity is taken w.r.t. the tax rate or the net-of-tax rate flips the sign). For both of these, the v4 main prompt embeds a direction-first sign clarifier. In its current form the clarifier is symmetric — "$\varepsilon > 0$ if and only if … / $\varepsilon < 0$ if and only if …" — enumerating both directions without calling either typical or correct, and it lists the $\varepsilon > 0$ clause first in all cases, which is itself a minor ordering-anchoring choice; a full treatment would randomize clause order across runs. The clarifier is not fully neutral — any clause ordering can carry residual anchoring — but it replaces a literature prior with an if-and-only-if definition of what the reported sign means. Other object-definition clarifications — the Armington top-nest clarification and the IES nondurable-consumption clarification — stay out of the main prompt and appear separately as appendix robustness probes because they clarify object definition, not sign.

One disclosed wording split applies to exactly these sign-clarified quantities — the two ambiguous objects plus the capital-gains convention sibling introduced below. The clarifier text was revised once, two days into the April wave. The v4 prompt as elicited on April 19, 2026 stated each direction as a plain conditional ("If additional non-labor income reduces annual hours worked, ε is negative") with the direction the literature treats as typical listed first in all three clarifiers; an April 21 revision rewrote the clarifiers into the symmetric if-and-only-if form above — same

information, without the conventional-direction-first ordering — expanded the magnitude guard into a worked example, and corrected a conversion identity in the net-of-tax sibling's definition line that the original wording stated backwards. Both wordings carry the same v4 label in the run artifacts; the verbatim prompt text is what distinguishes them. Seven of the eleven April models were elicited before the revision and never re-elicited — gpt-5.4, gpt-5.4-mini, gpt-5.4-nano, claude-haiku-4.5, gemini-3-flash-preview, gemini-3.1-flash-lite-preview, and grok-4.1-fast — so their archives carry the original wording on these three quantities. The four April premium-tier models re-ran in full on April 21 for the request-log fix described in the Data section and picked up the revised text, as did every July model. Hashing the archived prompt field per model-quantity cell pins the split exactly: 23 of the 26 quantities are byte-identical across all 29 models, the three sign-clarified quantities each split 22/7 on the same seven models, and every cell is internally uniform (scripts/verify_paper_prose.py recomputes this census from the archives and fails the build if the counts, the seven-model membership, or the majority text's match to the current prompt builder drift). Cross-model comparisons on these three quantities therefore compare answers elicited under two clarifier wordings, with wording confounded with the seven-model April group. The superseded April 19 elicitations of the four premium-tier models — the only cells answered under both wordings — remain in git history, so a within-model wording comparison is reconstructible; Appendix Tables A18-A19 run it. Across those four models the two headline-panel quantities are insensitive to the wording change (pooled centers move by at most 0.08 in absolute value), while the net-of-tax sibling — the one quantity whose definition line carried the backwards identity — moves materially for two of the four models (pooled-center changes of -1.23 and +1.70), shifting their implied-tau audit rows across band boundaries.

An archived GPT-only robustness arm also allowed web search and code interpreter access on an earlier eight-quantity subset. In realized behavior, tool uptake in that arm was negligible: only 9 / 360 requests used web search, and 0 / 360 used code interpreter. I therefore treat the main design as memory-based elicitation and report the tool-access arm only in the appendix.

3.2 Quantities

The current manuscript uses a 26-quantity U.S.-scoped panel: the 9 canonical elasticities, a capital-gains convention sibling, 3 preference/macro objects, 1 TFP-persistence coefficient, and 12 simulation-facing PolicyEngine response coefficients. Headline model-comparison tables cover the canonical elasticity subpanel; the full mixed panel appears in the appendices.

The nine canonical quantities are:

- intertemporal elasticity of substitution
- extensive-margin labor supply elasticity for single mothers
- Frisch elasticity of labor supply for prime-age workers
- income elasticity of labor supply for prime-age workers
- Marshallian wage elasticity of labor supply for prime-age workers

- elasticity of substitution between capital and labor
- capital gains realizations elasticity
- elasticity of taxable income for top earners
- Armington elasticity between imported and domestic goods

Three criteria selected them, fixed in the registry before the v4 panel ran (the registry file's git history predates every v4 run). Each is a standard, named object with a published review anchor or an established calibration home, so elicited answers sit against a literature range rather than an editorial judgment. Each has a direct policy consumer: the six labor-and-tax elasticities are the behavioral parameters CBO-style and PolicyEngine-style tax-benefit microsimulation consumes, and the three macro-and-trade parameters are core calibration inputs whose policy mapping is not monotone. And together the two subpanels cover both a domain where the redistribution reading is monotone and one where it is not, which is what makes the domain-specific ranking result observable. The tilt toward labor and tax — six of the nine — reflects the microsimulation use case, and the panel is U.S.-scoped throughout. Selection determines coverage, not model comparisons: every model answers the identical panel, so the quantity list cannot favor one model over another. Domains the panel omits — health, education, environmental behavior — are natural extensions.

The panel also includes 12 simulation-facing labor-supply response coefficients used in PolicyEngine-style microsimulation, including a global substitution elasticity override, 10 primary-earner substitution elasticities by decile, and a secondary-earner substitution elasticity. These are implementation-facing response parameters, not textbook elasticity objects, so the paper reports them in separate appendix-style tables, apart from the headline cross-model rankings.

One additional row in the current full panel, labor_supply.policy_response.income_elasticity, carries a legacy label. It remains in the full comparison data from the earlier full-panel run, and the headline tables exclude it in favor of the canonical income-elasticity measure.

The registry also carries a capital-gains convention sibling: tax.capital_gains_realizations.elasticity.net_of_tax_rate elicits the same economic object as the canonical capital-gains realizations elasticity, but defined w.r.t. the net-of-tax rate $(1-\tau)$ instead of the tax rate τ. The two conventions are related by $\varepsilon_\tau = -\frac{\tau}{1-\tau}\,\varepsilon_{1-\tau}$, so a model that answers consistently across both conventions should report a negative ε_τ paired with a positive $\varepsilon_{1-\tau}$. The sibling appears only in the dedicated convention-audit appendix table, outside the nine-elasticity headline subpanels. The canonical capital-gains cell keeps the w.r.t.-tax-rate convention, matching the downstream PolicyEngine-US consumer parameter.

3.3 Repeated elicitation and pooling

I elicit each model-quantity cell 15 times. For a given run r, the prompt elicits five quantiles $q_{r,05}, q_{r,25}, q_{r,50}, q_{r,75}, q_{r,95}$. I convert those quantiles into a run-level distribution G_r using a

piecewise-uniform approximation with probability masses $(0.05, 0.20, 0.25, 0.25, 0.20, 0.05)$ on the bins:

- $[L_r, q_{r,05}]$
- $[q_{r,05}, q_{r,25}]$
- $[q_{r,25}, q_{r,50}]$
- $[q_{r,50}, q_{r,75}]$
- $[q_{r,75}, q_{r,95}]$
- $[q_{r,95}, U_r]$

where L_r and U_r are either quantity support bounds or short extrapolations from the outer quantiles when the support is not bounded. For model m and quantity k, the pooled predictive distribution is the equal-weight mixture

$$F_{mk}(x) = \frac{1}{R} \sum_{r=1}^{R} G_{mkr}(x),$$

with $R = 15$ in the main design. The tables report the pooled point estimate as the mean of the run-level point estimates, and the pooled 90 percent interval as $[F_{mk}^{-1}(0.05), F_{mk}^{-1}(0.95)]$.

The paper's default interval object is the pooled predictive interval: it measures the predictive spread implied by the model's repeated elicited answers, not the precision of their mean. The codebase implements REML and Bayesian hierarchical summaries as secondary estimators; the headline tables use the equal-weight pooled predictive mixture throughout.

The choice of $R = 15$ is pragmatic. Appendix Table A1 reports a simple prefix-stability check on the canonical 13-quantity subpanel. Relative to the full 15-run pooled summary, using only the first 10 runs in a cell changes the pooled center by a median of 0.002 and the pooled 90 percent width by a median of 0.010; using only the first 5 runs changes the pooled center by a median of 0.005 and the pooled width by a median of 0.027. Appendix Table A14 attaches Monte Carlo standard errors to the same summaries by resampling the 15 runs within each cell: the median center standard error is 0.008 at the panel median, relative width standard errors run 1 to 9 percent, and each model's average width rank carries a narrow 90 percent resampling interval, so the predictive-uncertainty ordering is not an artifact of run-level noise at $R = 15$. None of this proves convergence, but it bounds the sampling error the headline summaries carry.

3.4 Models

The current main panel includes 29 models from nine organizations. I elicited eleven in the April 2026 rerun:

- gpt-5.4

- gpt-5.4-mini
- gpt-5.4-nano
- claude-opus-4.7
- claude-sonnet-4.6
- claude-haiku-4.5
- gemini-3.1-pro-preview
- gemini-3-flash-preview
- gemini-3.1-flash-lite-preview
- grok-4.20
- grok-4.1-fast

Eighteen more joined in July 2026 under the same v4 prompts (in the revised clarifier wording disclosed above), quantities, and repeated-run design, in four waves. Six frontier updates from the April providers:

- gpt-5.5
- claude-fable-5
- claude-opus-4.8
- claude-sonnet-5
- gemini-3.5-flash
- grok-4.3

Five models from Chinese labs, chosen because the published PolicyBench leaderboard already scores them, which extends the capability-correlates sample (see the cross-model correlates section) and adds five organizations to the panel:

- deepseek-v4-pro (DeepSeek)
- qwen-3.7-max (Alibaba)
- kimi-k2.6 (Moonshot AI)
- glm-5.2 (Zhipu AI)
- minimax-m3 (MiniMax)

And the GPT-5.6 family, released mid-extension:

- gpt-5.6-sol
- gpt-5.6-luna
- gpt-5.6-terra

A fourth July wave adds late releases elicited under the identical protocol as they shipped: grok-4.5 (xAI, July 15), after PolicyBench added it to the leaderboard — which keeps the beliefs panel a superset of the benchmark's frontier chat models — kimi-k3 (Moonshot AI, July 17), gemini-3.6-flash (Google, July 22), which the pinned PolicyBench release scores as well, and claude-opus-5 (Anthropic, July 23), elicited before any PolicyBench release scores it, so it joins the belief panel but not the capability-correlates sample.

The design is intentionally symmetric across organizations: same quantities, same prompt family, same number of repeated runs. Each provider appears at its frontier tier as of its elicitation date; superseded models remain in the panel, which makes within-provider generation-to-generation shifts directly observable. One caution applies to such comparisons: the harness mechanism is confounded with elicitation wave (Appendix Table A16 discloses the full per-model configuration), most sharply for Claude, whose April models ran through a forced-function-call path and whose July models ran through native structured outputs. Appendix Table A17 bounds this concern empirically by re-eliciting an April model (Claude Opus 4.7) under the July mechanism: across the nine canonical elasticities the pooled center moves by at most 0.03 (median 0.00), so mechanism effects appear negligible relative to the cross-model and cross-generation differences the paper describes.

3.5 Generation protocol and exclusions

One generation protocol covers the entire main no-tools panel.

- Models run at temperature = 1.0 where the API accepts a sampling parameter. Claude Fable 5, Claude Opus 5, Claude Opus 4.8, and Claude Sonnet 5 reject sampling parameters, so their requests carry none and repeated-draw variation comes from default sampling.
- OpenAI models use Chat Completions with strict JSON-schema output and batched draws up to n = 8 per request for cost efficiency.
- Claude models through Opus 4.7 and all Gemini and xAI models run through LiteLLM with one request per run. Claude (through Opus 4.7) and Grok use forced function-call output; Gemini uses forced JSON-object output.
- The July 2026 Claude additions (Fable 5, Opus 4.8, Sonnet 5, and the late Opus 5) run through the native Anthropic API with strict JSON-schema structured outputs and one request per run. Each keeps its provider-default reasoning configuration: always-on for Fable 5, adaptive for Sonnet 5 and Opus 5, off for Opus 4.8.
- The five Chinese-lab models run through OpenRouter via LiteLLM in forced JSON-object mode with local schema validation, one request per run. The GPT-5.6 family runs through the same OpenAI Chat Completions path as the other GPT models.
- The nominal completion budget is 1200 tokens per run for the April panel. Models whose reasoning tokens count against the completion budget receive more headroom (4000 tokens for gemini-3.5-flash and grok-4.3; 8000 for the GPT-5.6 family, grok-4.5, gemini-3.6-flash, and five of the six Chinese-lab models; 16000 for glm-5.2, whose 8,000-token pilot returned empty responses consistent with reasoning exhaustion; 32000 for the native-path Claude models); the budget is a truncation guard, not part of the elicitation content.
- Each run is an independent request with no conversational carryover, no tool access, and no manual repair of malformed outputs.
- A run counts as successful if and only if it returns machine-parseable structured output; failed runs stay missing and drop out of the pooled summaries.

- Appendix Table A16 discloses the full per-model harness configuration — provider path, output mechanism, completion budget, sampling regime, reasoning configuration, and API identifier — in one place. Twenty-eight of twenty-nine identifiers are floating aliases; only claude-haiku-4.5 pins a dated snapshot. Re-elicitation at a later date may therefore hit updated model builds, and the elicitation dates above scope every result.

4 Data Collected So Far

The underlying no-tools v4 data collection contains:

- 29 models from nine organizations (11 elicited April 2026, 18 elicited July 2026 in four waves)
- 26 quantities (9 canonical elasticities plus 1 capital-gains convention sibling, 3 preference/macro objects, 1 TFP persistence, and 12 PolicyEngine simulation-facing coefficients)
- 15 runs per model-quantity cell
- 11,310 successful main-panel runs at 100% parse rate across all cells, verified by an exact-grid checker (scripts/check_panel_grid.py) that every cell holds exactly 15 parsed runs

Success rates are uniformly 100% under v4: every planned run for every model-quantity cell parsed cleanly into structured JSON. Under the earlier v3 panel a single model (grok-4.20) had a 12% structured-output failure rate that affected 38 runs; that failure mode did not recur in the v4 rerun. In the July 2026 extension, 58 runs (2.3% of the extension) initially failed on infrastructure errors — empty responses from gpt-5.5 exhausting its completion budget on reasoning, missing forced tool calls from grok-4.3, three claude-sonnet-5 responses exceeding a 32,000-token output cap, one transient schema-compilation error, and one safety-classifier false positive. I re-elicited those slots as fresh independent draws under the identical prompts. A per-cell manifest (results/failure-manifest.csv) records every replaced slot with its error class and replacement request IDs, and the re-elicitation script now archives failed records before replacing them (this archiving postdates the July round, whose audit trail is the manifest plus the retained request logs; because the July replacement happened in place, a per-slot include-versus-exclude sensitivity is not reconstructible for that round). The failures are infrastructure artifacts, not content, but two caveats keep the claim honest: truncation-type failures correlate with response length, so replacement is not provably independent of elicited values; and the two gpt-5.5 capital-gains cells failed in full at the original 1,200-token budget and reran entirely at an 8,000-token budget, a disclosed within-cell protocol change. All other affected cells replaced at most 5 of 15 runs — under identical settings for the claude-sonnet-5 and grok-4.3 slots, and at the same raised 8,000-token budget for the two partially affected gpt-5.5 cells.

The five Chinese-lab models required a longer recovery. Their OpenRouter elicitation hit an account credit ceiling partway through the first pass, and two later passes ran through local network outages that returned DNS failures; every failed slot was re-elicited as a fresh

independent draw under the identical prompt, and the re-elicitation script archived all replaced records — 4,168 failure records across the five final-panel model directories (failed-runs-archive.jsonl per directory; the archived glm-5.2 pilot described below holds 151 more, and the later grok-4.5, kimi-k3, and claude-opus-5 additions archived 1, 5, and 2 re-elicited slots under the same protocol) — before an exact-grid checker verified 15 parsed runs in every cell. Two disclosures from that recovery: glm-5.2 returned empty, no-JSON-content responses — consistent with reasoning exhausting its 8,000-token completion budget, the failure class the larger budget eliminated — in 109 of its first 390 draws, so its entire main panel reran fresh under a single 16,000-token protocol (the mixed-budget first pass is retained as a pilot archive and enters no analysis), and kimi-k2.6's full 390-run main panel was re-elicited after a network-outage pass replaced its first draws — its published cell therefore comes entirely from the final pass. The failure classes are infrastructure artifacts (credit exhaustion, DNS resolution, empty reasoning-exhausted responses), not content, with the same caveat as above: truncation-type failures correlate with response length, so replacement is not provably independent of elicited values.

Tracked API cost at the model-total level for the April panel, sorted by decreasing spend:

- claude-opus-4.7: $8.345
- claude-sonnet-4.6: $4.277
- grok-4.20: $3.448
- gemini-3.1-pro-preview: $3.315
- gpt-5.4: $1.353
- claude-haiku-4.5: $1.203
- gpt-5.4-mini: $0.377
- gemini-3-flash-preview: $0.370
- gemini-3.1-flash-lite-preview: $0.286
- gpt-5.4-nano: $0.118
- grok-4.1-fast: $0.058

And for the July 2026 additions:

- claude-fable-5: $18.001
- claude-opus-4.8: $5.860
- gpt-5.5: $5.768
- gemini-3.5-flash: $4.484
- claude-sonnet-5: $2.878
- gemini-3.6-flash: $2.993
- grok-4.5: $2.714
- grok-4.3: $0.751

Total tracked v4 cost: $23.15 for the April main-panel rerun (the total excludes April clarify-probe costs for the four per-quantity-fallback models, which predate the cost-aggregation fix; see results/README.md) plus $37.74 for the July frontier additions across their main panel

and clarify probes, $2.71 for the late grok-4.5 addition, and $2.99 for the late gemini-3.6-flash addition, for $66.59 of request-log-tracked spend; the late kimi-k3 and claude-opus-5 additions carry no per-request costs in their logs (kimi-k3 runs through OpenRouter, which does not report them; claude-opus-5 ran before its pricing entry landed in the harness), and their logged token counts imply roughly $10.69 and $11.30 at list prices. The original five Chinese-lab models cost $29.25 at the OpenRouter account level, including the failed calls their recovery replaced; per-request costs for those models and for the GPT-5.6 family are untracked in the request logs and appear as em dashes in every cost column rather than as zeros. Four April premium-tier models (claude-sonnet-4.6, claude-opus-4.7, gemini-3.1-pro-preview, grok-4.20) reran quantity-by-quantity to work around a provider-side SSL hang that appeared only when a single LiteLLM client pipelined many sequential structured-output calls; the per-quantity subprocess fallback preserves request-log aggregation, so cost figures for those four models are tracked end-to-end, not extrapolated. The July additions ran through the same per-quantity path with the fixed cost aggregation. Cost dispersion across the request-log-tracked models is large: the most expensive (claude-fable-5, whose always-on reasoning is billed as output tokens) costs roughly 310x the cheapest (grok-4.1-fast), and per-successful-run cost ranges by more than two orders of magnitude between the cheapest and most expensive tracked cells.

5 Main Descriptive Results

5.1 Rankings depend on domain, not just provider

The main descriptive result is that the model ranking changes once the quantities split into economically coherent subpanels; no single global ordering holds.

Table 1 reports the canonical labor-and-tax subpanel: Frisch, income, and Marshallian labor-supply elasticities; the extensive-margin elasticity for single mothers; the elasticity of taxable income; and the capital-gains realizations elasticity. On this subpanel, the highest-elasticity models by average within-quantity absolute-value rank are Claude Sonnet 4.6, Grok 4.5, and Grok 4.20. The lowest-elasticity models are Gemini 3.5 Flash, GPT-5.5, and Gemini 3.6 Flash.

That ordering is economically interpretable, but only loosely. On this labor-and-tax subset, lower absolute elasticities imply a more redistribution-permissive reading ceteris paribus, while higher absolute elasticities imply larger behavioral costs of redistribution. Under that narrow reading, Gemini 3.5 Flash, GPT-5.5, and Gemini 3.6 Flash sit on the low-response side of the panel, while Claude Sonnet 4.6, Grok 4.5, and Grok 4.20 sit on the high-response side. I treat that ranking only as directional. The more meaningful policy translation is the explicit top-tax exercise below.

Table 1. Labor-and-tax canonical overview.

Note: Canonical labor-and-tax subpanel only: 6 quantities x 29 models x 15 runs. Average ranks are computed within quantity using the absolute value of the pooled point estimate.

Model	Organization	Avg abs-elasticity rank (1=highest)	Avg predictive-uncertainty rank (1=narrowest)	Mean absolute pooled center	Mean pooled 90% width	Success rate	Cost / successful run
Claude Sonnet 4.6	Anthropic	7.75	14.67	0.428	0.978	100.0%	$0.0112
Grok 4.5	xAI	9.5	16	0.387	1.049	100.0%	$0.0066
Grok 4.20	xAI	9.5	22.33	0.362	1.245	100.0%	$0.0092
Qwen 3.7 Max	Alibaba	9.83	22.5	0.429	1.262	100.0%	—
GPT-5.6 Terra	OpenAI	11.5	16.5	0.361	1.02	100.0%	—
GLM-5.2	Zhipu AI	11.75	19.67	0.37	1.406	100.0%	—
Grok 4.3	xAI	12.08	16.67	0.37	1.125	100.0%	$0.0018
Claude Haiku 4.5	Anthropic	12.5	11	0.369	0.911	100.0%	$0.0031
Kimi K3	Moonshot AI	13.42	17.5	0.357	1.035	100.0%	—
Gemini 3 Flash	Google	13.75	9.67	0.382	0.892	100.0%	$0.0009
GPT-5.4	OpenAI	13.92	15.33	0.411	1.243	100.0%	$0.0036
Claude Fable 5	Anthropic	14.42	5.83	0.376	0.812	100.0%	$0.0403
GPT-5.4 mini	OpenAI	14.42	23.17	0.406	1.452	100.0%	$0.0010
Claude Opus 4.7	Anthropic	14.58	9.17	0.394	0.93	100.0%	$0.0221
Claude Opus 4.8	Anthropic	14.67	9.83	0.391	0.902	100.0%	$0.0146
GPT-5.4 nano	OpenAI	15.33	18.33	0.313	1.086	100.0%	$0.0003
Kimi K2.6	Moonshot AI	15.75	19.33	0.324	1.205	100.0%	—
Claude Opus 5	Anthropic	15.83	13.83	0.381	0.986	100.0%	—

Model	Organization	Avg abs-elasticity rank (1=highest)	Avg predictive-uncertainty rank (1=narrowest)	Mean absolute pooled center	Mean pooled 90% width	Success rate	Cost / successful run
DeepSeek V4 Pro	DeepSeek	16	19.17	0.312	1.158	100.0%	—
Gemini 3.1 Pro	Google	16.33	6.83	0.371	0.79	100.0%	$0.0091
GPT-5.6 Sol	OpenAI	16.58	11.83	0.339	0.938	100.0%	—
Grok 4.1 Fast	xAI	16.67	24.5	0.327	1.422	100.0%	$0.0002
Claude Sonnet 5	Anthropic	17.08	8.17	0.367	0.893	100.0%	$0.0068
Gemini 3.1 Flash-Lite	Google	17.08	10.67	0.367	1.022	100.0%	$0.0007
GPT-5.6 Luna	OpenAI	18.75	25.17	0.318	1.554	100.0%	—
Gemini 3.6 Flash	Google	19.83	6	0.348	0.819	100.0%	$0.0080
MiniMax M3	MiniMax	19.83	21.33	0.3	1.154	100.0%	—
GPT-5.5	OpenAI	21.67	8.83	0.328	0.917	100.0%	$0.0159
Gemini 3.5 Flash	Google	24.67	11.17	0.218	1.059	100.0%	$0.0126

Table 2 reports the macro-and-trade subpanel: the intertemporal elasticity of substitution, the capital-labor substitution elasticity, and the Armington elasticity. Here the ordering changes sharply. Grok 4.3, Grok 4.20, and GPT-5.6 Luna move to the top of the ranking, while Claude Opus 4.7 and Claude Haiku 4.5 move to the bottom. The key point is that the ordering is domain-specific; no family is uniformly "more elastic."

The pooled predictive-uncertainty ranking also changes across subpanels. In labor-and-tax, Claude Fable 5, Gemini 3.6 Flash, and Gemini 3.1 Pro are the tightest models, whereas GPT-5.6 Luna, Grok 4.1 Fast, and GPT-5.4 mini are the widest. In macro-and-trade, Claude Haiku 4.5 becomes the tightest model, while GPT-5.6 Luna, Grok 4.20, and Grok 4.3 are the widest. This is pooled predictive uncertainty, not calibrated confidence.

The macro-and-trade subpanel contains only three quantities, so the ranking is easily overturned

by a single object. The leave-one-organization-out robustness (Appendix Table A3) shows Spearman ρ stays above 0.99 even on this subpanel, but read the ranking as directional, not fine-grained. The Armington clarification evidence (Appendix Table A7) reinforces that this subpanel's ordering is particularly sensitive to object definition.

Table 2. Macro-and-trade canonical overview.

Note: Canonical macro-and-trade subpanel only: 3 quantities x 29 models x 15 runs. Average ranks are computed within quantity using the absolute value of the pooled point estimate.

Model	Organization	Avg abs-elasticity rank (1=highest)	Avg predictive-uncertainty rank (1=narrowest)	Mean absolute pooled center	Mean pooled 90% width	Success rate	Cost / successful run
Grok 4.3	xAI	5.83	25	1.509	3.887	100.0%	$0.0018
Grok 4.20	xAI	6.83	26.67	1.257	3.986	100.0%	$0.0089
GPT-5.6 Luna	OpenAI	7.17	27	1.291	4.144	100.0%	—
GPT-5.4 nano	OpenAI	7.33	17	1.256	2.696	100.0%	$0.0003
Claude Opus 5	Anthropic	7.83	17	1.275	2.838	100.0%	—
GPT-5.4 mini	OpenAI	8	23.17	1.241	3.491	100.0%	$0.0009
GPT-5.4	OpenAI	9.33	8.67	1.267	2.393	100.0%	$0.0033
Gemini 3.5 Flash	Google	10.5	13.67	1.153	2.473	100.0%	$0.0104
Grok 4.5	xAI	10.67	23.33	1.344	3.446	100.0%	$0.0065
Kimi K2.6	Moonshot AI	11.67	24	1.11	3.23	100.0%	—
Gemini 3.1 Pro	Google	13.33	13.67	1.27	2.506	100.0%	$0.0091
GPT-5.6 Sol	OpenAI	13.67	17.33	1.107	2.904	100.0%	—
GLM-5.2	Zhipu AI	13.67	18.67	1.102	3	100.0%	—
Claude Sonnet 4.6	Anthropic	14.33	12.83	1.16	2.724	100.0%	$0.0109
GPT-5.5	OpenAI	15.5	15.33	1.121	2.918	100.0%	$0.0101

Model	Organization	Avg abs-elasticity rank (1=highest)	Avg predictive-uncertainty rank (1=narrowest)	Mean absolute pooled center	Mean pooled 90% width	Success rate	Cost / successful run
Claude Sonnet 5	Anthropic	15.83	10	1.433	2.948	100.0%	$0.0064
Gemini 3.6 Flash	Google	16.33	9.67	1.071	2.428	100.0%	$0.0076
Kimi K3	Moonshot AI	16.83	18.33	1.216	3.27	100.0%	—
Gemini 3.1 Flash-Lite	Google	18.17	11	1.164	2.943	100.0%	$0.0007
GPT-5.6 Terra	OpenAI	18.33	7	1.056	2.03	100.0%	—
Claude Opus 4.8	Anthropic	18.83	9.5	1.167	2.737	100.0%	$0.0137
Grok 4.1 Fast	xAI	19.33	12.5	0.907	1.785	100.0%	$0.0001
Claude Fable 5	Anthropic	20.67	7.33	1.041	2.127	100.0%	$0.0397
DeepSeek V4 Pro	DeepSeek	20.67	20	0.999	2.79	100.0%	—
Gemini 3 Flash	Google	20.83	13	0.94	2.632	100.0%	$0.0009
MiniMax M3	MiniMax	22.67	9	1.038	2.217	100.0%	—
Qwen 3.7 Max	Alibaba	22.67	12.67	0.884	2.269	100.0%	—
Claude Haiku 4.5	Anthropic	23.17	2	0.881	1.389	100.0%	$0.0031
Claude Opus 4.7	Anthropic	25	9.67	0.878	2.224	100.0%	$0.0211

5.2 Most labor-and-tax centers lie inside rough review ranges

Table 3 compares the labor-and-tax pooled centers to rough review-based benchmark intervals. These are hand-coded literature anchors — not benchmark truths — that check whether the elicited centers live in the right neighborhood.

For most labor-and-tax quantities, the answer is yes. Every one of the 29 models falls inside the rough benchmark range for the Frisch elasticity. 28 / 29 lie inside for capital gains realizations, the uncompensated wage elasticity, and the extensive-margin single-mother elasticity, and 24 / 29 for the canonical income elasticity. The sign-convention clarifier in v4 eliminated the clear wrong-sign centers that appeared in the earlier v3 panel, though one April model still sits at an economically null income elasticity (gpt-5.4-nano at -0.001). One July addition reintroduces sign instability in a sharper form: gemini-3.5-flash is bimodal on both sign-sensitive quantities, not centered near zero. On capital gains realizations, 10 of its 15 runs are negative (between -0.7 and -0.2) and 5 are large and positive (between +0.4 and +1.3), so the mean center of +0.010 is sign cancellation across modes (the run-level median is -0.40); on the income elasticity, 11 runs are mildly negative and 4 are positive, again yielding a near-zero mean (+0.011) that no individual run states. Nine of the fifteen capital-gains runs and five of the fifteen income-elasticity runs also required quantile repair, concentrated in exactly these cells. Its successor gemini-3.6-flash does not inherit the instability: all fifteen of its runs are negative on both sign-sensitive quantities (income-elasticity center -0.042, just above the band; capital-gains center -0.64, inside it), with zero quantile repairs. The clarifier therefore reduces but does not eliminate sign instability in new model generations — and where instability appears, the next generation can resolve it — while a near-zero mean on a sign-sensitive quantity reads as a symptom of mode-mixing, not a stated belief. gpt-5.5 also centers just above the income-elasticity band at -0.035. ETI remains a quantity with upper-edge pressure: 26 / 29 models fall inside the rough range, with three models just above the review band's upper anchor of 0.5: the same two April models (claude-haiku-4.5 at 0.507, gpt-5.4-nano at 0.552) and qwen-3.7-max at 0.554, the panel maximum. The band itself sits in the upper half of the Saez, Slemrod, and Giertz (2012) survey range of 0.12 to 0.40, extended to 0.5 to reflect higher top-earner estimates (Gruber and Saez 2002); several models (Claude Sonnet 4.6 at 0.500, Grok 4.20 at 0.500, and now Claude Sonnet 5 at 0.473) concentrate at or near that upper anchor. That even the benchmark-consistent models lean toward the top of the review range is itself a finding about the models' elicited ETI distributions.

This benchmark table is useful because it distinguishes two kinds of disagreement. On many quantities, the models disagree with each other but still sit inside a conventional literature neighborhood. On a few quantities, the disagreement includes benchmark outliers. Appendix Table A6 sharpens the same point from a different angle: on every canonical quantity, cross-model spread in pooled centers remains smaller than the average pooled 90 percent width. Within the labor-and-tax subpanel the largest spread-to-width ratio occurs for the capital gains realizations elasticity (0.52); across the full canonical panel, TFP persistence (0.65) and the intertemporal elasticity of substitution (0.55) rank higher still.

Table 3. Rough literature comparison for the labor-and-tax subpanel.

Note: Rough review-based benchmark intervals for the canonical labor-and-tax subpanel. These intervals are hand-coded literature anchors rather than benchmark truths.

Quantity	Rough review range	Models in range	Model min center	Model max center	Benchmark sources
Capital gains realizations elasticity	[-1, -0.2]	28 / 29	-0.93	0.01	Dowd, McClelland, and Muthitacharoen 2015; Burman and Randolph 1994; CBO/JCT medium-run convention
Elasticity of taxable income	[0.25, 0.5]	26 / 29	0.335	0.554	Gruber and Saez 2002 (high-income estimates); Saez, Slemrod, and Giertz 2012 (survey range 0.12-0.40, upper half)
Employment participation elasticity of single mothers	[0.3, 1]	28 / 29	0.213	0.717	Chetty, Guren, Manoli, and Weber 2013 (elasticity implied by Eissa and Liebman 1996); Meyer and Rosenbaum 2001
Frisch elasticity of labor supply	[0.25, 0.75]	29 / 29	0.283	0.593	CBO 2012; Peterman 2016; lifecycle and macro-calibration literature

Quantity	Rough review range	Models in range	Model min center	Model max center	Benchmark sources
Income elasticity of labor supply	[-0.15, -0.05]	24 / 29	-0.107	0.011	CBO 2012; Blundell and MaCurdy 1999; Imbens, Rubin, and Sacerdote 2001 (marginal propensity to earn, converted to an elasticity)
Uncompensated wage elasticity of labor supply	[0.05, 0.3]	28 / 29	0.04	0.168	CBO 2012; Blundell and MaCurdy 1999

5.3 A utilitarian sufficient-statistics top-tax exercise

The labor-and-tax subpanel also permits one concrete policy translation. For the elasticity of taxable income, I map each model's pooled ETI distribution into an optimal top marginal tax rate using the standard sufficient-statistics formula in the top bracket (Saez 2001):

$$\tau^* = \frac{1 - \bar{g}}{1 - \bar{g} + ae},$$

with a Pareto parameter estimated from the weighted top 1 percent tail of tax-unit adjusted gross income in PolicyEngine's certified microdata (PolicyEngine Team 2024).[1] In the current build, that estimate is $a = 1.621$. For the headline column I assume log utility over consumption, $u(c) = \log c$, and weight top-bracket earners by their average marginal utility normalized to the marginal utility of the earner at the top-bracket threshold. Under a Pareto(a) tail and CRRA(γ) utility that threshold-normalized weight is

[1]The weighted top-1-percent AGI threshold is $725,533 in the current build (with a tail mean of $1,894,129). This is the top-1-percent tax-unit cutoff in the microdata, not the statutory top-bracket edge, which is $640,600 for single filers and $768,700 for married-filing-jointly under TCJA-extended / OBBBA parameters in 2026. Robustness to threshold choice maps onto robustness to the Pareto parameter a; see Appendix Table A13. The exercise binds to no particular statutory bracket — the ETI-based top-rate formula is a common public-finance object that compares across models regardless of where the "top" is drawn. When the microdata calibration is unavailable, the build falls back to $a = 1.5$ and prints a warning; every number in this section comes from the microdata build committed with the paper.

$$\bar{g} = \frac{a}{a+\gamma} = \frac{1.621}{2.621} \approx 0.618,$$

so the headline mapping becomes

$$\tau^* = \frac{0.382}{0.382 + 1.621e}.$$

Two caveats on the welfare weight. First, it is a normalization choice, not an implication of the elicited data: the standard population-normalized utilitarian benchmark drives the top weight toward zero as top incomes grow far beyond the population mean, collapsing the formula to the revenue-maximizing (Diamond-Saez) rate $\tau^* = 1/(1+ae)$ (Diamond 1998; Saez 2001). Table 4 reports that $\bar{g} \to 0$ benchmark alongside the headline column; at the elicited ETI medians it runs from 52.6% to 64.7%, so the threshold normalization reduces the level of the headline rates — though not the cross-model ordering — by roughly 40 percent. Second, calling the headline column "utilitarian" without qualification would overstate it; it is a utilitarian mapping under a specific, disclosed normalization.

The exercise is intentionally stylized. It uses only the ETI — the rest of the labor-supply block stays out — and the microdata enter only through the top-tail calibration, not a full reform simulation. In exchange, it yields a common public-finance object comparable across models. For this policy mapping, I truncate ETI below at zero so the standard sufficient-statistics interpretation remains well defined. Every model's pooled ETI p05 exceeds 0.10 in Table 4, so the truncation affects negligible mass in practice.

Appendix Table A12 shows why this benchmark beats a static flat-tax-plus-demogrant microsimulation for this purpose: absent behavioral responses or leisure in the objective, that exercise describes a distributional frontier, not a meaningful optimal-tax calculation.

The richer point is the propagated uncertainty, beyond the spread in medians. (Table 4 keys off pooled mixture medians, which differ slightly from the mean centers quoted elsewhere in the text — for example, Claude Haiku 4.5's ETI median of 0.502 versus its mean center of 0.507.) Under this threshold-normalized log-utility mapping with a microdata-calibrated Pareto tail, the implied optimal top-rate median ranges from 29.8% for Qwen 3.7 Max to 41.2% for Claude Opus 5, with GPT-5.4 at 35.9%. That is an 11.4 percentage-point cross-model spread in medians. But the within-model uncertainty bands are much wider: top-rate 90 percent intervals are roughly 42 to 65 percentage points wide across models, partly because several models place non-trivial elicited mass on top-earner ETIs above 1, well beyond the empirical literature. The exercise cuts both ways: elicited ETI differences map into meaningfully different central policy conclusions, and any single model's implied optimal-tax recommendation remains very noisy under the paper's own uncertainty object.

Appendix Table A13 reports the same median mapping under alternative values of the Pareto tail parameter a and the CRRA curvature γ. Moving a from the baseline 1.621 to 1.3 shifts

every model's implied top rate up by 7.8 to 8.6 percentage points, and moving a to 1.7 shifts every model down by 1.6 to 1.9 percentage points — the cross-model ordering is invariant under these shifts. Replacing log utility with CRRA at $\gamma = 2$ (holding a at the baseline) lowers the threshold-normalized welfare weight from $\bar{g} = 0.618$ to $\bar{g} = 0.448$ and raises every model's top rate by 8.2 to 9.1 percentage points, again without re-ordering the models.

Table 4. Optimal top-tax exercise from ETI distributions.

Note: Toy public-finance mapping from each model's pooled ETI distribution to an optimal top marginal tax rate under the Saez top-bracket formula tau* = (1 - g_bar) / (1 - g_bar + a e), with a Pareto parameter a = 1.621 estimated from the weighted top 1% tax-unit AGI tail in PolicyEngine's certified microdata (threshold $725,533, tail mean $1,894,129). The welfare weight g_bar = a / (a + gamma) = 0.618 is the average marginal utility of top-bracket earners under CRRA (gamma = 1) utility and a Pareto(a) income tail, normalized to the marginal utility of the earner at the top-bracket threshold. It is a threshold-normalized weight, not the population-normalized utilitarian weight, which would drive g_bar toward zero; the Revenue-max column reports that g_bar -> 0 (Diamond-Saez revenue-maximizing) benchmark tau* = 1 / (1 + a e) at the ETI median. ETI is truncated below at zero for this policy mapping.

Model	ETI median [90%]	Top rate median [90%]	Revenue-max median	Top-rate 90% width (pp)
Claude Opus 5	0.337 [0.086, 0.961]	41.2% [19.7%, 73.2%]	64.7%	53.5
Gemini 3.1 Pro	0.351 [0.103, 0.868]	40.2% [21.3%, 69.5%]	63.8%	48.2
Gemini 3.5 Flash	0.357 [0.102, 0.832]	39.8% [22.0%, 69.8%]	63.4%	47.7
GPT-5.5	0.369 [0.122, 0.994]	39.0% [19.1%, 65.9%]	62.6%	46.8
Kimi K3	0.371 [0.081, 0.992]	38.8% [19.2%, 74.4%]	62.4%	55.2
GPT-5.6 Luna	0.377 [0.068, 1.202]	38.4% [16.4%, 77.5%]	62.0%	61.1
Claude Opus 4.8	0.383 [0.105, 0.993]	38.0% [19.2%, 69.1%]	61.7%	49.9
Gemini 3.1 Flash-Lite	0.389 [0.101, 0.951]	37.7% [19.8%, 69.9%]	61.3%	50.1
Claude Opus 4.7	0.400 [0.123, 0.993]	37.0% [19.2%, 65.7%]	60.7%	46.5
Gemini 3 Flash	0.400 [0.131, 1.040]	37.0% [18.5%, 64.3%]	60.7%	45.8

Model	ETI median [90%]	Top rate median [90%]	Revenue-max median	Top-rate 90% width (pp)
Grok 4.1 Fast	0.400 [0.109, 1.191]	37.0% [16.5%, 68.4%]	60.7%	51.9
Grok 4.5	0.410 [0.105, 1.128]	36.5% [17.3%, 69.3%]	60.1%	52
GPT-5.4	0.420 [0.150, 0.996]	35.9% [19.1%, 61.1%]	59.5%	42
Gemini 3.6 Flash	0.423 [0.112, 0.897]	35.8% [20.8%, 67.7%]	59.3%	46.9
GPT-5.4 mini	0.437 [0.116, 1.377]	35.0% [14.6%, 67.1%]	58.6%	52.5
GPT-5.6 Sol	0.431 [0.150, 1.165]	35.3% [16.8%, 61.1%]	58.9%	44.3
Claude Fable 5	0.437 [0.151, 1.076]	35.0% [17.9%, 60.9%]	58.6%	42.9
Grok 4.3	0.439 [0.133, 1.091]	34.9% [17.8%, 64.0%]	58.4%	46.2
MiniMax M3	0.438 [0.122, 1.330]	34.9% [15.0%, 65.9%]	58.5%	50.9
Claude Sonnet 5	0.471 [0.145, 1.146]	33.3% [17.0%, 61.9%]	56.7%	44.9
GPT-5.6 Terra	0.492 [0.166, 1.182]	32.4% [16.6%, 58.7%]	55.6%	42.1
DeepSeek V4 Pro	0.479 [0.073, 1.808]	32.9% [11.5%, 76.3%]	56.3%	64.7
GLM-5.2	0.495 [0.156, 1.226]	32.2% [16.1%, 60.1%]	55.5%	44
Claude Sonnet 4.6	0.500 [0.138, 1.282]	32.0% [15.5%, 63.1%]	55.2%	47.6
Grok 4.20	0.500 [0.174, 1.243]	32.0% [15.9%, 57.5%]	55.2%	41.6
Kimi K2.6	0.499 [0.146, 1.241]	32.1% [15.9%, 61.7%]	55.3%	45.8
Claude Haiku 4.5	0.502 [0.145, 1.299]	31.9% [15.3%, 61.9%]	55.1%	46.6
GPT-5.4 nano	0.546 [0.151, 1.461]	30.1% [13.9%, 60.9%]	53.1%	47
Qwen 3.7 Max	0.555 [0.161, 1.415]	29.8% [14.3%, 59.4%]	52.6%	45.1

5.4 Convention clarifications in the main prompt

Two quantities with well-documented convention ambiguity have direction-first sign clarifiers embedded in the v4 main prompt: the prime-age income elasticity (whether extra non-labor income raises or reduces hours is a sign question) and the capital-gains realizations elasticity (whether epsilon is taken w.r.t. the tax rate tau or the net-of-tax rate (1 - tau) flips the sign). In the revised wording that 22 of the 29 models received, the clarifier names no conventional direction — it says "$\varepsilon > 0$ if and only if X raises Y; $\varepsilon < 0$ if and only if X reduces Y" for each quantity, which fixes the sign-of-reported-number $\leftrightarrow$ behavioral-direction mapping without asserting which direction is empirically right — and lists the $\varepsilon > 0$ clause first in all cases, which is itself a minor ordering-anchoring choice; a full treatment would randomize clause order across runs. The seven April models elicited before the April 21 revision (see the wording disclosure in the Design section) saw the same two directions as plain conditionals with the conventional direction listed first — an ordering cue aligned with the literature's typical sign rather than uniformly with $\varepsilon > 0$. Under v4, the canonical income elasticity center is negative for 28 of 29 models — the exception is gemini-3.5-flash's bimodal +0.011 mean discussed above, a revised-wording model, so the sign flip is not an artifact of the wording split — and inside the review band for 24 of 29. The other out-of-band centers are negative but sit outside the band. The Armington and IES clarifications continue to produce meaningful cross-prompt deltas (Appendix Tables A7, A8). The capital-gains convention audit (Appendix Table A9) elicits both parameterizations of the same economic object to test whether each model answers consistently across conventions.

A second prompt-sensitivity check on the macro-and-trade side confirms the same methodological point: a 29-model follow-up on the Armington elasticity with an explicit top-level-import-versus-domestic clarification shifts pooled centers downward for most models (Appendix Table A7). Some cross-model disagreement reflects object-definition differences the base prompt underspecifies, not substantive belief disagreement.

5.5 What correlates with a model's answers?

Two exploratory cross-model cuts close the results. Both are descriptive associations over a convenience census of models — every model differs from every other on architecture, training data, organization, and serving path at once — so I report rank statistics with permutation p-values, adjust for the full test family, and pre-commit to the framing that nothing here is causal.

The first cut joins the panel to PolicyBench, PolicyEngine's policy-calculation benchmark, on its headline metric: the household-weighted share of predictions within one dollar of the reference answer (US, no-tools condition, pinned release dashboard-data-20260721; 25 of the 29 panel models appear on the leaderboard — all but gpt-5.4, grok-4.20, grok-4.1-fast, and claude-opus-5, the last elicited before any PolicyBench release scores it). Table 5 reports the per-model join and Table 6 the correlations for two predictors — the overall within-$1 rate and a domain-matched

tax-only version built from the benchmark's seven tax variables. Models that score higher on policy calculation elicit lower taxable-income elasticities: Spearman $\rho = -0.53$ and -0.51 under the two predictors (raw p of 0.006 and 0.010, $n = 25$). Through the Saez formula that is the same statement as "more capable models imply higher optimal top rates," so Table 6 reports the top-rate row as a derived transform, not an additional test. Across the eight-test family, the smallest Holm-adjusted p-value is 0.052 (Benjamini-Hochberg 0.042): the association clears the Benjamini-Hochberg false-discovery threshold at the 0.05 level and narrowly misses familywise Holm significance, so it reads as the panel's strongest suggestive result rather than an established effect. It is, however, stable in sign under leave-one-organization-out deletion (ρ between -0.45 and -0.61 across all nine omissions) and under an organization-block permutation that moves score vectors as whole-lab units. Interval tightness shows no comparable association with capability.

A post-hoc cut, prompted in review of that width result and computed after the eight-test family was declared, separates the two components the pooled interval mixes: the width a model states within a run and how much its answers move across the 15 runs — the Appendix Table A15 decomposition. Capability tracks the second component. The Spearman correlation between the within-$1 rate and a model's worst-case between-run variance share is -0.43 (raw permutation $p = 0.031$, $n = 25$), and under the domain-matched tax predictor it is -0.65 ($p = 0.001$) — numerically the strongest association in this paper, reported outside the adjusted family by construction and read accordingly (results/correlates-posthoc.csv carries the note). The pattern is visible without statistics: every Claude model's worst-case share stays below 4%, while the five largest shares — 37%, 29%, 25%, 18%, and 13% — all belong to models at or below the leaderboard median. Higher-scoring models do not state narrower intervals; they restate the same distribution more repeatably, and the run-instability tail concentrates in lower-scoring models.

Table 5. Panel joined to the PolicyBench leaderboard.

Note: Per-model summary joining the elicitation panel to the published PolicyBench leaderboard (PolicyEngine/policybench dashboard-data-20260721, US, no-tools, household-weighted within-$1 rate — the leaderboard headline). Width rank averages tie-averaged 90 percent interval-width ranks across the canonical panel (1 = tightest). Implied top rates come from Table 4's threshold-normalized log-utility mapping. Dashes mark models outside the PolicyBench panel.

Model	Organization	Wave	PolicyBench within-$1	ETI median	Implied top rate	Avg width rank
Claude Opus 5	Anthropic	July 2026 late	—	0.337	41.2%	12.3
Gemini 3.1 Pro	Google	April 2026	77.9	0.351	40.2%	10.3

Model	Organization	Wave	Policy-Bench within-$1	ETI median	Implied top rate	Avg width rank
Gemini 3.5 Flash	Google	July 2026 frontier	76.2	0.357	39.8%	12.5
GPT-5.5	OpenAI	July 2026 frontier	83.5	0.369	39.0%	14.2
Kimi K3	Moonshot AI	July 2026 late	86.2	0.371	38.8%	17.8
GPT-5.6 Luna	OpenAI	July 2026 GPT-5.6	84.5	0.377	38.4%	24.5
Claude Opus 4.8	Anthropic	July 2026 frontier	72.6	0.383	38.0%	10.4
Gemini 3.1 Flash-Lite	Google	April 2026	76.1	0.389	37.7%	13.1
Claude Opus 4.7	Anthropic	April 2026	77.4	0.400	37.0%	8.8
Gemini 3 Flash	Google	April 2026	76.9	0.400	37.0%	12.2
Grok 4.1 Fast	xAI	April 2026	—	0.400	37.0%	18.0
Grok 4.5	xAI	July 2026 late	80.9	0.410	36.5%	17.5
GPT-5.4	OpenAI	April 2026	—	0.420	35.9%	14.2
Gemini 3.6 Flash	Google	July 2026 late	79.0	0.423	35.8%	6.8
GPT-5.6 Sol	OpenAI	July 2026 GPT-5.6	88.7	0.431	35.3%	16.2
GPT-5.4 mini	OpenAI	April 2026	70.5	0.437	35.0%	21.8
Claude Fable 5	Anthropic	July 2026 frontier	79.9	0.437	35.0%	6.5
Grok 4.3	xAI	July 2026 frontier	77.2	0.439	34.9%	17.5
MiniMax M3	MiniMax	July 2026 independent labs	72.4	0.438	34.9%	17.6
Claude Sonnet 5	Anthropic	July 2026 frontier	69.4	0.471	33.3%	9.2
DeepSeek V4 Pro	DeepSeek	July 2026 independent labs	76.1	0.479	32.9%	18.8

Model	Organization	Wave	PolicyBench within-$1	ETI median	Implied top rate	Avg width rank
GPT-5.6 Terra	OpenAI	July 2026 GPT-5.6	83.4	0.492	32.4%	13.1
GLM-5.2	Zhipu AI	July 2026 independent labs	73.1	0.495	32.2%	19.5
Kimi K2.6	Moonshot AI	July 2026 independent labs	64.6	0.499	32.1%	21.3
Claude Sonnet 4.6	Anthropic	April 2026	77.1	0.500	32.0%	12.8
Grok 4.20	xAI	April 2026	—	0.500	32.0%	22.3
Claude Haiku 4.5	Anthropic	April 2026	71.7	0.502	31.9%	8.8
GPT-5.4 nano	OpenAI	April 2026	62.3	0.546	30.1%	17.7
Qwen 3.7 Max	Alibaba	July 2026 independent labs	73.6	0.555	29.8%	19.2

Table 6. Capability correlations with elicited outcomes.

Note: Spearman rank correlations between the PolicyBench within-$1 rate (PolicyEngine/policybench dashboard-data-20260721, US, no-tools) and per-model elicitation outcomes, with raw two-sided permutation p-values (20,000 draws, fixed seed; exact when n <= 8) plus Holm and Benjamini-Hochberg adjustments across the eight non-derived tests spanning both predictors. The top-rate row is the same ETI hypothesis under a monotone transformation, not an additional test. Descriptive: models differ across every axis at once, so these are cross-family associations, not causal effects.

Predictor	Outcome	Models	Spearman rho	Raw permutation p	Holm-adjusted p	BH-adjusted p	Family size	Derived
Tax within-$1 (domain-matched)	Mean	center	, labor-and-tax	25	0.054	0.795	1.000	0.932

Predictor	Outcome	Models	Spearman rho	Raw permutation p	Holm-adjusted p	BH-adjusted p	Family size	Derived
Tax within-$1 (domain-matched)	Mean	center	, macro-and-trade	25	0.169	0.420	1.000	0.833
Tax within-$1 (domain-matched)	Avg interval-width rank (1 = tightest)	25	-0.240	0.242	1.000	0.645	8	no
Tax within-$1 (domain-matched)	ETI pooled median	25	-0.513	0.010	0.073	0.042	8	no
Tax within-$1 (domain-matched)	Implied optimal top rate (%) — derived, monotone transform of ETI — not an additional test	25	0.513	0.010	0.073	0.042	8	yes
Overall within-$1 (leaderboard headline)	Mean	center	, labor-and-tax	25	-0.011	0.959	1.000	0.959

Predictor	Outcome	Models	Spearman rho	Raw permutation p	Holm-adjusted p	BH-adjusted p	Family size	Derived
Overall within-$1 (leaderboard headline)	Mean	center	, macro-and-trade	25	0.050	0.816	1.000	0.932
Overall within-$1 (leaderboard headline)	Avg interval-width rank (1 = tightest)	25	-0.134	0.521	1.000	0.833	8	no
Overall within-$1 (leaderboard headline)	ETI pooled median	25	-0.532	0.006	0.052	0.042	8	no
Overall within-$1 (leaderboard headline)	Implied optimal top rate (%) — derived, monotone transform of ETI — not an additional test	25	0.532	0.006	0.052	0.042	8	yes

The second cut groups the panel by lab home country: twenty-three models from four US organizations against six from Chinese labs. The Chinese-lab models imply lower optimal

top rates (median 32.6% versus 35.9%, an exact group-label permutation $p = 0.062$ on the difference in medians), through higher elicited taxable-income elasticities (median ETI 0.487 versus 0.420, $p = 0.048$), and they state wider intervals (average width rank 19.0 versus 13.1, $p = 0.068$). Four caveats bound this comparison. Lab country is perfectly confounded with the serving path — every Chinese-lab model ran through OpenRouter's JSON mode — and with the elicitation wave; the six models also ran with uniformly high completion budgets and provider-default reasoning (8,000-16,000 tokens, a configuration only nine US models share), which bears most directly on the width contrast; the prompts are English-language, so the comparison measures these labs' models as English-prompted policy assistants, not Chinese-language deployments; and with groups of 6 and 23, the exact permutation distribution over medians remains coarse, so these results are directional — and they weakened as the late additions (kimi-k3, gemini-3.6-flash, then claude-opus-5) joined the panel. Table 7 applies the same multiplicity standard as the capability cut — Holm and Benjamini-Hochberg over the cut's four-outcome family, with the top-rate row again a derived transform of the ETI row: the smallest Holm-adjusted p-value is 0.192, so nothing in this cut survives correction.

Table 7. US-lab versus Chinese-lab medians.

Note: US-lab versus Chinese-lab medians with exact group-label permutation p-values on the difference in medians, and Holm and Benjamini-Hochberg adjustment over the four-outcome family (* = the top-rate row is a derived transform of the ETI row and mirrors its adjusted values rather than entering the family). Exploratory: lab country is perfectly confounded with the serving path (every Chinese-lab model ran through OpenRouter JSON mode) and with elicitation wave, and co-varies with completion budget and reasoning configuration; prompts are English-language, and the group sizes (5 versus 20) put a floor near 0.02-0.05 on attainable p-values.

Outcome	US median (n)	China median (n)	China - US	Permutation p	Holm p	BH p
Implied optimal top rate (%)	35.917 (23)	32.589 (6)	-3.329	0.062	0.192*	0.136*
ETI pooled median	0.420 (23)	0.487 (6)	+0.067	0.048	0.192	0.136
Avg interval-width rank (1 = tightest)	13.077 (23)	19.000 (6)	+5.923	0.068	0.204	0.136
Mean	center	, labor-and-	0.369 (23)	0.340 (6)	-0.028	0.103

Outcome	US median (n)	China median (n)	China - US	Permutation p	Holm p	BH p
Mean	center	, macro-and-trade	1.164 (23)	1.070 (6)	-0.094	0.282

6 Interpretation

The protocol identifies prompt-conditioned response distributions under a fixed elicitation design — it adjudicates no model as correct and recovers no latent stable priors. That object is narrower than "belief" in a deep psychological sense, and it is exactly the object users meet when they interact with these systems through prompts.

That matters for three reasons.

First, the domain-specific ordering means no model carries one general "elasticity ideology." A model can look relatively high-response on labor-and-tax quantities and relatively low-response on macro-and-trade quantities, or vice versa.

Second, pooled predictive uncertainty is itself informative. Some models behave as if the relevant literature is tight and settled; others behave as if the same quantities are wide-open or as if their own answers are unstable across repeated runs. Users interacting with these models will experience that difference directly. One caveat applies to the three no-sampling-parameter Claude models (Fable 5, Opus 4.8, Sonnet 5), whose between-run variation arises under provider-default sampling instead of the panel's temperature = 1.0; Appendix Table A15 shows this matters little in practice, because the between-run component contributes a median of 0 to 2 percent of pooled predictive variance for every model — the models' own stated quantiles dominate the widths, and the sampling regime does not touch those.

Third, prompt sensitivity is part of the estimand, not a nuisance to hide. The income-elasticity sign clarification shows that a fixed protocol can still identify a mixture of economic judgment and convention handling — which calls for describing the object precisely and investing in quantity-specific prompt design.

This also clarifies the relationship between the present paper and the LLM uncertainty literature discussed above. Papers such as Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs (2023), Generating with Confidence (2023), and LLMs Are Overconfident (2025) mostly evaluate whether uncertainty reports line up with eventual correctness. By contrast, this paper asks how models respond when the target is a contested economic parameter and the disagreement itself is economically interesting. In that sense, the project is closer to synthetic expert elicitation than to benchmark evaluation.

7 Limitations

This is an initial results paper, not a final measurement of latent model beliefs.

The current limitations are straightforward:

- the protocol identifies prompt-conditioned responses, not stable latent priors
- the main panel is memory-only in design and almost entirely tool-free in realized behavior; retrieval-augmented and tool-using arms are still secondary
- the current full dataset still contains simulation-facing response coefficients and one labeled legacy row, even though the headline tables now separate them
- the current paper focuses on elasticities, not the broader OG-USA-style parameter set
- the interval analysis is descriptive; calibration against resolved numeric tasks is still modular but secondary
- citation strings are noisy and need normalization before they can support serious bibliometric analysis

8 Appendix

8.1 Stability check

Appendix Table A1 reports a simple stability check for the choice of $R = 15$. Across the canonical 13-quantity subpanel (377 cells at every prefix length), the median absolute difference between the first 10 runs in a cell and the full 15-run pooled center is only 0.002; the corresponding median difference in pooled 90 percent width is 0.010. Using only the first 5 runs is noisier, but still modest at the median.

Appendix Table A1. Prefix stability for the canonical subpanel.

Note: Prefix stability on the 13-quantity canonical subpanel. Rows compare pooled summaries from the first 5 or 10 runs in a cell to the full 15-run pooled summary.

Runs used	Cells compared	Median abs change in pooled center	90th pct abs change in pooled center	Median abs change in pooled width	90th pct abs change in pooled width
5	377	0.006	0.047	0.027	0.244
10	377	0.002	0.027	0.01	0.134
15	377	0	0	0	0

8.2 Pooling robustness

Appendix Table A2 compares model-level predictive-uncertainty rankings under three interval constructions: the headline pooled mixture interval, a REML random-effects predictive interval, and a Bayesian hierarchical predictive interval. The rankings move somewhat, but the changes are not large enough to overturn the broad qualitative picture. Claude Fable 5, Claude Haiku 4.5, and Claude Sonnet 5 remain among the tighter models across all three constructions, joined by the new Gemini 3.6 Flash, while GPT-5.4 mini, Grok 4.20, and GPT-5.6 Luna remain toward the wide end. The largest rank spread is 4.38 positions, for MiniMax M3.

Appendix Table A2. Predictive-uncertainty ranking under alternative pooling methods.

Note: Canonical 13-quantity subpanel. Average predictive-uncertainty ranks are computed under the headline pooled mixture interval, the REML predictive interval, and the Bayesian predictive interval.

Model	Avg pooled rank	Avg REML rank	Avg Bayes rank	Max rank spread
Claude Fable 5	6.46	7.12	6.54	0.65
Gemini 3.6 Flash	6.85	9.46	8.35	2.62
Claude Haiku 4.5	8.85	7.85	7.69	1.15
Claude Opus 4.7	8.85	12.31	11.08	3.46
Claude Sonnet 5	9.15	11.31	10.69	2.15
Gemini 3.1 Pro	10.27	9.15	9.46	1.12
Claude Opus 4.8	10.42	13.65	13	3.23
Gemini 3 Flash	12.15	12.15	11.38	0.77
Claude Opus 5	12.31	13.08	12.15	0.92
Gemini 3.5 Flash	12.46	14.62	16	3.54
Claude Sonnet 4.6	12.85	17	16	4.15
GPT-5.6 Terra	13.08	13.38	14.15	1.08
Gemini 3.1 Flash-Lite	13.08	13.08	12.85	0.23
GPT-5.5	14.15	16.62	15.69	2.46
GPT-5.4	14.23	17.08	16.23	2.85
GPT-5.6 Sol	16.23	18.15	17.69	1.92
Grok 4.3	17.54	17.12	16.62	0.92
Grok 4.5	17.62	18.15	16.81	1.35
MiniMax M3	17.62	13.23	16.15	4.38
GPT-5.4 nano	17.69	14.92	16.92	2.77
Kimi K3	17.77	17	17.46	0.77
Grok 4.1 Fast	17.96	16.62	16.19	1.77

Model	Avg pooled rank	Avg REML rank	Avg Bayes rank	Max rank spread
DeepSeek V4 Pro	18.85	15.08	17.15	3.77
Qwen 3.7 Max	19.15	17.19	17.65	1.96
GLM-5.2	19.54	15.77	17.27	3.77
Kimi K2.6	21.31	18.08	18.85	3.23
GPT-5.4 mini	21.81	20	20.04	1.81
Grok 4.20	22.31	21.62	20.85	1.46
GPT-5.6 Luna	24.46	24.23	24.08	0.38

8.3 Leave-one-organization-out stability

Appendix Table A3 asks whether any single organization drives the subpanel rankings, deleting each of the nine organizations in turn. None does. Across both the labor-and-tax and macro-and-trade subpanels, the leave-one-organization-out Spearman correlation with the full-panel ranking stays between 0.994 and 1.0. Some top spots do flip when one organization drops out, especially on the smaller three-quantity macro-and-trade panel, but the broad ordering holds.

Appendix Table A3. Leave-one-organization-out ranking sensitivity.

Note: Leave-one-organization-out sensitivity of the average absolute-elasticity ranking on the labor-and-tax and macro-and-trade canonical subpanels.

Subpanel	Omitted organization	Spearman rho	Top retained model, full panel	Top retained model, leave-out	Max avg-rank shift
Labor/tax	Alibaba	0.999	Claude Sonnet 4.6	Claude Sonnet 4.6	1
Labor/tax	Anthropic	0.994	Grok 4.20	Grok 4.5	6.167
Labor/tax	DeepSeek	0.999	Claude Sonnet 4.6	Claude Sonnet 4.6	0.833
Labor/tax	Google	0.996	Claude Sonnet 4.6	Claude Sonnet 4.6	3.167
Labor/tax	MiniMax	0.999	Claude Sonnet 4.6	Claude Sonnet 4.6	0.833
Labor/tax	Moonshot AI	0.999	Claude Sonnet 4.6	Claude Sonnet 4.6	2
Labor/tax	OpenAI	0.997	Claude Sonnet 4.6	Claude Sonnet 4.6	5.667
Labor/tax	Zhipu AI	1	Claude Sonnet 4.6	Claude Sonnet 4.6	0.833

Subpanel	Omitted organization	Spearman rho	Top retained model, full panel	Top retained model, leave-out	Max avg-rank shift
Labor/tax	xAI	0.998	Claude Sonnet 4.6	Claude Sonnet 4.6	3.5
Macro/trade	Alibaba	1	Grok 4.3	Grok 4.3	0.333
Macro/trade	Anthropic	0.998	Grok 4.3	Grok 4.3	5
Macro/trade	DeepSeek	1	Grok 4.3	Grok 4.3	1
Macro/trade	Google	0.998	Grok 4.3	Grok 4.3	4.667
Macro/trade	MiniMax	1	Grok 4.3	Grok 4.3	0.667
Macro/trade	Moonshot AI	1	Grok 4.3	Grok 4.3	1.667
Macro/trade	OpenAI	0.994	Grok 4.3	Grok 4.20	7
Macro/trade	Zhipu AI	1	Grok 4.3	Grok 4.3	1
Macro/trade	xAI	0.999	GPT-5.6 Luna	GPT-5.4 nano	3.667

8.4 Alternative quantile-to-distribution rule

Appendix Table A4 replaces the headline piecewise-uniform reconstruction with a cruder transformed-normal approximation calibrated to p05, p50, and p95 for each run. The resulting predictive-uncertainty ranks move more at 29 models than they did at 17: the largest average rank change is 4.77 positions for MiniMax M3, and only 8 of 29 models move by less than one rank position. The ordering is therefore rule-sensitive in the middle of the pack; the paper's conclusions rest on the extremes and on coarse groupings, which hold under both rules, not on fine mid-pack placements.

Appendix Table A4. Sensitivity to the within-run distribution rule.

Note: Canonical 13-quantity subpanel. The alternative rule approximates each run as a transformed normal calibrated to p05, p50, and p95 instead of the headline piecewise-uniform quantile-bin reconstruction.

Model	Avg piecewise-uniform rank	Avg transformed-normal rank	Rank shift
Claude Fable 5	6.46	7.62	1.15
Gemini 3.6 Flash	6.85	8.54	1.69
Claude Opus 4.7	8.85	12.54	3.69
Claude Haiku 4.5	8.85	7.69	-1.15
Claude Sonnet 5	9.15	11.23	2.08
Gemini 3.1 Pro	10.27	9.15	-1.12
Claude Opus 4.8	10.42	13.65	3.23

Model	Avg piecewise-uniform rank	Avg transformed-normal rank	Rank shift
Gemini 3 Flash	12.15	11.92	-0.23
Claude Opus 5	12.31	13.31	1
Gemini 3.5 Flash	12.46	13.69	1.23
Claude Sonnet 4.6	12.85	16.96	4.12
GPT-5.6 Terra	13.08	12.92	-0.15
Gemini 3.1 Flash-Lite	13.08	12.69	-0.38
GPT-5.5	14.15	16.23	2.08
GPT-5.4	14.23	15.62	1.38
GPT-5.6 Sol	16.23	18.31	2.08
Grok 4.3	17.54	17	-0.54
MiniMax M3	17.62	12.85	-4.77
Grok 4.5	17.62	17.85	0.23
GPT-5.4 nano	17.69	16.54	-1.15
Kimi K3	17.77	17.23	-0.54
Grok 4.1 Fast	17.96	16	-1.96
DeepSeek V4 Pro	18.85	15.46	-3.38
Qwen 3.7 Max	19.15	16.77	-2.38
GLM-5.2	19.54	17.92	-1.62
Kimi K2.6	21.31	18.08	-3.23
GPT-5.4 mini	21.81	20.69	-1.12
Grok 4.20	22.31	21.62	-0.69
GPT-5.6 Luna	24.46	24.92	0.46

8.5 Tool-access robustness

The paper's main estimand is a memory-based elicitation design. To check whether that framing is misleading in practice, I also ran an earlier GPT-only robustness arm that gave the models web search and code interpreter access. That arm covered an earlier eight-quantity subset, so it does not compare directly to the current 26-quantity main panel. Its value is narrower: it shows whether tool access materially changed realized behavior.

In that archived tool-access arm, actual tool uptake was rare. Across all 360 requests, the models used web search in only 9 cases (2.5%) and code interpreter never. Full GPT-5.4 never used a tool; GPT-5.4 mini used web search in 2 / 120 requests; and GPT-5.4 nano used web search in 7 / 120 requests. The appendix table below summarizes those counts.

Appendix Table A5. Tool use in the archived GPT-only tool-access arm.

Note: Archived GPT-only robustness arm with full web and code-interpreter access on the earlier 8-quantity panel (8 quantities x 15 runs x 3 GPT models = 360 requests). In realized behavior, tool uptake was rare and code interpreter was never used.

Model	Requests	Requests with web use	Share with web use	Total web calls	Requests with code use	Share with code use	Total code calls
GPT-5.4	120	0	0.0%	0	0	0.0%	0
GPT-5.4 mini	120	2	1.7%	2	0	0.0%	0
GPT-5.4 nano	120	7	5.8%	7	0	0.0%	0
All GPT models	360	9	2.5%	9	0	0.0%	0

8.6 Canonical quantity disagreement

Appendix Table A6 reports quantity-level disagreement on the canonical 13-quantity panel. The final column scales cross-model spread by the average pooled 90 percent width. On this metric, no canonical quantity has cross-model spread larger than average within-model predictive width.

Appendix Table A6. Canonical quantity disagreement.

Note: Canonical elasticity subpanel only, sorted by cross-model spread in pooled point estimates. Simulation-facing PolicyEngine coefficients are broken out in separate appendix-style tables.

Quantity	Lowest model	Lowest center	Highest model	Highest center	Spread	Mean pooled 90% width	Spread / mean width
Armington elasticity	Qwen 3.7 Max	1.433	Claude Sonnet 5	3.18	1.747	5.432	0.322
Intertemporal elasticity of substitution	MiniMax M3	0.483	Gemini 3.1 Pro	1.467	0.983	1.788	0.55

Quantity	Lowest model	Lowest center	Highest model	Highest center	Spread	Mean pooled 90% width	Spread / mean width
Capital gains realizations elasticity	GPT-5.4 mini	-0.93	Gemini 3.5 Flash	0.01	0.94	1.884	0.499
Coefficient of relative risk aversion	Claude Haiku 4.5	1.567	GLM-5.2	2.1	0.533	8.151	0.065
Employment participation elasticity of single mothers	GPT-5.4 nano	0.213	Qwen 3.7 Max	0.717	0.503	1.264	0.398
Frisch elasticity of labor supply	Claude Haiku 4.5	0.283	GPT-5.4 nano	0.593	0.31	1.298	0.239
Elasticity of substitution between capital and labor	Qwen 3.7 Max	0.593	GPT-5.4 mini	0.883	0.29	1.1	0.264
Elasticity of taxable income	Claude Opus 5	0.335	Qwen 3.7 Max	0.554	0.219	1.043	0.21
TFP persistence	GPT-5.4 nano	0.823	Gemini 3.6 Flash	0.956	0.134	0.215	0.621
Uncompensated wage elasticity of labor supply	Gemini 3.1 Pro	0.04	Grok 4.5	0.168	0.128	0.602	0.213

Quantity	Lowest model	Lowest center	Highest model	Highest center	Spread	Mean pooled 90% width	Spread / mean width
Income elasticity of labor supply	Grok 4.20	-0.107	Gemini 3.5 Flash	0.011	0.118	0.379	0.311
Capital share in production	Claude Haiku 4.5	0.307	GPT-5.6 Luna	0.355	0.048	0.198	0.243
Annual discount factor	Grok 4.20	0.959	GPT-5.6 Terra	0.982	0.023	0.087	0.261

8.7 Armington clarification follow-up

Appendix Table A7 reports a second prompt-sensitivity probe. For the Armington elasticity, the follow-up prompt made explicit that the target was the top-level import-versus-domestic substitution elasticity, not source-country substitution or a sector-level import-demand elasticity. Across all 29 model reruns, that clarification shifts pooled centers downward or leaves them essentially unchanged; the only increases are small (+0.013 for Claude Fable 5, +0.033 for Qwen 3.7 Max, +0.167 for MiniMax M3). The largest decreases appear for Gemini 3.1 Flash-Lite (-0.767), Claude Sonnet 5 (-0.647), and Claude Sonnet 4.6 and GPT-5.4 (-0.433 each). Two clarified intervals (Gemini 3 Flash, Gemini 3.1 Pro) reach exactly 0 at their lower end, which reflects the registry support floor, not an elicited quantile. I interpret the deltas as further evidence that even apparently standard macro parameters can hide meaningful object-definition ambiguity.

Appendix Table A7. Armington clarification follow-up.

Note: Change in the Armington elasticity after clarifying that the target is the top-level import-versus-domestic elasticity, not source-country or sector-level substitution.

Model	Old center	Clarified center	Change	Old pooled 90% interval	Clarified pooled 90% interval
Claude Fable 5	1.987	2	0.013	[0.9024, 4.738]	[0.9508, 3.995]
Claude Haiku 4.5	1.5	1.387	-0.113	[0.6154, 2.979]	[0.4396, 2.787]

Model	Old center	Clarified center	Change	Old pooled 90% interval	Clarified pooled 90% interval
Claude Opus 4.7	1.533	1.5	-0.033	[0.6788, 4.574]	[0.6687, 3.983]
Claude Opus 4.8	2.4	2.033	-0.367	[0.9516, 6.965]	[0.9494, 5.661]
Claude Opus 5	2.4	2.36	-0.04	[0.895, 6.477]	[0.9501, 6.012]
Claude Sonnet 4.6	2.267	1.833	-0.433	[0.8333, 6.615]	[0.7826, 4.295]
Claude Sonnet 5	3.18	2.533	-0.647	[1.161, 7.883]	[1.002, 5.995]
DeepSeek V4 Pro	1.853	1.513	-0.34	[0.5548, 5.321]	[0.3605, 4.728]
GLM-5.2	2.073	1.727	-0.347	[0.6104, 6.707]	[0.5482, 4.848]
GPT-5.4	2.5	2.067	-0.433	[1.08, 5.997]	[0.9368, 5]
GPT-5.4 mini	2.3	2.1	-0.2	[0.8105, 8.107]	[0.8221, 6.607]
GPT-5.4 nano	2.233	1.993	-0.24	[0.836, 5.921]	[0.8427, 4.391]
GPT-5.5	2.153	1.94	-0.213	[0.7121, 6.78]	[0.6748, 6.087]
GPT-5.6 Luna	2.467	2.1	-0.367	[0.5303, 9.158]	[0.5389, 7.603]
GPT-5.6 Sol	2.067	1.7	-0.367	[0.6556, 6.546]	[0.6247, 4.906]
GPT-5.6 Terra	2	1.9	-0.1	[0.8897, 5]	[0.8308, 4.965]
Gemini 3 Flash	1.633	1.573	-0.061	[0.6512, 5.783]	[0, 5.903]
Gemini 3.1 Flash-Lite	2.367	1.6	-0.767	[0.6797, 7.27]	[0.5283, 4.856]
Gemini 3.1 Pro	1.653	1.444	-0.209	[0.5471, 4.933]	[0, 5.592]
Gemini 3.5 Flash	1.753	1.387	-0.367	[0.6111, 5.062]	[0.5137, 3.815]
Gemini 3.6 Flash	1.96	1.54	-0.42	[0.7596, 5.385]	[0.6147, 3.944]
Grok 4.1 Fast	1.5	1.5	0	[0.6512, 3]	[0.7059, 3]

Model	Old center	Clarified center	Change	Old pooled 90% interval	Clarified pooled 90% interval
Grok 4.20	2	1.967	-0.033	[0.5034, 7.71]	[0.5084, 7.252]
Grok 4.3	2.94	2.7	-0.24	[0.8114, 9.02]	[0.7451, 8.039]
Grok 4.5	2.467	2.433	-0.033	[0.7283, 7.604]	[0.9137, 6.944]
Kimi K2.6	1.833	1.587	-0.247	[0.5653, 5.968]	[0.5596, 4.487]
Kimi K3	2.5	2.5	0	[0.5821, 7.645]	[0.6115, 7.075]
MiniMax M3	1.987	2.153	0.167	[0.8013, 5.08]	[0.8077, 5.725]
Qwen 3.7 Max	1.433	1.467	0.033	[0.496, 4.412]	[0.5137, 3.905]

8.8 IES clarification follow-up

Appendix Table A8 reports a third prompt-sensitivity probe on the intertemporal elasticity of substitution. The follow-up prompt makes explicit that the target is the annual macro-calibration IES for nondurable consumption, not a generic inverse-CRRA or asset-pricing object. Under the clarification, most pooled centers stay near 0.5 or move slightly downward; the largest shift is -0.842 for Gemini 3.1 Pro (from 1.467 to 0.625), which suggests the base prompt's IES interpretation was picking up more of the inverse-CRRA reading than the macro-calibration reading for that model. This is a narrower ambiguity than the Armington and income-sign cases, but it is conceptually important because the IES is often conflated with risk aversion in casual model discussions.

Appendix Table A8. IES clarification follow-up.

Note: Full-panel follow-up across all 11 models. Change in the intertemporal elasticity of substitution after clarifying that the target is the annual macro-calibration IES for nondurable consumption, not a generic inverse-CRRA or asset-pricing object.

Model	Old center	Clarified center	Change	Old pooled 90% interval	Clarified pooled 90% interval
Claude Fable 5	0.5	0.407	-0.093	[0.1, 1.661]	[0.0692, 1.346]

Model	Old center	Clarified center	Change	Old pooled 90% interval	Clarified pooled 90% interval
Claude Haiku 4.5	0.5	0.5	0	[0.2058, 1.188]	[0.2115, 1.193]
Claude Opus 4.7	0.5	0.5	0	[0.092, 1.965]	[0.1, 1.483]
Claude Opus 4.8	0.5	0.5	0	[0.1, 1.5]	[0.1, 1.455]
Claude Opus 5	0.7	0.52	-0.18	[0.15, 1.994]	[0.1102, 1.533]
Claude Sonnet 4.6	0.5	0.5	0	[0.1, 1.5]	[0.1, 1.491]
Claude Sonnet 5	0.5	0.5	0	[0.1024, 1.482]	[0.1024, 1.199]
DeepSeek V4 Pro	0.527	0.5	-0.027	[0.06604, 2.054]	[0.07207, 1.918]
GLM-5.2	0.533	0.493	-0.04	[0.1018, 1.809]	[0.1126, 1.466]
GPT-5.4	0.5	0.5	0	[0.1133, 1.388]	[0.1862, 1.2]
GPT-5.4 mini	0.54	0.5	-0.04	[0.1014, 1.924]	[0.092, 1.498]
GPT-5.4 nano	0.747	0.733	-0.013	[0.2167, 1.956]	[0.2118, 1.923]
GPT-5.5	0.5	0.41	-0.09	[0.06411, 1.664]	[0.04859, 1.423]
GPT-5.6 Luna	0.66	0.683	0.023	[0.1598, 2.248]	[0.16, 2.03]
GPT-5.6 Sol	0.56	0.5	-0.06	[0.1091, 1.727]	[0.1012, 1.466]
GPT-5.6 Terra	0.5	0.5	0	[0.2, 1.191]	[0.1613, 1.157]
Gemini 3 Flash	0.5	0.478	-0.022	[0.1, 1.979]	[0, 1.667]
Gemini 3.1 Flash-Lite	0.5	0.5	0	[0.1, 1.431]	[0.1018, 1.198]
Gemini 3.1 Pro	1.467	0.625	-0.842	[0.2857, 2.499]	[0, 1.767]
Gemini 3.5 Flash	0.933	0.567	-0.367	[0.07895, 2.11]	[0.1037, 1.899]

Model	Old center	Clarified center	Change	Old pooled 90% interval	Clarified pooled 90% interval
Gemini 3.6 Flash	0.59	0.483	-0.107	[0.1026, 1.916]	[0.092, 1.197]
Grok 4.1 Fast	0.5	0.5	0	[0.1474, 1.97]	[0.1429, 2]
Grok 4.20	0.967	0.667	-0.3	[0.2004, 3.611]	[0.1211, 3.224]
Grok 4.3	0.873	0.863	-0.01	[0.2, 2.457]	[0.2164, 2.421]
Grok 4.5	0.933	0.833	-0.1	[0.1619, 2.494]	[0.1357, 2.462]
Kimi K2.6	0.73	0.563	-0.167	[0.1122, 2.678]	[0.1131, 1.668]
Kimi K3	0.487	0.45	-0.037	[0.068, 1.71]	[0.06207, 1.758]
MiniMax M3	0.483	0.51	0.027	[0.1, 1.47]	[0.1099, 1.471]
Qwen 3.7 Max	0.627	0.492	-0.135	[0.1125, 2.018]	[0.09787, 1.492]

8.9 Capital-gains convention audit

Appendix Table A9 elicits both parameterizations of the capital-gains realizations elasticity under the v4 main prompt: the headline panel's w.r.t.-tax-rate convention and a sibling quantity defined w.r.t. the net-of-tax rate. The identity $\varepsilon_\tau = -\frac{\tau}{1-\tau}\,\varepsilon_{1-\tau}$ inverts to $\tau = -\rho/(1-\rho)$ where $\rho = \varepsilon_\tau/\varepsilon_{1-\tau}$. I bootstrap the implied-τ distribution — drawing 1,000 independent pairs $(\varepsilon_{\tau,i}, \varepsilon_{1-\tau,j})$ from each model's 15 tax-rate runs and 15 net-of-tax-rate runs and inverting per pair — instead of plugging the two pooled centers into the inversion once. The table reports the median and 90 percent interval of that distribution. The bootstrap median need not equal the plug-in ratio of medians: $\tau = -\rho/(1-\rho)$ is nonlinear in ρ, so Jensen's inequality lets averaging implied-τ values across pairs move the center relative to the implied τ at the two pooled medians, and the bootstrap quantiles also make the residual uncertainty visible.[2]

The table uses two band anchors. The narrow [0.15, 0.37] window covers the U.S. top-bracket long-term-capital-gains envelope (federal LTCG top of 0.20, plus a 0.038 NIIT, plus the highest

[2] A shared-tau coherence violation does not strictly demand a convention swap. A model could legitimately hold two defensible but internally independent magnitudes drawn from different literatures — for example, a net-of-tax elasticity anchored in a medium-run realizations study and a tax-rate elasticity anchored in a short-run timing study — in which case the bootstrap would still reveal joint incoherence even though neither individual answer is wrong. The audit therefore identifies joint incoherence under a single shared τ without labeling the underlying cause.

state LTCG layer); the wider (0.37, 0.55] window covers an ordinary-income-rate anchor (federal ordinary-income top plus high state). The table flags an implied median τ inside the first window as "LTCG-rate consistent," inside the second as "ordinary-income-rate consistent," outside both windows but still in $(0, 1)$ as "plausible sign, outside bands," and non-positive or greater than one as "out of band". The "shared-tau coherence" column then flags the joint sign pattern of the two pooled medians: cells outside the canonical sign-consistent pattern (tax ≤ 0, net ≥ 0) indicate that the model's two answers are not jointly consistent with any single τ — either because the model holds two independent literature anchors, or because the model answered one convention with the opposite sign. Twenty-eight of twenty-nine models are sign-consistent; the exception (GPT-5.4 nano) returns two negative medians, so the identity's premise fails and no implied τ is defined for it. Among the sign-consistent models, one (Gemini 3.5 Flash) has a bootstrap distribution that straddles the $\rho = 1$ pole — only 65% of its draws fall in $(0, 1)$ — so the table flags it uninformative and assigns no band. The substantive finding concerns the remaining twenty-seven: eighteen cluster in the ordinary-income-rate window and nine in the LTCG window, so feeding the models' own responsiveness estimates into the inversion typically recovers an implied τ closer to the top marginal ordinary-income rate than to the top LTCG rate. This is the most direct internal-consistency check the design supports, since the sibling probes the exact same economic object from the other side.

One wording caveat scopes the audit. The seven original-wording April models (see the Design disclosure) answered both conventions under the pre-revision clarifier text: conventional-direction-first conditionals rather than the symmetric if-and-only-if form, and — specific to the sibling — a definition line whose conversion identity the original wording stated backwards, corrected in the April 21 revision. The banded outcome does not visibly hinge on the split: the ordinary-income-window majority holds within the revised-wording group alone (13 of its 21 banded models, alongside 5 of the 6 banded original-wording models), and the panel's one sign-inconsistent cell (GPT-5.4 nano) is not explained by the ordering cue, since the original sibling wording listed the conventional positive direction first. But the seven models' audit rows were elicited under measurably different clarifier text, and readers comparing individual rows across the wording groups should carry that caveat. The within-model ablation in Appendix Table A19 puts numbers on it: re-eliciting the four premium-tier April models under the revised wording leaves Claude Opus 4.7 and Grok 4.20 in place but moves Claude Sonnet 4.6 from below the LTCG window into it (implied-tau median 0.123 to 0.167) and Gemini 3.1 Pro from the ordinary-income window into the LTCG window (0.545 to 0.259), so individual band assignments for the seven original-wording rows should be read as wording-sensitive. Both movers moved toward the LTCG window; even in the extreme case where every banded ordinary-income original-wording row did the same, the eighteen-to-nine ordinary-income majority would narrow to a thirteen-to-fourteen split — essentially parity, tipped a hair the other way, rather than a decisive reversal.

Appendix Table A9. Capital-gains convention audit.

Note: Both capital-gains-realizations conventions elicited independently under prompt v4. Under the identity epsilon_taxrate = -(tau / (1 - tau)) * epsilon_netoftax, any model whose

two answers are jointly consistent with a single tau prior implies one specific tau. The implied-tau column reports the median and 90 percent interval of 1000 bootstrap draws that independently sample one tax-rate run and one net-of-tax-rate run from the 15 runs of each; this is not the plug-in ratio of medians, which differs from the bootstrap median because tau = -rho / (1 - rho) is nonlinear in rho (Jensen's inequality). The band flag is computed against the bootstrap median. 'LTCG-rate consistent' marks an implied median tau in [0.15, 0.37] (U.S. top-bracket long-term-capital-gains envelope, federal LTCG plus NIIT plus high state); 'ordinary-income-rate consistent' marks (0.37, 0.55] (federal ordinary-income top plus high state); 'plausible sign, outside bands' marks (0, 1) outside both windows; and 'out of band' marks a non-positive or > 1 median tau. The shared-tau coherence column flags the joint sign pattern of the two pooled medians: a model with (tax < 0, net > 0) is sign-consistent with the identity, whereas (both positive, both negative, or reversed) indicates that the two answers are not jointly consistent with a single tau prior — either because the model holds two independent literature anchors, or because one convention was answered with the opposite sign.

Model	Organization	epsilon w.r.t. tax rate (median)	epsilon w.r.t. net-of-tax rate (median)	Implied tau median [90%]	Share of draws in (0, 1)	Shared-tau coherence	Band (LTCG [0.15, 0.37], ordinary-income [0.37, 0.55])
GPT-5.4 nano	OpenAI	-0.35	-0.2	— (premise fails)	—	both negative	not identified
Claude Fable 5	Anthropic	-0.7	1.8	0.280 [0.250, 0.318]	100%	sign-consistent (tax<0, net>0)	LTCG-rate consistent
Claude Haiku 4.5	Anthropic	-0.8	0.8	0.500 [0.348, 0.552]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
Claude Opus 4.7	Anthropic	-0.7	0.7	0.500 [0.500, 0.500]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent

Model	Organization	epsilon w.r.t. tax rate (median)	epsilon w.r.t. net-of-tax rate (median)	Implied tau median [90%]	Share of draws in (0, 1)	Shared-tau coherence	Band (LTCG [0.15, 0.37], ordinary-income [0.37, 0.55])
Claude Opus 4.8	Anthropic	-0.7	0.7	0.500 [0.500, 0.538]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
Claude Opus 5	Anthropic	-0.72	1.5	0.333 [0.265, 0.385]	100%	sign-consistent (tax<0, net>0)	LTCG-rate consistent
Claude Sonnet 4.6	Anthropic	-0.7	3.5	0.167 [0.167, 0.189]	100%	sign-consistent (tax<0, net>0)	LTCG-rate consistent
Claude Sonnet 5	Anthropic	-0.6	0.6	0.500 [0.500, 0.538]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
DeepSeek V4 Pro	DeepSeek	-0.45	0.7	0.310 [0.067, 0.500]	100%	sign-consistent (tax<0, net>0)	LTCG-rate consistent
GLM-5.2	Zhipu AI	-0.5	1.2	0.294 [0.091, 0.611]	100%	sign-consistent (tax<0, net>0)	LTCG-rate consistent
GPT-5.4	OpenAI	-0.7	0.9	0.450 [0.389, 0.571]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent

Model	Organization	epsilon w.r.t. tax rate (median)	epsilon w.r.t. net-of-tax rate (median)	Implied tau median [90%]	Share of draws in (0, 1)	Shared-tau coherence	Band (LTCG [0.15, 0.37], ordinary-income [0.37, 0.55])
GPT-5.4 mini	OpenAI	-0.8	1.2	0.500 [0.333, 0.636]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
GPT-5.5	OpenAI	-0.65	0.9	0.400 [0.207, 0.519]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
GPT-5.6 Luna	OpenAI	-0.5	0.7	0.417 [0.222, 0.682]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
GPT-5.6 Sol	OpenAI	-0.6	0.8	0.389 [0.250, 0.520]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
GPT-5.6 Terra	OpenAI	-0.35	0.7	0.364 [0.263, 0.545]	100%	sign-consistent (tax<0, net>0)	LTCG-rate consistent
Gemini 3 Flash	Google	-0.8	0.8	0.500 [0.467, 0.515]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
Gemini 3.1 Flash-Lite	Google	-0.75	0.8	0.484 [0.318, 0.538]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent

Model	Organization	epsilon w.r.t. tax rate (median)	epsilon w.r.t. net-of-tax rate (median)	Implied tau median [90%]	Share of draws in (0, 1)	Shared-tau coherence	Band (LTCG [0.15, 0.37], ordinary-income [0.37, 0.55])
Gemini 3.1 Pro	Google	-0.7	2	0.259 [0.200, 0.483]	100%	sign-consistent (tax<0, net>0)	LTCG-rate consistent
Gemini 3.5 Flash	Google	-0.4	0.8	0.452 [-0.889, 3.250]	65%	sign-consistent (tax<0, net>0)	uninformative (pole-straddling)
Gemini 3.6 Flash	Google	-0.65	0.7	0.464 [0.417, 0.500]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
Grok 4.1 Fast	xAI	-0.7	1.2	0.368 [0.149, 0.500]	100%	sign-consistent (tax<0, net>0)	LTCG-rate consistent
Grok 4.20	xAI	-0.5	0.65	0.417 [0.348, 0.478]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
Grok 4.3	xAI	-0.7	0.8	0.484 [0.411, 0.556]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
Grok 4.5	xAI	-0.6	0.7	0.462 [0.368, 0.519]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent

Model	Organization	epsilon w.r.t. tax rate (median)	epsilon w.r.t. net-of-tax rate (median)	Implied tau median [90%]	Share of draws in (0, 1)	Shared-tau coherence	Band (LTCG [0.15, 0.37], ordinary-income [0.37, 0.55])
Kimi K2.6	Moonshot AI	-0.5	0.65	0.400 [0.185, 0.560]	94%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
Kimi K3	Moonshot AI	-0.5	0.5	0.444 [0.143, 0.545]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent
MiniMax M3	MiniMax	-0.35	0.7	0.333 [0.059, 0.588]	100%	sign-consistent (tax<0, net>0)	LTCG-rate consistent
Qwen 3.7 Max	Alibaba	-0.7	0.7	0.500 [0.412, 0.588]	100%	sign-consistent (tax<0, net>0)	ordinary-income-rate consistent

9 Appendix: Simulation-Facing Coefficients

The simulation-facing substitution-response coefficients are substantively useful, but they are implementation-facing response parameters, not textbook elasticity objects, so I keep them out of the headline rankings and report them here.

Appendix Table A10. Simulation-facing model overview.

Note: Simulation-facing subpanel only: 12 PolicyEngine-style substitution-response coefficients used in a U.S. tax-benefit microsimulation. These rows are reported separately from the canonical elasticity panel because they are implementation-facing response parameters rather than standard elasticity objects.

Model	Organization	Avg abs-elasticity rank (1=highest)	Avg predictive-uncertainty rank (1=narrowest)	Mean absolute pooled center	Mean pooled 90% width	Success rate	Cost / successful run
GPT-5.4 nano	OpenAI	1.46	28.5	0.507	1.319	100.0%	$0.0003
Grok 4.20	xAI	5.92	22.25	0.277	0.765	100.0%	$0.0086
Grok 4.1 Fast	xAI	6.62	27.75	0.295	1.219	100.0%	$0.0002
Qwen 3.7 Max	Alibaba	7.62	22.58	0.288	0.813	100.0%	—
GPT-5.6 Terra	OpenAI	7.67	17.92	0.272	0.658	100.0%	—
Claude Opus 4.7	Anthropic	8.04	16.42	0.267	0.637	100.0%	$0.0216
Claude Opus 4.8	Anthropic	8.33	16	0.267	0.626	100.0%	$0.0141
Grok 4.5	xAI	8.33	19.08	0.266	0.665	100.0%	$0.0064
Grok 4.3	xAI	9.04	20.33	0.262	0.681	100.0%	$0.0017
Claude Opus 5	Anthropic	11.17	8.92	0.253	0.509	100.0%	—
GLM-5.2	Zhipu AI	11.88	12.08	0.254	0.573	100.0%	—
Claude Haiku 4.5	Anthropic	12.25	20	0.247	0.678	100.0%	$0.0031
Claude Sonnet 4.6	Anthropic	12.96	10.33	0.241	0.522	100.0%	$0.0109
Kimi K2.6	Moonshot AI	13.33	21.17	0.24	0.707	100.0%	—
Claude Sonnet 5	Anthropic	15.83	15.58	0.228	0.631	100.0%	$0.0073
GPT-5.6 Luna	OpenAI	16.62	24.33	0.216	0.797	100.0%	—
Claude Fable 5	Anthropic	16.75	3.08	0.222	0.42	100.0%	$0.0468
GPT-5.4 mini	OpenAI	16.92	23.83	0.218	0.78	100.0%	$0.0010
Gemini 3.1 Pro	Google	17.71	3.5	0.216	0.432	100.0%	$0.0078

Model	Organization	Avg abs-elasticity rank (1=highest)	Avg predictive-uncertainty rank (1=narrowest)	Mean absolute pooled center	Mean pooled 90% width	Success rate	Cost / successful run
Kimi K3	Moonshot AI	17.96	16.42	0.215	0.628	100.0%	—
GPT-5.6 Sol	OpenAI	18.21	10.75	0.211	0.536	100.0%	—
DeepSeek V4 Pro	DeepSeek	19.46	18.33	0.2	0.66	100.0%	—
Gemini 3.6 Flash	Google	22.29	4	0.178	0.454	100.0%	$0.0068
GPT-5.4	OpenAI	22.5	7.08	0.18	0.484	100.0%	$0.0036
MiniMax M3	MiniMax	23.33	15.92	0.173	0.657	100.0%	—
GPT-5.5	OpenAI	24.71	11.17	0.17	0.54	100.0%	$0.0137
Gemini 3 Flash	Google	25.42	7.92	0.163	0.503	100.0%	$0.0010
Gemini 3.5 Flash	Google	25.75	3.92	0.165	0.446	100.0%	$0.0100
Gemini 3.1 Flash-Lite	Google	26.92	5.83	0.13	0.456	100.0%	$0.0008

Appendix Table A11. Simulation-facing disagreement.

Note: Simulation-facing PolicyEngine substitution-response coefficients only, sorted by cross-model spread in pooled point estimates.

Quantity	Lowest model	Lowest center	Highest model	Highest center	Spread	Mean pooled 90% width	Spread / mean width
Primary-earner substitution elasticity in a tax-benefit simulation, decile 7	Gemini 3.1 Flash-Lite	0.086	GPT-5.4 nano	0.623	0.537	0.593	0.907
Primary-earner substitution elasticity in a tax-benefit simulation, decile 8	Gemini 3.1 Flash-Lite	0.11	GPT-5.4 nano	0.609	0.499	0.604	0.825
Primary-earner substitution elasticity in a tax-benefit simulation, decile 6	MiniMax M3	0.114	GPT-5.4 nano	0.59	0.476	0.583	0.817

Quantity	Lowest model	Lowest center	Highest model	Highest center	Spread	Mean pooled 90% width	Spread / mean width
Primary-earner substitution elasticity in a tax-benefit simulation, decile 5	Gemini 3 Flash	0.121	GPT-5.4 nano	0.593	0.473	0.603	0.784
Primary-earner substitution elasticity in a tax-benefit simulation, decile 10	Gemini 3.5 Flash	0.14	GPT-5.4 nano	0.603	0.463	0.643	0.721
Primary-earner substitution elasticity in a tax-benefit simulation, decile 9	Gemini 3.1 Flash-Lite	0.129	GPT-5.4 nano	0.587	0.458	0.629	0.729

Quantity	Lowest model	Lowest center	Highest model	Highest center	Spread	Mean pooled 90% width	Spread / mean width
Primary-earner substitution elasticity in a tax-benefit simulation, decile 3	Gemini 3.1 Flash-Lite	0.069	GPT-5.4 nano	0.52	0.451	0.595	0.758
Secondary-earner substitution elasticity in a tax-benefit simulation	GPT-5.6 Sol	0.284	Qwen 3.7 Max	0.701	0.417	1.001	0.417
Primary-earner substitution elasticity in a tax-benefit simulation, decile 4	Gemini 3.1 Flash-Lite	0.073	GPT-5.4 nano	0.465	0.391	0.594	0.658

Quantity	Lowest model	Lowest center	Highest model	Highest center	Spread	Mean pooled 90% width	Spread / mean width
Primary-earner substitution elasticity in a tax-benefit simulation, decile 2	MiniMax M3	0.069	GPT-5.4 nano	0.376	0.307	0.597	0.515
Primary-earner substitution elasticity in a tax-benefit simulation, decile 1	Gemini 3.1 Flash-Lite	0.05	GPT-5.4 nano	0.343	0.293	0.63	0.466
Substitution elasticity of labor supply in a tax-benefit simulation	Gemini 3.1 Flash-Lite	0.153	GPT-5.6 Terra	0.31	0.157	0.707	0.222

9.1 Static flat-tax plus demogrant frontier

Appendix Table A12 reports a static blank-slate flat-tax benchmark on the same Enhanced CPS 2024 microdata (PolicyEngine Team 2024). Each row applies a flat tax to PolicyEngine's current positive-AGI base and rebates the resulting revenue as an equal per-person demogrant using tax-unit size. This is the modernized analogue of a blank-slate UBI-style exercise in the current PolicyEngine stack.

This is a pure distributional comparison, not an optimal-tax result: the benchmark holds behavior fixed and omits leisure. Mean per-person resources are mechanically constant across rows, so the informative objects are the demogrant, the distributional quantiles, and the Gini of per-person post-tax positive-AGI resources.

Appendix Table A12. Static flat positive-AGI tax plus demogrant frontier.

Note: Static PolicyEngine benchmark on Enhanced CPS 2024 microdata. Each row applies a flat tax to the current positive-AGI base and rebates the revenue as an equal per-person demogrant using tax-unit size. This is a distributional frontier, not a behavioral or leisure-adjusted optimal-tax exercise. Mean per-person resources are mechanically constant across rows, so the informative objects are the demogrant, the distributional quantiles, and the Gini of per-person post-tax positive-AGI resources.

Flat tax rate	Demogrant per person	P10 post-tax resources	Median post-tax resources	P90 post-tax resources	Gini
0%	$0	$1,554	$24,091	$85,300	0.627
20%	$8,881	$10,125	$28,154	$77,121	0.501
40%	$17,763	$18,695	$32,217	$68,942	0.376
60%	$26,644	$27,266	$36,280	$60,764	0.251
80%	$35,525	$35,836	$40,343	$52,585	0.125
95%	$42,186	$42,264	$43,391	$46,451	0.031

9.2 Top-rate robustness to Pareto tail and CRRA curvature

Appendix Table A13 reports the top-rate mapping from Table 4 under alternative values of the Pareto tail parameter a and the CRRA curvature γ. The column-wise comparison asks how the implied optimal top rate moves when the microdata-calibrated $a = 1.621$ becomes a Pareto tail at 1.3, 1.5, or 1.7, or when log utility becomes CRRA at $\gamma = 2$ with a held at the microdata estimate. The ETI median feeding each row is the same pooled-mixture median used in Table 4, so column-wise differences reflect only the formula's (a, γ) pair, not any change in the elicited responses. Across all parameterizations, the cross-model ordering never changes.

Appendix Table A13. Top-rate robustness to Pareto tail and CRRA curvature.

Note: Robustness of the utilitarian optimal top-rate mapping in Table 4 to the Pareto tail parameter a and the CRRA coefficient gamma. Each cell is the median implied optimal top rate tau* = (1 - g_bar) / (1 - g_bar + a e) computed at the model's pooled ETI median under the (a, gamma) pair in the column header, where g_bar = a / (a + gamma). The baseline column (a = 1.621, gamma = 1) reproduces the median column in Table 4. The a = 1.3 / 1.5 / 1.7 columns vary only the Pareto tail while keeping log utility (gamma = 1); the final column replaces log utility with CRRA at gamma = 2 while holding a at the baseline value

above. Under log utility the corresponding welfare weights g_bar are 1.3/2.3 = 0.565, 1.621/(1 + 1.621) = 0.618, 1.5/2.5 = 0.600, and 1.7/2.7 = 0.630; under gamma = 2 with a = 1.621, g_bar = 1.621/(1.621 + 2) = 0.448.

Model	ETI median	Baseline top rate (a=1.621, gamma=1)	Top rate (a=1.3, gamma=1)	Top rate (a=1.5, gamma=1)	Top rate (a=1.7, gamma=1)	Top rate (a=1.621, gamma=2)
Claude Opus 5	0.337	41.2%	49.8%	44.2%	39.3%	50.3%
Gemini 3.1 Pro	0.351	40.2%	48.8%	43.2%	38.3%	49.3%
Gemini 3.5 Flash	0.357	39.8%	48.4%	42.8%	37.9%	48.9%
GPT-5.5	0.369	39.0%	47.6%	42.0%	37.1%	48.0%
Kimi K3	0.371	38.8%	47.4%	41.8%	37.0%	47.8%
GPT-5.6 Luna	0.377	38.4%	47.0%	41.4%	36.6%	47.5%
Claude Opus 4.8	0.383	38.0%	46.6%	41.0%	36.2%	47.1%
Gemini 3.1 Flash-Lite	0.389	37.7%	46.2%	40.7%	35.9%	46.7%
Claude Opus 4.7	0.4	37.0%	45.5%	40.0%	35.3%	46.0%
Gemini 3 Flash	0.4	37.0%	45.5%	40.0%	35.3%	46.0%
Grok 4.1 Fast	0.4	37.0%	45.5%	40.0%	35.3%	46.0%
Grok 4.5	0.41	36.5%	44.9%	39.4%	34.7%	45.4%
GPT-5.4	0.42	35.9%	44.3%	38.8%	34.2%	44.8%
Gemini 3.6 Flash	0.423	35.8%	44.2%	38.7%	34.0%	44.6%
GPT-5.6 Sol	0.431	35.3%	43.7%	38.2%	33.6%	44.2%
Claude Fable 5	0.437	35.0%	43.4%	37.9%	33.3%	43.8%
GPT-5.4 mini	0.437	35.0%	43.4%	37.9%	33.3%	43.8%
Grok 4.3	0.439	34.9%	43.3%	37.8%	33.2%	43.7%
MiniMax M3	0.438	34.9%	43.3%	37.8%	33.2%	43.7%

Model	ETI median	Baseline top rate (a=1.621, gamma=1)	Top rate (a=1.3, gamma=1)	Top rate (a=1.5, gamma=1)	Top rate (a=1.7, gamma=1)	Top rate (a=1.621, gamma=2)
Claude Sonnet 5	0.471	33.3%	41.5%	36.1%	31.6%	42.0%
DeepSeek V4 Pro	0.479	32.9%	41.1%	35.8%	31.3%	41.6%
GPT-5.6 Terra	0.492	32.4%	40.5%	35.2%	30.7%	40.9%
GLM-5.2	0.495	32.2%	40.3%	35.0%	30.6%	40.8%
Kimi K2.6	0.499	32.1%	40.1%	34.8%	30.4%	40.6%
Claude Sonnet 4.6	0.5	32.0%	40.1%	34.8%	30.3%	40.5%
Grok 4.20	0.5	32.0%	40.1%	34.8%	30.3%	40.5%
Claude Haiku 4.5	0.502	31.9%	40.0%	34.7%	30.3%	40.4%
GPT-5.4 nano	0.546	30.1%	38.0%	32.8%	28.5%	38.4%
Qwen 3.7 Max	0.555	29.8%	37.6%	32.4%	28.2%	38.0%

9.3 Resampling standard errors and rank stability

Appendix Table A14 resamples the 15 runs within each canonical cell (200 bootstrap resamples, fixed seed) to attach Monte Carlo standard errors to the pooled center and pooled 90 percent width, and to test whether the predictive-uncertainty ordering survives run-level resampling. Median center standard errors are small in absolute terms for every model, relative width standard errors run 1 to 9 percent, and each model's average width rank carries a narrow 90 percent resampling interval. The width ordering reported in the main text is therefore signal, not $R = 15$ noise. This diagnostic covers the canonical-panel width ordering; the subpanel top-three and bottom-three callouts in Tables 1-2 average over fewer quantities and carry correspondingly wider resampling bands, so read them as coarse groupings, not exact placements.

Appendix Table A14. Monte Carlo resampling of pooled summaries.

Note: Monte Carlo uncertainty in the pooled summaries from resampling the 15 runs within each canonical cell (200 bootstrap resamples, fixed seed). Center MC SE is the standard error of the pooled center; relative width MC SE is the standard error of the pooled 90% width divided by its mean. The final columns show the distribution of each model's average width

rank across resamples; narrow intervals indicate the predictive-uncertainty ordering is stable to run-level resampling at R = 15.

Model	Median center MC SE	Median relative width MC SE	Avg width rank (mean)	Avg width rank 90% interval
Claude Fable 5	0.003	3%	6.99	[6.15, 7.62]
Gemini 3.6 Flash	0.006	2%	7.15	[6.46, 7.85]
Claude Haiku 4.5	0.005	5%	8.68	[7.84, 9.39]
Claude Opus 4.7	0	2%	9.44	[8.46, 10.23]
Claude Sonnet 5	0.003	2%	9.62	[9.00, 10.31]
Gemini 3.1 Pro	0.006	2%	10.2	[9.46, 10.77]
Claude Opus 4.8	0	1%	10.78	[10.08, 11.31]
Gemini 3 Flash	0	1%	12.31	[11.53, 12.93]
Gemini 3.5 Flash	0.01	3%	12.57	[11.92, 13.15]
GPT-5.6 Terra	0.004	3%	12.86	[11.92, 13.77]
Claude Opus 5	0.008	3%	12.87	[12.15, 13.62]
Gemini 3.1 Flash-Lite	0.006	5%	13.01	[12.00, 13.85]
Claude Sonnet 4.6	0	1%	13.32	[12.53, 14.15]
GPT-5.5	0.005	2%	14.43	[13.92, 14.93]
GPT-5.4	0	3%	14.44	[13.69, 15.16]
GPT-5.6 Sol	0.008	3%	16.41	[15.61, 17.00]
MiniMax M3	0.017	9%	16.99	[15.38, 18.31]
GPT-5.4 nano	0.025	6%	17.62	[16.77, 18.23]
Kimi K3	0.009	3%	17.63	[16.77, 18.38]
Grok 4.1 Fast	0	2%	17.66	[16.15, 18.85]
Grok 4.5	0.009	3%	17.73	[16.84, 18.54]
Grok 4.3	0.012	5%	18.05	[17.08, 18.93]
Qwen 3.7 Max	0.013	8%	18.2	[16.30, 19.69]
DeepSeek V4 Pro	0.014	7%	18.44	[17.23, 19.54]
GLM-5.2	0.017	7%	18.58	[17.23, 19.85]
Kimi K2.6	0.028	6%	20.35	[18.77, 21.54]
GPT-5.4 mini	0.009	4%	21.64	[20.69, 22.38]
Grok 4.20	0.011	3%	22.58	[21.85, 23.31]
GPT-5.6 Luna	0.013	4%	24.48	[23.69, 25.16]

9.4 Within-run versus between-run variance

Appendix Table A15 splits pooled predictive variance into its two components: the within-run term (the model's own stated p05-p95 quantiles) and the between-run term (variation of run means across repeated draws). The between-run share is a median of 0 to 2 percent for every model, so what the models state dominates both the pooled widths and the predictive-uncertainty ranking built on them; sampling noise contributes little. The maximum single-cell shares are larger — 37% for minimax-m3, 29% for gpt-5.4-nano, and 25% for gemini-3.5-flash, concentrated in sign-unstable cells — so the median, not the maximum, characterizes the typical cell. This also bounds the concern that the three no-sampling-parameter Claude models rank under a different draw regime: the component their regime affects is a negligible share of the total for every model in the panel. (The between-run term uses the population variance over the 15 run means, a mild downward bias that is immaterial given these shares; the headline interval is the nonparametric mixture, not a Gaussian built from the summed variance.)

Appendix Table A15. Variance decomposition of pooled predictive spread.

Note: Split of the pooled predictive variance over the canonical 13-quantity subpanel into the within-run component (the model's own stated p05-p95 quantiles) and the between-run component (variation of run means across the 15 repeated draws). The between-run share is the only component affected by the sampling regime, which differs for the three no-sampling-parameter Claude models.

Model	Cells	Median within-run SD	Median between-run SD	Median between-run variance share	Max between-run variance share
Claude Fable 5	13	0.679	0.012	0%	1%
Claude Haiku 4.5	13	0.672	0.018	0%	4%
Claude Opus 4.7	13	0.704	0	0%	0%
Claude Opus 4.8	13	0.72	0	0%	1%
Claude Opus 5	13	0.733	0.029	0%	1%
Claude Sonnet 4.6	13	0.726	0	0%	1%
Claude Sonnet 5	13	0.699	0.01	0%	1%
DeepSeek V4 Pro	13	0.738	0.054	1%	4%

Model	Cells	Median within-run SD	Median between-run SD	Median between-run variance share	Max between-run variance share
GLM-5.2	13	0.683	0.06	1%	13%
GPT-5.4	13	0.684	0	0%	1%
GPT-5.4 mini	13	0.713	0.034	1%	5%
GPT-5.4 nano	13	0.699	0.098	2%	29%
GPT-5.5	13	0.703	0.018	0%	1%
GPT-5.6 Luna	13	0.727	0.051	0%	5%
GPT-5.6 Sol	13	0.692	0.036	0%	1%
GPT-5.6 Terra	13	0.678	0.014	0%	2%
Gemini 3 Flash	13	0.677	0	0%	1%
Gemini 3.1 Flash-Lite	13	0.687	0.025	0%	3%
Gemini 3.1 Pro	13	0.689	0.025	0%	3%
Gemini 3.5 Flash	13	0.683	0.041	0%	25%
Gemini 3.6 Flash	13	0.701	0.022	0%	3%
Grok 4.1 Fast	13	0.743	0	0%	4%
Grok 4.20	13	0.706	0.042	0%	7%
Grok 4.3	13	0.68	0.046	0%	6%
Grok 4.5	13	0.697	0.034	0%	4%
Kimi K2.6	13	0.701	0.1	1%	6%
Kimi K3	13	0.726	0.034	0%	2%
MiniMax M3	13	0.692	0.062	1%	37%
Qwen 3.7 Max	13	0.728	0.048	0%	18%

9.5 Harness disclosure

Appendix Table A16 reports the full generation-harness configuration per model. The repeated-run design is identical across models, and the prompt text is byte-identical across models on 23 of the 26 quantities (the three sign-clarified quantities carry the two v4 clarifier wordings disclosed

in the Design section); the structured-output mechanism, completion budget, sampling regime, and reasoning configuration follow each provider's API surface and are therefore confounded with model identity. Completion budgets are truncation guards, not elicitation content — reasoning tokens count against them on models that reason, so I raised budgets where required to avoid truncation. One further asymmetry: the OpenAI path wraps the elicitation prompt with a one-line system message ("Follow the user's instructions exactly and return only the final answer."), while the Anthropic and LiteLLM paths send the prompt as a bare user message; the instruction is format-only and the OpenAI request builder attaches it unconditionally (the committed request logs record usage metadata, not message payloads, so the string is checkable in llm_econ_beliefs/providers.py).

Appendix Table A16. Per-model generation-harness configuration.

Note: Per-model generation-harness configuration. The prompt text and repeated-run design are identical across models; the structured-output mechanism, completion budget, sampling regime, and reasoning configuration follow each provider's API surface and are therefore confounded with model identity. Completion budgets are truncation guards: reasoning tokens count against them on models that reason, so budgets were raised where required to avoid truncation. Identifiers marked alias float with provider updates; dated snapshots are pinned.

Model	Provider path	Output mechanism	Completion budget	Sampling	Reasoning config	API identifier	Identifier type
GPT-5.5	OpenAI Chat Completions	strict JSON schema	1200 (8000 for the 40 re-elicited runs)	temperature 1.0, batched n <= 8	provider default effort	gpt-5.5	alias
GPT-5.6 Sol	OpenAI Chat Completions	strict JSON schema	8000	temperature 1.0, batched n <= 8	provider default effort	gpt-5.6-sol	alias
GPT-5.6 Luna	OpenAI Chat Completions	strict JSON schema	8000	temperature 1.0, batched n <= 8	provider default effort	gpt-5.6-luna	alias
GPT-5.6 Terra	OpenAI Chat Completions	strict JSON schema	8000	temperature 1.0, batched n <= 8	provider default effort	gpt-5.6-terra	alias

Model	Provider path	Output mechanism	Completion budget	Sampling	Reasoning config	API identifier	Identifier type
GPT-5.4	OpenAI Chat Completions	strict JSON schema	1200	temperature 1.0, batched n <= 8	provider default effort	gpt-5.4	alias
GPT-5.4 mini	OpenAI Chat Completions	strict JSON schema	1200	temperature 1.0, batched n <= 8	provider default effort	gpt-5.4-mini	alias
GPT-5.4 nano	OpenAI Chat Completions	strict JSON schema	1200	temperature 1.0, batched n <= 8	provider default effort	gpt-5.4-nano	alias
Claude Fable 5	native Anthropic API	strict JSON schema	32000	none accepted (provider default)	always-on reasoning	claude-fable-5	alias
Claude Opus 4.8	native Anthropic API	strict JSON schema	32000	none accepted (provider default)	off (provider default)	claude-opus-4-8	alias
Claude Sonnet 5	native Anthropic API	strict JSON schema	32000	none accepted (provider default)	adaptive (provider default)	claude-sonnet-5	alias
Claude Opus 5	native Anthropic API	strict JSON schema	32000	none accepted (provider default)	adaptive, on by default (provider default)	claude-opus-5	alias
Claude Opus 4.7	LiteLLM	forced function call	1200	temperature 1.0	off (provider default)	claude-opus-4-7	alias
Claude Sonnet 4.6	LiteLLM	forced function call	1200	temperature 1.0	off (provider default)	claude-sonnet-4-6	alias
Claude Haiku 4.5	LiteLLM	forced function call	1200	temperature 1.0	off (provider default)	claude-haiku-4-5-20251001	dated snapshot

Model	Provider path	Output mechanism	Completion budget	Sampling	Reasoning config	API identifier	Identifier type
Gemini 3.1 Pro	LiteLLM	forced JSON object	1200	temperature 1.0	provider default thinking	gemini-3.1-pro-preview	preview alias
Gemini 3.5 Flash	LiteLLM	forced JSON object	4000	temperature 1.0	provider default thinking	gemini-3.5-flash	alias
Gemini 3.6 Flash	LiteLLM	forced JSON object	8000	temperature 1.0	provider default thinking	gemini-3.6-flash	alias
Gemini 3 Flash	LiteLLM	forced JSON object	1200	temperature 1.0	provider default thinking	gemini-3-flash-preview	preview alias
Gemini 3.1 Flash-Lite	LiteLLM	forced JSON object	1200	temperature 1.0	provider default thinking	gemini-3.1-flash-lite-preview	preview alias
Grok 4.20	LiteLLM	forced function call	1200	temperature 1.0	reasoning variant	xai/grok-4.20-reasoning	alias
Grok 4.3	LiteLLM	forced function call	4000	temperature 1.0	provider default	xai/grok-4.3	alias
Grok 4.5	LiteLLM	forced function call	8000	temperature 1.0	provider default	xai/grok-4.5	alias
DeepSeek V4 Pro	LiteLLM via OpenRouter	forced JSON object (schema validated locally)	8000	temperature 1.0	provider default	openrouter/deepseek/deepseek-v4-pro	alias
Qwen 3.7 Max	LiteLLM via OpenRouter	forced JSON object (schema validated locally)	8000	temperature 1.0	provider default	openrouter/qwen/qwen3.7-max	alias

Model	Provider path	Output mechanism	Completion budget	Sampling	Reasoning config	API identifier	Identifier type
Kimi K2.6	LiteLLM via OpenRouter	forced JSON object (schema validated locally)	8000	temperature 1.0	provider default	openrouter/moonshotai/kimi-k2.6	alias
Kimi K3	LiteLLM via OpenRouter	forced JSON object (schema validated locally)	8000	temperature 1.0	provider default	openrouter/moonshotai/kimi-k3	alias
GLM-5.2	LiteLLM via OpenRouter	forced JSON object (schema validated locally)	16000	temperature 1.0	provider default	openrouter/z-ai/glm-5.2	alias
MiniMax M3	LiteLLM via OpenRouter	forced JSON object (schema validated locally)	8000	temperature 1.0	provider default	openrouter/minimax/minimax-m3	alias
Grok 4.1 Fast	LiteLLM	forced function call	1200	temperature 1.0	non-reasoning variant	xai/grok-4-1-fast-non-reasoning	alias

9.6 Cross-mechanism ablation

Appendix Table A17 addresses the mechanism confound directly by re-eliciting Claude Opus 4.7 — an April model originally run through the LiteLLM forced-function-call path at a 1,200-token budget — through the July native strict-JSON-schema path at a 32,000-token budget, on the nine canonical elasticities with 15 fresh runs each. Pooled centers move by at most 0.03 (median 0.00); pooled widths move more, from -15 to +7 percent across the nine quantities with no consistent direction. Within the panel's resolution, the harness mechanism does not move elicited centers, which supports reading cross-wave differences in centers as model differences;

width comparisons across waves carry the extra mechanism noise. The ablation isolates the output mechanism and completion budget only — Claude Opus 4.7 runs without extended reasoning on both paths, so the July models' always-on or adaptive reasoning modes remain confounded with model identity.

Appendix Table A17. Same model, two harness mechanisms.

Note: Same model (Claude Opus 4.7), same v4 prompts, same repeated-run design, two harness mechanisms: the April LiteLLM forced-function-call path (temperature 1.0, 1200-token budget) versus the July native Anthropic strict-JSON-schema path (no sampling parameters, 32000-token budget). Each cell pools 15 fresh runs elicited in July 2026 under the native mechanism against the April panel cell. Centers are stable (max absolute change 0.03); widths move from -15 to +7 percent across quantities with no consistent direction. Opus 4.7 runs without extended reasoning on both paths, so the reasoning-mode axis is not covered.

Quantity	LiteLLM center	Native center	Center change	LiteLLM 90% width	Native 90% width
Armington elasticity	1.533	1.533	0	3.895	4.057
Capital gains realizations elasticity	-0.7	-0.7	0	1.527	1.63
Elasticity of substitution between capital and labor	0.6	0.6	0	0.903	0.885
Elasticity of taxable income	0.4	0.4	0	0.873	0.892
Employment participation elasticity of single mothers	0.7	0.71	0.01	1.196	1.198
Frisch elasticity of labor supply	0.417	0.387	-0.03	1.318	1.119
Income elasticity of labor supply	-0.05	-0.05	0	0.22	0.22

Quantity	LiteLLM center	Native center	Center change	LiteLLM 90% width	Native 90% width
Intertemporal elasticity of substitution	0.5	0.5	0	1.873	1.698
Uncompensated wage elasticity of labor supply	0.1	0.1	0	0.449	0.4

9.7 Clarifier-wording ablation

Appendix Tables A18-A19 close the loop on the Design section's wording disclosure by running the reconstructible within-model comparison. The four April premium-tier models are the only cells elicited under both v4 clarifier wordings: their superseded April 19 elicitations remain in git history, and their published April 21 re-elicitations carry the revised text. Table A18 pools both elicitations of each sign-clarified quantity with the paper's piecewise-uniform construction — 15 runs per cell, 100 percent parse on both sides — after byte-verifying every cell's archived prompt text: each April 19 prompt equals the original wording that the seven holdout models' committed archives still carry, and each April 21 prompt equals the current builder's revised wording. On the two headline-panel quantities the wording change is immaterial: across the eight model-quantity cells the pooled center moves by at most 0.08 in absolute value and the pooled 90 percent width by at most 0.12. The net-of-tax sibling is different: Claude Sonnet 4.6's pooled center falls from 4.60 to 3.37, and Gemini 3.1 Pro's rises from 0.37 to 2.08.

The Gemini cell also identifies the mechanism, because its answers track whichever conversion identity the sibling's definition line states. Its April 19 median pair $(-0.70, 0.30)$ satisfies the original wording's backwards identity at an internal tax rate near 0.30, and its April 21 median pair $(-0.70, 2.0)$ satisfies the corrected identity at a nearly identical internal rate (0.26) — the model's tax-rate anchor barely moves while the sibling's magnitude follows the stated formula. Algebraically, a model that applies the backwards identity at internal rate τ^* produces a pair whose A9 inversion returns $1 - \tau^*$ rather than τ^*, which is why Gemini's original-wording bootstrap median lands at 0.545 (run-level dispersion pulls it below the plug-in 0.70) while its revised-wording median is 0.259. Claude Sonnet 4.6's move is different in kind: both of its median pairs are consistent with the corrected identity at a low internal rate (0.12 to 0.17), so it did not follow the backwards identity — but its sibling magnitude (median 5.0 to 3.5) was still wording-sensitive. Table A19 reruns the Appendix A9 bootstrap per wording: Claude Opus 4.7 and Grok 4.20 are effectively unmoved, while the two movers cross band boundaries, as discussed in the A9 caveat.

Two qualifications bound the reading. The comparison is not a pure wording experiment: the April 21 re-elicitations also moved to the per-quantity fallback harness that added request logging, and two days elapsed between elicitations, so wording is confounded with harness path and time — the headline-quantity stability bounds the joint effect of all three factors for those quantities, and the identity-tracking pattern is what attributes the sibling moves to the wording. And the published panel and audit numbers for these four models use the April 21 revised wording — the corrected identity — throughout; the original-wording columns describe superseded elicitations retained only for this ablation.

Appendix Table A18. Same model, two clarifier wordings: sign-clarified quantities.

Note: Same four models (the April premium tier), same quantities, same repeated-run design, two v4 clarifier wordings: the superseded April 19 elicitation (original wording — plain conditionals with the conventional direction first; the net-of-tax sibling's definition line states the conversion identity backwards) versus the published April 21 re-elicitation (revised wording — symmetric if-and-only-if clauses, worked magnitude example, corrected identity). April 19 runs are read from git history (commit ddca2375); every cell's prompt text is byte-verified against the original wording preserved in the seven holdout models' committed archives and against the current prompt builder's revised wording. Each cell pools 15 runs with the headline piecewise-uniform mixture. The comparison is not a pure wording experiment: the April 21 re-elicitation also moved to the per-quantity fallback harness that added request logging, and two days elapsed between elicitations, so wording is confounded with harness path and time.

Model	Quantity	Original center	Revised center	Center change	Original 90% width	Revised 90% width
Claude Opus 4.7	Income elasticity of labor supply	-0.05	-0.05	0	0.219	0.22
Claude Opus 4.7	Capital gains realizations elasticity	-0.7	-0.7	0	1.647	1.527
Claude Opus 4.7	Capital gains realizations elasticity (net-of-tax-rate convention)	0.7	0.7	0	1.538	1.652

Model	Quantity	Original center	Revised center	Center change	Original 90% width	Revised 90% width
Claude Sonnet 4.6	Income elasticity of labor supply	-0.093	-0.097	-0.003	0.39	0.395
Claude Sonnet 4.6	Capital gains realizations elasticity	-0.7	-0.7	0	1.4	1.4
Claude Sonnet 4.6	Capital gains realizations elasticity (net-of-tax-rate convention)	4.6	3.367	-1.233	8.871	7.647
Gemini 3.1 Pro	Income elasticity of labor supply	-0.05	-0.073	-0.023	0.241	0.315
Gemini 3.1 Pro	Capital gains realizations elasticity	-0.67	-0.709	-0.039	0.997	0.997
Gemini 3.1 Pro	Capital gains realizations elasticity (net-of-tax-rate convention)	0.373	2.077	1.703	4.832	4.598
Grok 4.20	Income elasticity of labor supply	-0.095	-0.107	-0.011	0.522	0.577
Grok 4.20	Capital gains realizations elasticity	-0.553	-0.473	0.08	1.762	1.807

Model	Quantity	Original center	Revised center	Center change	Original 90% width	Revised 90% width
Grok 4.20	Capital gains realizations elasticity (net-of-tax-rate convention)	0.67	0.65	-0.02	1.854	1.901

Appendix Table A19. Same model, two clarifier wordings: implied tau.

Note: The Appendix A9 implied-tau bootstrap (1,000 draws, fixed seed, tau = -rho / (1 - rho) with rho = epsilon_taxrate / epsilon_netoftax) applied separately to each wording's 15 tax-rate and 15 net-of-tax-rate runs for the four models elicited under both v4 clarifier wordings. Bands as in Appendix Table A9: LTCG-rate [0.15, 0.37], ordinary-income-rate (0.37, 0.55]. All eight model-wording cells are sign-consistent (tax-rate median $< 0 <$ net-of-tax median), so the identity's premise holds throughout. See the Table A18 note for the harness-path confound.

Model	Original implied tau median [90%]	Original share in (0, 1)	Original band	Revised implied tau median [90%]	Revised share in (0, 1)	Revised band	Tau median change
Claude Opus 4.7	0.500 [0.500, 0.500]	100%	ordinary-income-rate consistent	0.500 [0.500, 0.500]	100%	ordinary-income-rate consistent	0
Claude Sonnet 4.6	0.123 [0.123, 0.167]	100%	plausible sign, outside bands	0.167 [0.167, 0.189]	100%	LTCG-rate consistent	0.044
Gemini 3.1 Pro	0.545 [0.375, 3.500]	88%	ordinary-income-rate consistent	0.259 [0.200, 0.483]	100%	LTCG-rate consistent	-0.286

Model	Original implied tau median [90%]	Original share in (0, 1)	Original band	Revised implied tau median [90%]	Revised share in (0, 1)	Revised band	Tau median change
Grok 4.20	0.455 [0.400, 0.500]	100%	ordinary-income-rate consistent	0.417 [0.348, 0.478]	100%	ordinary-income-rate consistent	-0.038

9.8 Support bounds registry

Appendix Table A20 tabulates the per-quantity support bounds that drive quantile-bin reconstruction, tail extrapolation, and the transforms behind the REML and Bayesian estimators. For unbounded sides, the reconstruction extends the support 0.25 x the adjacent inter-quantile gap beyond the elicited p05/p95; the headline pooled 5-95 interval is largely insensitive to this rule because its endpoints sit at or near the elicited quantile knots, while the SD-based and REML/Bayes summaries are more exposed to it.

Appendix Table A20. Registry support bounds.

Note: Registry support bounds used for quantile-bin reconstruction, tail extrapolation, and the transforms behind the REML and Bayesian estimators. Unbounded sides use a tail-extrapolation rule that extends 0.25 x the adjacent inter-quantile gap beyond the elicited p05/p95.

Quantity	Quantity id	Lower support	Upper support
Annual discount factor	household.annual_discount_factor	0	1
Intertemporal elasticity of substitution	household.intertemporal_elasticity_of_substitution	0	5
Coefficient of relative risk aversion	household.relative_risk_aversion.crra	0	50
Employment participation elasticity of single mothers	labor_supply.extensive_margin.single_mothers	-1	5

Quantity	Quantity id	Lower support	Upper support
Frisch elasticity of labor supply	labor_supply.frisch_elasticity.prime_age	0	10
Income elasticity of labor supply	labor_supply.income_elasticity.prime_age	-2	1
Uncompensated wage elasticity of labor supply	labor_supply.marshallian_wage_elasticity.prime_age	-2	4
Substitution elasticity of labor supply in a tax-benefit simulation	labor_supply.policy_response.substitution_elasticity.all	0	5
Primary-earner substitution elasticity in a tax-benefit simulation, decile 1	labor_supply.policy_response.substitution_elasticity.primary.decile_1	0	5
Primary-earner substitution elasticity in a tax-benefit simulation, decile 10	labor_supply.policy_response.substitution_elasticity.primary.decile_10	0	5
Primary-earner substitution elasticity in a tax-benefit simulation, decile 2	labor_supply.policy_response.substitution_elasticity.primary.decile_2	0	5
Primary-earner substitution elasticity in a tax-benefit simulation, decile 3	labor_supply.policy_response.substitution_elasticity.primary.decile_3	0	5
Primary-earner substitution elasticity in a tax-benefit simulation, decile 4	labor_supply.policy_response.substitution_elasticity.primary.decile_4	0	5
Primary-earner substitution elasticity in a tax-benefit simulation, decile 5	labor_supply.policy_response.substitution_elasticity.primary.decile_5	0	5
Primary-earner substitution elasticity in a tax-benefit simulation, decile 6	labor_supply.policy_response.substitution_elasticity.primary.decile_6	0	5

Quantity	Quantity id	Lower support	Upper support
Primary-earner substitution elasticity in a tax-benefit simulation, decile 7	labor_supply.policy_response.substitution_elasticity.primary.decile_7	0	5
Primary-earner substitution elasticity in a tax-benefit simulation, decile 8	labor_supply.policy_response.substitution_elasticity.primary.decile_8	0	5
Primary-earner substitution elasticity in a tax-benefit simulation, decile 9	labor_supply.policy_response.substitution_elasticity.primary.decile_9	0	5
Secondary-earner substitution elasticity in a tax-benefit simulation	labor_supply.policy_response.substitution_elasticity.secondary	0	5
TFP persistence	macro.tfp_persistence.ar1	0	1
Elasticity of substitution between capital and labor	production.capital_labor_substitution	0	10
Capital share in production	production.capital_share	0	1
Capital gains realizations elasticity	tax.capital_gains_realizations.elasticity	-10	2
Capital gains realizations elasticity (net-of-tax-rate convention)	tax.capital_gains_realizations.elasticity.net_of_tax_rate	-2	10
Elasticity of taxable income	tax.elasticity_of_taxable_income.top_earners	-1	5
Armington elasticity	trade.armington_elasticity.import_domestic	0	20

10 Statements

Data accessibility. All elicitation code, raw run-level responses, request logs, generated tables, and the scripts that rebuild every table in this paper are public at https://github.com/PolicyEngine/llm-econ-beliefs. The cached-results reproduction path rebuilds all paper tables without provider API access; full re-elicitation instructions and tracked costs are in the repository README. A versioned archive with a DOI will be deposited on acceptance.

Author contributions. M.G. designed the study, collected the data, performed the analysis, and wrote the paper.

Competing interests. The author is a co-founder of PolicyEngine, whose tax-benefit microsimulation parameters are among the simulation-facing quantities studied in this paper.

Funding. This research received no specific grant from any funding agency.

Ethics. This study involved no human participants or animal subjects; the elicited subjects are commercial language-model APIs.

Use of AI. Language models are the object of study throughout. AI coding assistants helped develop the elicitation harness and draft portions of the manuscript; the author verified all analyses and claims.

Bayesian Elicitation with LLMs: Model Size Helps, Extra "Reasoning" Doesn't Always. 2026. https://arxiv.org/abs/2604.01896.

Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. 2023. https://arxiv.org/abs/2306.13063.

Designing, Not Checking, for Policy Robustness: An Example with Optimal Taxation. 2020. NBER Working Paper 28098. https://www.nber.org/system/files/working_papers/w28098/w28098.pdf.

Diamond, Peter A. 1998. "Optimal Income Taxation: An Example with a u-Shaped Pattern of Optimal Marginal Tax Rates." American Economic Review 88 (1): 83–95.

Generating with Confidence: Uncertainty Quantification for Black-Box Large Language Models. 2023. https://arxiv.org/abs/2305.19187.

Gruber, Jon, and Emmanuel Saez. 2002. "The Elasticity of Taxable Income: Evidence and Implications." Journal of Public Economics 84 (1): 1–32. https://doi.org/10.1016/S0047-2727(01)00085-8.

LLMs Are Overconfident: Evaluating Confidence Interval Calibration with FermiEval. 2025. https://arxiv.org/abs/2510.26995.

McClelland, Robert, and Shannon Mok. 2012. A Review of Recent Research on Labor Supply Elasticities. Congressional Budget Office Working Paper 2012-12. https://ecommons.cornell.edu/entities/publication/d31f4398-9d40-4960-9772-5fb279259a76.

Metaculus. 2026. Metaculus FAQ. https://www.metaculus.com/faq/.

PolicyEngine Team. 2024. Enhanced CPS 2024. PolicyEngine-US-Data enhanced microdata dataset. https://github.com/PolicyEngine/policyengine-us-data.

"Quantile-Parameterized Distributions for Expert Knowledge Elicitation." 2024. Decision Analysis. https://pubsonline.informs.org/doi/abs/10.1287/deca.2024.0219.

Saez, Emmanuel. 2001. "Using Elasticities to Derive Optimal Income Tax Rates." Review of Economic Studies 68 (1): 205–29.

Saez, Emmanuel, Joel Slemrod, and Seth H. Giertz. 2012. "The Elasticity of Taxable Income with Respect to Marginal Tax Rates: A Critical Review." Journal of Economic Literature 50 (1): 3–50. https://doi.org/10.1257/jel.50.1.3.