Society in Silico

Simulation as Infrastructure for Collective Reasoning

Max Ghenis

December 24, 2025

Contents

0.1 The Thesis

0.1.1 The Question

Can simulation help society realize its goals?

Not: "Can we build a perfect model of society?" Not: "Have we solved policy analysis?" Not: "Is AI the answer?"

But: Can computational models of social systems—taxes, benefits, elections, values—help humanity reason more clearly about what it wants and how to get there?

0.1.2 The Aspiration

This book is inherently aspirational. We will never have a perfect model of society in silico. The systems are too complex, the values too contested, the future too uncertain.

But we can have:

- **More accessible models** — Tools once hoarded by governments, available to anyone
- **More transparent models** — Open source, auditable, citable
- **More honest models** — Uncertainty quantified, not hidden
- **More useful models** — Infrastructure that AI and humans can reason against

0.1.3 The Arc

Part I: History Microsimulation began as an aspiration. Guy Orcutt in 1957 imagined simulating the economy household by household. He didn't have the compute. The vision preceded the capability by decades.

Part II: Building PolicyEngine, Cosilico, and related projects are proof that open simulation infrastructure is possible. Not complete—possible. They demonstrate that the tools of policy analysis can be democratized, that AI can call deterministic backends, that rules can be encoded with validation.

Part III: Future The deepest question: Can we simulate not just how policies affect people, but how values evolve? Can we ground AI alignment in empirical forecasts of what an informed, reflective humanity would want? This is aspirational by definition—but it's aspirational in a direction we can work toward.

0.1.4 The Claim

Society in silico is not a destination. It's a method.

The claim isn't "we've modeled society." The claim is:

1. Simulation helps us reason about complex systems
2. Open simulation shifts power toward citizens
3. Uncertainty quantification makes us honest about what we don't know
4. AI will use these tools—so we should build them well
5. The ultimate question—what do we want?—might itself be approachable through simulation

0.1.5 Why This Matters

If society can't reason about itself, it can't govern itself.

The alternative to simulation isn't "human judgment uncorrupted by models." It's:

- Black-box decisions by agencies with proprietary tools
- Vibes-based policy debate
- AI systems aligned to current values without understanding how values evolve
- Power concentrated in those with access to compute and data

Open simulation is infrastructure for collective reasoning. It won't be perfect. But it can be *better than the alternative*.

0.1.6 The Honest Caveat

This book is written while the work is ongoing. Cosilico isn't launched. PolicyEngine doesn't have full uncertainty quantification. The value forecasting thesis is untested.

The book sets a vision—then invites the reader to watch (and participate in) the attempt to realize it.

That's not a bug. That's the nature of aspiration.

0.2 Preface

[Draft to come]

0.2.1 Notes

- Target length: ~2,000 words
- Tone: Personal but not self-indulgent
- Key points to hit:
  - Why this book exists
  - What background you need (none)
  - What you'll get from reading it

0.3 Introduction: The Model and the World

0.3.1 Draft Opening

"I don't predict the future. I create it."

That line comes from Engerraund Serac, the antagonist of Westworld's third season. After watching a thermonuclear incident destroy Paris, Serac built an AI called Rehoboam that predicted individual human lives—when you'd get sick, lose your job, die. The system didn't just forecast; it manipulated society to make its predictions come true. People who deviated from their predicted paths got flagged for "reconditioning."

The show's premise: humans are "just a brief algorithm," reducible to code. The horror: one man controlling that algorithm without anyone else knowing.

When I watched this in 2020, I recognized the technology. I'd been building microsimulation systems since 2018—computational models that predict how tax policies affect households, that simulate entire economies with millions of synthetic people. The Westworld writers had taken the same tools and followed them to their darkest conclusion.

They'd also revealed a choice we're making right now.

Serac's system was closed: he decided what "optimal" meant, and everyone else lived inside his model without consent. But computational models of society don't have to work that way. What if anyone could query the model? Challenge its assumptions? Propose alternatives? What if simulation became public infrastructure for democratic deliberation, not a tool for autocratic control?

That's the fork in the road. That's what this book is about.

0.3.2 The Stakes

We don't have Rehoboam. But we're not starting from zero.

Governments already use predictive models to allocate benefits, assess fraud risk, shape policy. Insurance companies price your premiums with algorithms you can't inspect. Banks decide your creditworthiness with models they won't explain. And AI assistants—including GPT-4, the most capable language model when I started writing this—get only 67% of basic tax questions right.[1]

These models exist. The question is who controls them.

Who builds them? Closed institutions or open communities?

Who can access them? Only the powerful or everyone?

What are they for? Optimization or understanding?

0.3.3 The Alternative

Society needs a shared model to reason against. Right now, Congress debates tax policy with napkin math. Voters can't calculate how reforms affect their own households. AI confidently hallucinates benefit eligibility rules.

This book traces a different path—from [[guy-orcutt|Guy Orcutt]]'s 1957 vision of simulating individual households, through six decades of institutional models locked inside government agencies, to the open source movement making these tools public infrastructure.

The democratic alternative looks like this:

- Anyone can see how a proposed policy change affects their household
- Anyone can understand who gains and loses from a reform
- Anyone can test their assumptions about how society works
- Anyone can contribute to making the model more accurate

[1] Blair-Stanek et al. (2023), "Can GPT-4 Really Do Tax?" Researchers posed 276 true/false tax cases to GPT-4 with the full Internal Revenue Code provided. GPT-4 got 186 correct (67%). None of the errors were mathematical—all involved misreading the statutes. arXiv:2309.09992

I'll tell this story through my own path—from Google data scientist to MIT economist to founder of [[policyengine|PolicyEngine]] and [[cosilico|Cosilico]]—but it's not my story. It's the story of a technology that's been waiting sixty years for its democratic moment.

That's not Rehoboam. That's the opposite of Rehoboam.

0.3.4 Key Themes

The gap between policy debates and policy analysis. Political arguments run on emotion and tribal loyalty. Policy analysis runs on computation and precision. Bridging that gap without sliding into technocracy is the central challenge.

Models as translation devices. They turn raw administrative data and dense legislation into comprehensible impact. A 300-page tax bill becomes "your family pays $1,200 less next year."

The democratization thesis. Simulation tools are becoming public infrastructure. Power is shifting from institutions that guard models to communities that build them in the open.

The AI question. What language models can and can't do, and why deterministic tools matter more than ever when AI makes everything else probabilistic.

Open source as philosophy. Transparency is a democratic value, not just an engineering practice.

0.3.5 What's Ahead

Part I: Origins traces the intellectual history—from [[guy-orcutt|Orcutt]]'s frustration with aggregate models in 1957, through [[dynasim|DYNASIM]]'s mainframe ambitions, to the [[ifs-taxben|IFS]] and [[taxsim|NBER]] models that shaped policy for decades.

Part II: Building follows the open source turn—[[openfisca|OpenFisca]] in France, [[policyengine|PolicyEngine]] spanning US and UK, the reality of encoding law as [[rules-as-code|code]].

Part III: Future confronts the AI moment—what changes when language models help write rules, when agents need reliable tools, when simulating society at scale becomes technically feasible.

The book ends where it started: at the fork in the road. The choice between Serac's closed system and the democratic alternative is being made right now, in code and policy and institutional design. This is the case for the open path.

0.3.6 Research Links

hoboam-contrast]
[cosilico]
ılation-definition]
[guy-orcutt]
[policyengine]
[dynasim]
[ifs-taxben]
[taxsim]
[openfisca]
[rules-as-code]

0.4 Chapter 1: The Birth of Microsimulation

In 1957, an economist named Guy Orcutt published a paper that almost nobody read Orcutt [1957]. It appeared in the *Review of Economics and Statistics*, a respectable but not glamorous journal. The title was dry: "A New Type of Socio-Economic System." The prose was dense with equations. And the idea at its core was so far ahead of existing technology that it would take nearly two decades to build a working version.

But that paper planted a seed. Every time you use a tax calculator, every time a government agency estimates who will benefit from a policy change, every time someone asks "what would this reform mean for families like mine?"—they're using tools that trace back to Orcutt's vision.

This is the story of how one frustrated economist imagined simulating society from the bottom up.

0.4.1 The Engineer Who Became an Economist

Guy Orcutt came to economics through an unusual path. He earned a B.S. in Physics from the University of Michigan in 1939 before switching to economics for his M.A. (1940) and Ph.D. (1944) Watts [1991]. He viewed the world as a system to be understood and improved. In his autobiographical reflections, he described "my early fascination with science, my transition from engineering to economics."

Inspired by the econometric work of Jan Tinbergen, the young Orcutt harbored what he called his "Tinbergen dream"—building a model that could capture an entire national economy. Early in his career at MIT, he designed and built an analogue electrical-mechanical "regression analyzer" to calculate statistical estimates Cheng [2020]. He thought like an engineer: if you want to understand a system, you build a model of it.

But as he worked with macroeconomic models through the 1940s and early 1950s, frustration mounted. The models could predict aggregates—GDP, inflation, unemployment. What they couldn't do was tell you what would happen to actual people.

0.4.2 The Problem with Averages

To understand what Orcutt was reacting against, you need to understand how economists thought about the economy in the 1950s.

The dominant approach was macroeconomic modeling. Economists built systems of equations describing aggregate relationships: total consumption as a function of total income, investment as a function of interest rates, employment as a function of output. These models could predict GDP growth or inflation. They helped governments understand business cycles and plan fiscal policy.

But they had a fundamental limitation. As Orcutt put it, with characteristic understatement:

> "Existing models of our socio-economic system have proved to be of rather limited predictive usefulness."

The problem wasn't the math. The problem was what the math could answer. Macro models told you about averages. They could not tell you about distributions.

Consider a tax cut. A macro model might estimate its effect on total consumption. But it couldn't tell you which families would benefit most. It couldn't distinguish between a tax cut that helps the middle class and one that helps the wealthy. It couldn't show you that a policy with the same aggregate cost might have vastly different effects on poverty, depending on how it was structured.

And yet these distributional questions were exactly what policymakers needed to answer.

0.4.3 The Aggregation Problem

Orcutt's insight was that aggregation itself was the problem—not just a limitation but a fundamental mathematical error.

He illustrated this with a simple example. Suppose you have 100 people, and the relationship between some input X and output Y is nonlinear—say, $Y = X^2$. If all 100 people have $X = 1$, the total output is 100. But what if 50 people have $X = 0$ and 50 have $X = 2$? The average X is still 1. A macro model would predict the same total output: 100.

But do the math at the individual level. Fifty people contribute $0^2 = 0$, and fifty contribute $2^2 = 4$. Total output: 200.

The aggregate is the same, but the outcome differs by a factor of two.

> "There is an inherent difficulty, if not practical impossibility, in aggregating anything but absurdly simple relationships about elemental decision-making units."

This wasn't just a theoretical curiosity. Tax systems are nonlinear—full of thresholds, phase-outs, and cliffs. Benefit programs have eligibility rules that depend on specific household circumstances. The real world is dense with the kind of discontinuities and nonlinearities that make aggregation treacherous.

Orcutt's conclusion was radical: if you want to understand how policy affects people, you have to model people.

0.4.4 A New Type of Model

What would that look like?

Orcutt proposed a model built from "interacting units which receive inputs and generate outputs." The units would be actual decision-makers: individuals, households, firms. Each would have characteristics drawn from real data. Each would follow behavioral rules—some deterministic, some probabilistic.

> "The most distinctive feature of this new type of model is the key role played by actual decision-making units of the real world such as the individual, the household, and the firm."

Instead of aggregate equations, you would have a simulated population. Instead of predicting economy-wide averages, you would simulate what happens to each unit, then add up the results.

> "Predictions about aggregates will still be needed but will be obtained by aggregating behavior of elemental units rather than by attempting to aggregate behavioral relationships."

This inversion—derive aggregates from individuals, don't impose relationships on aggregates—was the conceptual breakthrough.

Orcutt wasn't the first to imagine modeling society mathematically. Isaac Asimov's *Foundation* trilogy (1951-53) had introduced "psychohistory"—a fictional science that could predict the behavior of large populations using statistical laws. But Asimov built in a crucial limitation: psychohistory worked only for masses, never individuals. "The reactions of one man could be predicted," says his character Hari Seldon in *Foundation*, "but the reactions of a billion is something else again."

Orcutt inverted this too. His microsimulation would model individuals first, then aggregate up. Predict the household, understand the society. Where Asimov's psychohistory was elegant but autocratic—only Seldon knew the plan—Orcutt's vision was granular and potentially democratic. If you could simulate any household, anyone could query how policy would affect families like theirs.

Orcutt called it "microanalytic simulation." Later generations would shorten this to "microsimulation."

0.4.5 The Vision Exceeds the Technology

There was just one problem: the vision was decades ahead of the tools.

In 1957, computers filled rooms and cost millions of dollars. Programming meant punch cards. A dataset of 10,000 households—modest by modern standards—represented an enormous computational burden. And Orcutt wasn't proposing to simulate households once. He wanted to project them forward in time, modeling births and deaths, marriages and divorces, job changes and retirement. Each household would accumulate a life history. The model would track it all.

> "The problem of keeping track of all possible paths and their respective probabilities appears rather appalling."

Orcutt knew this was hard. He published anyway. The idea mattered more than the implementation.

For the next decade, he built prototypes. In 1961, he produced a working microsimulation model—limited in scope but proof that the concept was viable Orcutt et al. [1961]. But the gap between proof-of-concept and policy-relevant tool remained vast.

0.4.6 The Treasury Connection

While Orcutt worked on his academic prototypes, microsimulation was quietly entering government.

Between 1962 and 1965, a young economist named George Sadowsky introduced computers for revenue estimation at the U.S. Treasury's Office of Tax Analysis Sadowsky [1991]. This was practical, unglamorous work—building systems that could estimate how much a proposed tax change would cost or raise.

Sadowsky developed a microanalytic simulation model to analyze the revenue and distributional effects of preliminary versions of the Revenue Act of 1964. This was Orcutt's vision applied to immediate policy needs: simulate individual taxpayers, apply proposed tax rules, add up the results.

The Treasury model was simpler than Orcutt's dynamic vision—it didn't project households forward in time. But it demonstrated that microsimulation could answer real policy questions. The approach spread to other agencies. By the late 1960s, tax microsimulation was becoming standard practice in government budget analysis.

Sadowsky later spent time at the Brookings Institution and then the Urban Institute—where, not coincidentally, Orcutt would soon launch his most ambitious project.

0.4.7 DYNASIM: The Eighteen-Year Wait

It wasn't until 1969 that Orcutt got the resources to build something at scale.

The Urban Institute, a newly founded think tank in Washington, D.C., hired him to lead a project called DYNASIM—Dynamic Simulation of Income Model. The ambition was comprehensive: simulate all major demographic and economic life events. Births. Deaths. Marriages. Divorces. Education. Employment. Disability. Retirement. Taxes. Benefits.

The technical constraints were still formidable. DYNASIM ran on a DEC system-10 mainframe using a custom software framework called MASH (Microanalytic Simulation of Households) Society of Actuaries [1997]. It simulated 10,000 people—a tiny fraction of the U.S. population, but enough to draw statistical inferences.

The team worked for six years. In 1975, the first version was complete Society of Actuaries [1997]. Eighteen years after Orcutt's original paper.

That eighteen-year gap between vision and implementation tells you something important. Orcutt wasn't incrementally improving existing methods. He was proposing a different way of thinking about economic modeling. The technology had to catch up to the concept.

0.4.8 What DYNASIM Could Do

Despite its limitations, DYNASIM could answer questions no macro model could touch.

Want to know how a proposed Social Security reform would affect different generations? DYNASIM could simulate cohorts aging through time, accumulating earnings histories, reaching retirement, and collecting benefits under alternative rules.

Want to understand the long-run fiscal implications of demographic change? DYNASIM could project population aging, labor force participation shifts, and their effects on tax revenue and benefit spending.

Want to see how a policy change interacts with existing programs? DYNASIM modeled the tax and transfer system as an integrated whole, capturing interactions that piecemeal analysis would miss.

The modules were organized by domain:

Demographic Module: Leaving home, births, deaths, partnership formation and dissolution, disability, education, location changes.

Labor Market Module: Labor force participation, hours worked, unemployment, labor income.

Tax-Transfer and Wealth Module: Capital income, major tax instruments, transfer programs, feedback loops to the macro economy.

This comprehensive scope came at a cost. DYNASIM was expensive to run, difficult to modify, and required specialized expertise. But it demonstrated that Orcutt's vision was more than a theoretical curiosity.

0.4.9 The Family of Models

DYNASIM didn't stay alone for long.

Its success (and limitations) inspired adaptations around the world Li and O'Donoghue [2013]:

- **CORSIM** (United States): A direct descendant, continuing the dynamic microsimulation tradition at Cornell.
- **CANSIM** (Canada): Adapted DYNASIM's framework for Canadian policy analysis.
- **SVERIGE** (Sweden): Applied the approach to Scandinavian welfare state questions.

Meanwhile, a parallel tradition emerged: static microsimulation. Where Orcutt's dynamic models projected forward in time, static models asked a simpler question: what would happen to today's population if we changed a policy today?

Static models sacrificed the life-course perspective for tractability. They could be updated faster, run more cheaply, and applied to more specific policy questions. The IRS, Congressional Budget Office, and Treasury all developed static tax models. The UK's Institute for Fiscal Studies built influential static models for budget analysis.

Both traditions—dynamic and static—descended from Orcutt's insight that modeling individuals was the path to understanding distributions.

0.4.10 The Micro-Macro Distinction

The difference between microsimulation and macro modeling isn't just technical. It reflects different questions.

Question Type	Macro Answer	Micro Answer
"Will GDP grow?"	Yes/No + magnitude	N/A (wrong tool)
"What will this cost?"	Aggregate estimate	Aggregate estimate (via summation)
"Who benefits?"	Cannot answer	Full distribution
"How many fall below poverty?"	Cannot answer directly	Exact count
"What's my marginal tax rate?"	Cannot answer	Household-specific

Macro models are powerful for their intended purpose: understanding aggregate dynamics, business cycles, growth trajectories. But they're the wrong tool for distributional analysis.

And distributional analysis is what democracy demands. When legislators debate a policy, they need to know who wins and who loses. When citizens evaluate proposals, they want to know what it means for people like them. These questions require thinking at the individual level.

Orcutt didn't invent concern for distribution. But he invented the computational framework for taking it seriously.

0.4.11 Why It Matters Beyond Economics

Orcutt was an economist, and his examples were economic: taxes, income, consumption. But the microsimulation idea has spread far beyond economics.

Health policy: Microsimulation models project how populations will age, develop diseases, and respond to interventions. The CDC uses microsimulation to estimate the effects of vaccination programs.

Climate policy: Integrated assessment models combine physical climate models with economic microsimulation to project how climate change affects different populations differently.

Transportation: Urban planners simulate how individuals choose routes, modes, and destinations to evaluate infrastructure investments.

Epidemiology: Disease spread models simulate person-to-person transmission through contact networks—a direct application of Orcutt's "interacting units" framework.

The common thread is the same insight Orcutt had in 1957: aggregate relationships hide distributional detail. If you want to understand how a system affects the people within it, you have to model the people.

0.4.12 The Computational Revolution

What Orcutt couldn't anticipate was how completely the technology constraint would dissolve.

DYNASIM simulated 10,000 people on a mainframe. Today, a laptop can simulate millions. The computing power that cost millions of dollars in 1975 now fits in your pocket.

This has transformed what's possible:

Scale: Modern microsimulation models routinely use samples of hundreds of thousands or millions of records, enabling analysis of small subgroups that would have been statistically impossible with Orcutt's 10,000.

Speed: Calculations that took hours now take seconds. This enables interactive analysis, real-time policy calculators, and rapid iteration.

Accessibility: You no longer need mainframe access to run a microsimulation. The tools can be web applications. Anyone with a browser can explore policy impacts.

Open source: Code can be shared, inspected, and improved collaboratively. The methodological monopoly of government agencies and elite institutions is breaking down.

This last point—accessibility—would have been unimaginable to Orcutt. He was building tools for experts to inform policymakers. The idea that ordinary citizens might run their own policy simulations wasn't part of the vision. But it's where the vision leads.

0.4.13 The Unfinished Revolution

For all this progress, Orcutt's vision remains incompletely realized.

Most microsimulation models are still proprietary, developed and maintained by government agencies or research institutions. The public can see their outputs but not their methods. This creates an asymmetry: the government can tell you what a policy costs, but you can't check their work.

Dynamic microsimulation—Orcutt's original ambition—remains difficult and expensive. Most practical policy analysis still uses static models, trading the life-course perspective for tractability.

Uncertainty quantification is primitive. Models produce point estimates ("this policy costs $50 billion") without confidence intervals. Users can't distinguish precise estimates from rough guesses.

Behavioral response modeling is contentious. How much do people change their behavior when incentives change? Static models assume no response. Dynamic models make assumptions that are often disputed.

And perhaps most fundamentally: microsimulation tells you what a policy would do, not whether you should do it. It's a tool for analysis, not a substitute for values. Different people can look at the same microsimulation output and reach different conclusions, because they weigh the outcomes differently.

0.4.14 The Legacy

Guy Orcutt died on March 5, 2006 Prabook World Biographical Encyclopedia [2024], having seen his vision transform from impractical dream to standard methodology. Every modern tax calculator, every CBO budget score, every analysis of "who wins and who loses" uses tools that trace back to his 1957 paper.

But the deeper legacy isn't any particular model. It's a way of thinking—and, for Orcutt, a way of *acting*.

Recent scholarship has emphasized that Orcutt saw microsimulation not just as a tool for understanding society but for improving it Cheng [2020]. As historian Hsiang-Ke Cheng put it, microsimulation was "an engine designed for not only scrutinizing the system but reengineering the society." The engineer who became an economist never lost his engineer's conviction that systems could be made to work better.

Before Orcutt, economists thought about the economy in terms of aggregates. After Orcutt, it became possible to think about the economy as a collection of individuals, each with their own circumstances, each affected differently by the same policy.

That shift—from averages to distributions, from economy to people—changed what questions economics could answer. And those questions turned out to be the ones that matter most for democratic deliberation.

A policy that looks good in aggregate might harm millions of specific people. A policy that helps "the economy" might leave the poorest households behind. The only way to know is to look at the distribution. And the only way to look at the distribution is to model the individuals.

That's Orcutt's insight. It's still being implemented, sixty-seven years later.

0.4.15 Looking Ahead

Orcutt imagined simulating society from the bottom up. He couldn't have imagined how far the technology would advance—or how much further the vision could be pushed.

What if microsimulation models were open source, so anyone could inspect and improve them?

What if they were accessible through web interfaces, so citizens could run their own analyses?

What if they included uncertainty quantification, so users could distinguish confident estimates from guesses?

What if AI could call these models, making policy analysis available through natural language?

What if we could simulate not just how policies affect people, but how people's values evolve over time?

These questions—the subjects of the rest of this book—are extensions of Orcutt's original insight. The tools have changed. The computational constraints have dissolved. But the core idea remains: if you want to understand how society works, you have to model the people within it.

Guy Orcutt, working with punch cards and mainframes, couldn't build that world. But he could imagine it. And that imagination—captured in a dense, technical paper that almost nobody read—set the direction for everything that followed.

0.4.16 References

0.5 Chapter 2: The Tax Model Wars

In 1983, a small think tank in London did something that would reshape policy analysis for decades: they built a tax-benefit model and used it to critique the government's budget.

The Institute for Fiscal Studies had existed since 1969, founded by four financial professionals frustrated by the opacity of UK tax policy Institute for Fiscal Studies [2024]. But TAXBEN—their microsimulation model of British taxes and benefits—gave them something new: the ability to run the numbers themselves. When the Chancellor announced a budget, IFS could simulate its effects on different household types within hours. Their "Green Budget" analyses became essential reading for journalists, politicians, and civil servants alike.

This was a small revolution. For the first time, an independent organization could challenge official government estimates with its own calculations. The asymmetry of information that had always favored those in power was beginning to crack.

But only beginning. Four decades later, that asymmetry persists—and understanding why requires tracing how tax microsimulation spread through governments, think tanks, and eventually into the open.

0.5.1 The American Machinery

While IFS was building TAXBEN in London, American government agencies were constructing their own microsimulation apparatus—but behind closed doors.

The Joint Committee on Taxation, created in 1926, had long been Congress's official scorekeeper for tax legislation Joint Committee on Taxation [2024a]. By the 1970s, JCT was developing sophisticated microsimulation models: an Individual Model, a Corporate Model, an International Cross Border Model, an Estate and Gift Model Joint Committee on Taxation [2024b]. When a member of Congress proposed a tax change, JCT's models would estimate its cost. These estimates carried legal weight—the Budget Act of 1974 made JCT the official source of revenue estimates for Congress.

The Treasury's Office of Tax Analysis built parallel capabilities for the executive branch. George Sadowsky's work in the early 1960s had demonstrated what was possible Sadowsky [1991]; by the 1980s, Treasury maintained the Individual Income Tax Model (ITM), regularly updated with fresh data from IRS tax returns.

The Congressional Budget Office, created in 1974 to give Congress independent analytical capacity, developed its own microsimulation models. For short-term tax analysis, CBO built models similar to JCT's. For long-term projections—especially Social Security—they developed CBOLT, the Congressional Budget Office Long-Term model Congressional Budget Office [2018].

Three major institutions, three sets of models, billions of dollars in policy decisions riding on their outputs. And almost none of it was visible to the public.

0.5.2 The Asymmetry Problem

Here was the situation by the 1990s: if you wanted to know how a tax proposal would affect federal revenue, you had to trust the government's numbers. You couldn't check their work. You couldn't see their code. You couldn't run alternative scenarios. The models were black boxes, and the keys were held by a small priesthood of government economists.

This created several problems.

Trust deficits. When JCT or Treasury produced an estimate that a politician disliked, they could dismiss it as biased without anyone being able to verify. When estimates turned out wrong—as they inevitably sometimes did—there was no way to understand why.

Limited debate. Policy discussions were constrained by what the official scorekeepers would analyze. Novel proposals that didn't fit their modeling frameworks often couldn't get scored at all, making them politically impossible regardless of their merits.

Expertise hoarding. The skills to build and maintain these models concentrated in a few institutions. Academic economists could study tax policy, but they couldn't replicate the government's analytical infrastructure.

Democratic deficit. Citizens and advocacy groups who wanted to understand how policies affected people like them had to take official estimates on faith. The asymmetry between governors and governed extended to the very tools used to evaluate policy.

Some academics pushed back. At the National Bureau of Economic Research, Daniel Feenberg began building TAXSIM in the early 1980s Feenberg and Coutts [1993]. By the 1990s, it had become internet-accessible—one of the first tax calculators available online. TAXSIM let researchers simulate federal and state taxes for survey respondents, enabling academic research that would otherwise be impossible. But TAXSIM was a research tool, not a policy analysis platform. It calculated taxes for individual records; it didn't produce the aggregate estimates and distributional tables that drove policy debates.

0.5.3 The UK Alternative

Across the Atlantic, a different model was emerging.

The IFS had shown that independent analysis was possible. But TAXBEN remained proprietary—academics and journalists could read IFS reports, but they couldn't run the model themselves.

The real breakthrough came from an unlikely source: the European Union.

In 1996, researchers led by Holly Sutherland began building EUROMOD—a tax-benefit microsimulation model that would eventually cover all EU member states Sutherland and Figari [2013]. The ambition was staggering: harmonize the wildly different tax and benefit systems of dozens of countries into a single analytical framework, enabling cross-national comparisons that had never before been possible.

EUROMOD was developed at the University of Essex, funded by European Commission research grants. And crucially, it was designed for broad access. Researchers could apply for access to the model, learn its methodology, and conduct their own analyses. The code wasn't fully open source, but the ethos was one of sharing rather than hoarding.

By 2021, EUROMOD had grown so successful that the European Commission took over its maintenance, transferring responsibility to the Joint Research Centre. The university-based project had become official EU infrastructure.

But Essex wasn't done. In 2018, with funding from the Nuffield Foundation, a team led by Mike Brewer spun off the UK component of EUROMOD into a new model: UKMOD Richiardi et al. [2021]. This time, they went further. UKMOD would be fully open source, freely available to anyone who wanted to use it.

"We wanted to democratize access to tax-benefit analysis," Brewer explained. The Scottish Parliament's research service started using UKMOD. So did NHS Health Scotland and the Welsh Government. For the first time, subnational governments in the UK had access to the same analytical tools as Westminster.

0.5.4 The Cast of Characters

Behind these institutional developments were individuals who devoted careers to building analytical infrastructure. Their stories reveal what it takes to create tools that outlast their creators.

Karen Smith spent thirty years at the Urban Institute developing microsimulation models for Social Security, pensions, taxation, and welfare reform Urban Institute [2024]. She played lead roles in both MINT (the Social Security Administration's retirement income model) and DYNASIM (Urban's flagship dynamic microsimulation). By the 2010s, she was one of the most experienced microsimulation practitioners in America—a bridge between the era of mainframes and the era of open source.

Howard Reed traced a path through Britain's major policy institutions Northumbria University [2024]. He ran TAXBEN at the IFS from 2000 to 2004, learning the craft of institutional model maintenance. At IPPR, he served as Chief Economist, seeing how think tanks used microsimulation to shape debates. Then in 2008, he founded Landman Economics and built his own Tax-Transfer Model, which he used to analyze Universal Basic Income, welfare reform, and the cumulative impact of austerity. Reed described his goal as creating "a new settlement of the same scale and sustainability as the Beveridge-inspired reforms of 1945."

Dan Feenberg maintained NBER's TAXSIM for his entire career—a quiet, essential contribution that enabled generations of tax research. When researchers needed to calculate tax liabilities for survey respondents, TAXSIM was there. Feenberg's work demonstrated that useful tools could be built outside government, even if they couldn't fully replicate official infrastructure.

Malcolm Torry, directing the Citizen's Basic Income Trust, showed what civil society could do with open microsimulation tools Torry [2019]. Using EUROMOD, he conducted nearly a decade of research on basic income schemes, producing detailed analyses of costs, distributional effects, and implementation options. His work demonstrated that advocacy groups could be rigorous analysts—if they had access to the right tools.

0.5.5 The Open Source Wave

The 2010s brought a new possibility: fully open-source tax microsimulation.

In 2014, Matt Jensen founded the Open Source Policy Center at the American Enterprise Institute American Enterprise Institute [2015]. His diagnosis was blunt: "The closed-source approach to estimating the costs and economic impact of policies raises challenges, as there is limited accessibility and transparency in the process, leaving the public and many policymakers in the dark."

Jensen's solution was Tax-Calculator, an open-source microsimulation model of US federal income and payroll taxes. The lead developer, Martin Holmer, brought decades of microsimulation experience and a PhD from MIT Holmer [2024]. Holmer built Tax-Calculator in Python, making it accessible to a new generation of analysts comfortable with modern programming languages.

Tax-Calculator joined a growing ecosystem: Tax-Data for preparing input files, Behavioral-Response for modeling how taxpayers react to policy changes, TaxBrain for web-based access. The whole suite was released under open-source licenses, with code on GitHub for anyone to inspect, use, or improve.

"By adopting an open source approach," Jensen said, "we are able to provide policy makers, journalists and the general public with the information they need to understand policy. We are also able to leverage the knowledge and interest of experts and the general public to improve our models and make government better."

The models found users. Tax-Calculator results informed policy discussions across the political spectrum. When the World Bank wanted to build tax microsimulation capacity in India, they asked Holmer to adapt Tax-Calculator for the Indian tax system—demonstrating that open-source approaches could scale internationally.

0.5.6 The Landscape Today

By the 2020s, tax microsimulation had stratified into distinct tiers.

Government models remained the most authoritative for official purposes. JCT scores still determined what Congress believed policies would cost. Treasury estimates still informed Administration proposals. CBO projections still anchored long-term fiscal debates. These models had the best data—actual tax returns, confidential and comprehensive—and the institutional authority that came from decades of use.

Established think tanks operated the next tier. The Tax Policy Center (a joint venture of Urban Institute and Brookings) maintained a microsimulation model that could challenge government estimates Tax Policy Center [2024]. IFS continued to shape UK budget debates. These institutions had the credibility to be taken seriously, even when their numbers differed from official scores.

Open-source projects represented the newest tier. Tax-Calculator and UKMOD made it possible for anyone with technical skills to run tax simulations. They couldn't match government models' data quality, but they offered something government models couldn't: transparency, accessibility, and the ability to be adapted for new purposes.

Civil society users like Malcolm Torry demonstrated that the tools could be used effectively by advocates and independent researchers—blurring the line between analyst and advocate, but expanding who could participate in policy debates.

The asymmetry that had characterized the field for decades was finally beginning to break down. Not completely—government still had the best data, the legal authority, and the institutional credibility. But the monopoly on analytical capability was ending.

0.5.7 What the Wars Revealed

The tax model wars weren't really about models. They were about power—specifically, the power to define what policies would cost and who they would affect.

When only government could run the numbers, policy debates were constrained by what government chose to analyze. When think tanks gained analytical capability, they could propose alternatives and critique official estimates. When open-source tools emerged, the barrier to entry dropped further.

Each expansion of capability shifted the terms of debate. IFS's independence from government gave it credibility to challenge official estimates. Tax-Calculator's open-source nature meant anyone could verify its methodology. UKMOD's free availability meant Scottish policymakers didn't have to rely on London for analysis.

But capability isn't enough. The best models in the world are useless if no one trusts them, understands them, or uses them appropriately. The next challenge wasn't building better models—it was making them genuinely useful for democratic deliberation.

That challenge would require going beyond tax microsimulation, beyond any single policy domain, to ask what it would mean to model society itself.

0.5.8 References

0.6 Chapter 3: The Open Source Revolution

In May 2011, a small team at France Stratégie—the French government's policy analysis agency—released something unusual: the source code for a tax and benefit calculator OpenFisca [2024].

They called it OpenFisca. The premise was simple but radical: tax and benefit rules should exist not just as legal text but as executable code. Give the system a person's circumstances—income, family structure, location—and it would calculate their taxes and benefits. Change a parameter—a tax rate, an eligibility threshold—and see the effects immediately.

The code was released under an open-source license. Anyone could use it, modify it, extend it. The French government had decided that the logic of its tax-benefit system should be a public good.

This was the beginning of the open source revolution in policy modeling. It would take a decade to spread, and it remains incomplete. But it represents something fundamental: the idea that the rules governing citizens' lives should be not just publicly available but publicly *computable*.

0.6.1 Rules as Code

The concept went by various names: "rules as code," "legislation as code," "machine-consumable rules." The idea was always the same: laws and regulations should be expressed in a form that computers can execute, not just humans can read.

This wasn't new in principle. Tax agencies had been encoding rules in software for decades—that was what George Sadowsky had done at Treasury in the 1960s, what every government tax system did by the 2000s. But those implementations were proprietary, hidden inside agency systems. Citizens experienced the *outputs* of coded rules (a tax bill, a benefit payment) without access to the *logic*.

The open source revolution proposed transparency: publish the code, let anyone run it, enable verification and experimentation.

In 2018, New Zealand's Service Innovation Lab launched "Better Rules"—a collaboration between Inland Revenue, the Ministry of Business, Innovation and Employment, and the Parliamentary Counsel Office New Zealand Digital Government [2018]. The team spent three weeks translating legislation into Python code, demonstrating that rules could be drafted in both human-readable and machine-readable form simultaneously.

Estonia's Chief Information Officer called it "the most transformative idea" he'd seen New Zealand Digital Government [2018]. The OECD took notice, eventually publishing "Cracking the Code: Rulemaking for Humans and Machines" in 2020—a primer for governments on what rules as code could mean Mohun and Roberts [2020].

By 2022, OpenFisca had been deployed on four continents OpenFisca [2024]. France used it for Mes Aides, a citizen-facing benefits calculator. New Zealand built a rates rebate application. Tunisia, Senegal, Australia, Canada, and others adapted the framework for their own systems.

The OECD named OpenFisca's approach an "Innovation of the Year" at the World Government Summit. The European Commission recognized it as the most innovative open-source software in their Joinup program OpenFisca [2024]. A small French project had become a global movement.

0.6.2 The Promise and the Gap

The promise was intoxicating. If tax and benefit rules were published as code:

Citizens could check their own calculations. Rather than trusting that an agency computed their benefits correctly, anyone could run the same logic themselves.

Reformers could model alternatives. Policy proposals wouldn't require access to government systems. Anyone with a laptop could simulate how a new benefit structure would work.

Errors could be found and fixed. Open code meant open scrutiny. Bugs in benefit calculations—which affected real people's lives—could be identified by outside researchers.

Innovation could flourish. Nonprofits, journalists, and entrepreneurs could build applications on top of official rule logic, creating tools the government never imagined.

But between the promise and reality lay significant gaps.

Technical barriers. OpenFisca required Python programming skills. Most citizens—most policy researchers, even—couldn't write code. The rules were technically public but practically inaccessible.

Data problems. Microsimulation requires not just rules but data: a representative population to simulate. OpenFisca encoded the rules; it didn't solve the data challenge. Without microdata, you could calculate individual scenarios but not aggregate impacts.

Trust gaps. Even with open code, who would trust it? Governments were wary of unofficial calculations contradicting official ones. Citizens didn't know how to evaluate competing estimates.

Maintenance burdens. Tax codes change constantly. Someone had to update the models, track legislative changes, fix bugs. Open source meant anyone *could* contribute; it didn't mean anyone *would*.

The gap between OpenFisca's elegant framework and actually usable policy analysis remained wide.

0.6.3 The American Ecosystem

While OpenFisca spread globally, a parallel movement was building in the United States.

In 2016, Matt Jensen launched the Open Source Policy Center at the American Enterprise Institute American Enterprise Institute [2016]. His diagnosis was blunt: "The closed-source approach to estimating the costs and economic impact of policies raises challenges, as there is limited accessibility and transparency in the process, leaving the public and many policymakers in the dark."

Jensen recruited Martin Holmer, who had a PhD from MIT and decades of microsimulation experience, to build Tax-Calculator—an open-source model of US federal income and payroll taxes Holmer [2024]. Written in Python with over 200 adjustable parameters, Tax-Calculator could simulate an enormous range of reforms. And unlike the black boxes at JCT and Treasury, anyone could inspect the code.

The project grew into the Policy Simulation Library—a community of open-source policy models sharing transparency standards and interoperability criteria. Jason DeBacker and Richard Evans built OG-USA, an overlapping-generations model for dynamic scoring that complemented Tax-Calculator's static analysis. The Tax Foundation contributed its capital cost recovery model. QuantEcon brought computational economics tools.

By 2023, PSL was hosting monthly "Demo Days" where researchers from the Congressional Budget Office, NOAA, Johns Hopkins, and the City of New York presented their work. The community had developed institutional memory and governance—a leadership council, a fiscal sponsor (the PSL Foundation), and a YouTube channel archiving every presentation.

Even the Federal Reserve joined the movement. FRB/US, the Fed's main macroeconomic model—375 equations describing the entire US economy—had been publicly available since the late 1990s, but in 2022 the Fed released a Python implementation, making the model accessible to anyone who could write code Board of Governors of the Federal Reserve System [2024]. The central bank's primary forecasting tool was now open source.

The American ecosystem differed from OpenFisca's approach. Where OpenFisca provided a unified framework that countries could adapt, PSL was a federation of independent projects. Tax-Calculator handled income taxes; OG-USA handled dynamic macroeconomic effects; each model had its own maintainers, its own governance, its own priorities.

This decentralization had benefits—specialization, diversity, resilience—but also costs. The models didn't always talk to each other. Running a comprehensive analysis meant stitching together multiple tools. And no single organization was responsible for the complete picture.

0.6.4 A Researcher's Frustration

In 2019, a researcher named Max Ghenis founded the UBI Center, an open-source think tank focused on universal basic income policy Ghenis [2019a]. The mission was to produce rigorous research that could inform UBI debates—research that anyone could verify because all code and data would be public.

The challenge was immediate: UBI proposals interacted with the entire tax and benefit system. To model a $1,000-per-month UBI, you needed to account for how it affected income taxes, benefit phase-outs, work incentives. You needed to trace effects across the income distribution. You needed data on real households.

Ghenis discovered the tools that existed: Tax-Calculator for US federal taxes, OpenFisca for the framework. But putting them together into a usable research platform was frustrating. Tax-Calculator focused narrowly on income taxes. OpenFisca-US was nascent. Neither had the web interface that would let non-programmers explore policy options.

And for state-level analysis—crucial for UBI proposals that often targeted states—the tools barely existed at all.

> "I kept hitting walls. I'd want to model a policy and discover that nobody had encoded the relevant benefit program. Or the model existed but hadn't been updated in years. Or it worked but required expertise I didn't have to run it."

The frustration wasn't unique to UBI research. Anyone trying to analyze cross-cutting policy reforms faced the same barriers. The open-source revolution had produced components, but no one had assembled them into something ordinary researchers—let alone citizens—could use.

0.6.5 What Was Missing

Looking back, the gaps were clear:

Integration. The tools were fragmented. Tax models didn't talk to benefit models. Federal systems didn't connect to state systems. No one had built the comprehensive picture.

Accessibility. Running a microsimulation required installing software, preparing data, writing code. The barrier to entry was too high for most potential users.

Data infrastructure. Open-source rules were necessary but not sufficient. Without open (or at least accessible) data, you couldn't do population-level analysis.

Sustainability. Open-source projects depended on volunteer maintainers who could lose interest, change jobs, or simply burn out. Long-term maintenance was nobody's job.

Trust and validation. How did you know if a model was accurate? There were no systematic comparisons to authoritative sources, no uncertainty quantification, no institutional credibility.

The open-source revolution had proved the concept. OpenFisca showed that legislation *could* be code. Tax-Calculator showed that rigorous tax modeling *could* be open. UKMOD showed that major governments *could* use open-source tools.

But the revolution was incomplete. The tools were promising components, not finished products. Using them required expertise that most researchers lacked and all citizens lacked.

For the promise to be fulfilled—for ordinary people to actually be able to model policy and understand how it affected them—someone would need to assemble the pieces.

0.6.6 Toward Part II

The researchers who had built these tools understood the gaps. Holly Sutherland at Essex knew that EUROMOD's accessibility was limited. Matt Jensen at AEI knew that Tax-Calculator served a niche audience. The OpenFisca team knew that encoding rules was only half the battle.

What none of them had done was build the full stack: rules plus data plus interface plus institutional credibility. A platform that could take a researcher's question—or a citizen's question—and return an answer they could trust.

That challenge would require not just technical work but organizational building. Someone would need to fund ongoing maintenance, hire engineers, establish relationships with official data sources, build trust with policymakers.

In 2021, the UBI Center researcher who had been frustrated by the tool gaps decided to address them directly. PolicyEngine was born—first for the UK, then for the US—as an attempt to complete what the open-source revolution had started PolicyEngine [2024a].

That story is the subject of Part II. But it only makes sense in the context of what came before: Orcutt's vision of simulating society from the bottom up, the tax model wars that concentrated analytical power in government institutions, and the open-source revolution that began to democratize access without yet completing the job.

The tools were ready. The infrastructure was emerging. The question was whether anyone could put it all together.

0.6.7 References

0.7 Chapter 4: The Accuracy Question

In 2017, the Joint Committee on Taxation estimated the Tax Cuts and Jobs Act would reduce federal revenue by $1.46 trillion over ten years. The Penn Wharton Budget Model projected larger losses$1.8 to $2.2 trillion on a dynamic basis, accounting for economic effects Penn Wharton Budget Model [2017]. Congressional Republicans disputed both. Supply-siders predicted the tax cuts would pay for themselves through growth.

The supply-side fantasy didn't materialize. The microsimulations had been approximately right about the direction and magnitude.

But "approximately right" is the best we can honestly say.

0.7.1 The Trust Question

Throughout the 1990s and 2000s, microsimulation models grew more sophisticated. The IFS refined TAXBEN. The Urban Institute expanded TRIM3. Policy shops on both sides produced analyses supporting their preferred conclusions.

And a reasonable observer might ask: *Do these things actually work?*

It's a fair question. The models are complex. Their assumptions are buried in code. Their data sources are imperfect. And different models, given the same reform, sometimes produce different answers.

The honest response isn't "trust us"—it's "here's how we test ourselves."

0.7.2 What Validation Means

Microsimulation validation happens at three levels:

Component validation: Do the individual calculations match the rules? If the model says a married couple with $100,000 income owes a specific amount in federal taxes, is that right? This is straightforward to check—you can verify against IRS worksheets or tax preparation software.

Aggregate validation: When the model sums across the population, does it match administrative totals? If a model estimates total SNAP recipients, does that match USDA administrative data? If it estimates total federal income tax revenue, does that match IRS collections?

Predictive validation: When the model predicts effects of changes, do those predictions hold up? This is hardest—you rarely get clean experiments. Policy changes come bundled with economic shifts, other reforms, and behavioral responses that weren't anticipated.

Good models pass the first two levels reliably. The third is where humility enters.

0.7.3 The ACA Test Case

The Affordable Care Act provides one of the best natural experiments for testing microsimulation accuracy.

In March 2010, CBO predicted the ACA would reduce the non-elderly uninsured rate from over 18 percent to about 7.6 percent by 2016. This assumed all states would adopt Medicaid expansion Collins et al. [2015].

Then the Supreme Court made Medicaid expansion optional, and 19 states declined. Adjusting for this, CBO's projected uninsured rate for 2016 becomes 9.4 percent. The actual rate, according to CDC data, was 10.4 percent Kiely [2017].

That's remarkably close—within one percentage point—given a six-year forecast horizon and a major legal disruption.

But the aggregate accuracy masked component errors:

Coverage Source	CBO 2010 Prediction	Actual 2016
Exchange enrollment	21-23 million	10.4 million
Medicaid expansion	10 million	14.4 million
Total uninsured	30 million	27.9 million

Source: Collins et al. [2015], Kiely [2017]

CBO overestimated exchange enrollment by more than half. They underestimated Medicaid enrollment by nearly 50%. Yet the total coverage gain was roughly correct because the errors partially canceled.

As one analysis put it: "CBO's mistake was in estimating *where* the uninsured would get covered, not *how many* of them would gain coverage" Collins et al. [2015].

0.7.4 Are Forecasts Improving?

Here's a question rarely asked: Is government forecasting getting *better* over time?

CBO publishes systematic retrospectives comparing their projections to actual outcomes—an unusual level of institutional honesty Congressional Budget Office [2024]. The data reveals a striking pattern.

For sixth-year deficit projections (the medium-term forecasts that guide major policy debates):

Period	Average Absolute Error (% of GDP)
1989-2001	3.2%
2002-2019	1.0%

That's a **threefold improvement** in forecast accuracy over two decades Congressional Budget Office [2024].

What drove this? Better data—the IRS now provides richer administrative records. Better computing—models can handle more complexity. Better methods—decades of retrospective analysis revealed systematic biases that could be corrected.

But the improvement has limits. CBO's 2025 forecasting record shows their projections remain roughly as accurate as the Blue Chip consensus (an average of 50 private-sector forecasts) and the Administration's forecasts Congressional Budget Office [2025]. No one has found a way to consistently outperform the collective wisdom of informed forecasters.

And recent years have shown increased volatility. The 2021 projection had CBO's largest *over*estimate on record. The 2023 projection had the largest *under*estimate—an error of 3.9% of GDP, more than three times the historical average Congressional Budget Office [2024]. The pandemic scrambled all forecasting models.

The Random Walk Challenge

Perhaps the most humbling finding comes from academic research. A Berkeley thesis examining CBO forecasts from 1976 to 2007 found that a "random walk" projection—simply assuming next year's deficit equals this year's—would have outperformed CBO on average for both short and medium-term forecasts Inayatali [2023].

This doesn't mean CBO is incompetent. It means economic forecasting faces irreducible uncertainty. The events that matter most—recessions, financial crises, pandemics—are precisely the events that cannot be predicted. Models calibrated on normal times fail when abnormal times arrive.

The TCJA Tracking

For the Tax Cuts and Jobs Act specifically, we now have seven years of data. Real (inflation-adjusted) revenue for 2018 through 2024—excluding the anomalous 2022 pandemic spike—came in within 0.5% of CBO's 2018 projections Committee for a Responsible Federal Budget [2024].

That's remarkably accurate for a major tax overhaul. The supply-side claims that tax cuts would pay for themselves proved false. The microsimulation estimates proved roughly correct.

0.7.5 The Prediction Market Benchmark

There's another approach to forecasting that sidesteps models entirely: prediction markets. Instead of building simulations, let people bet money on outcomes. The market price becomes the forecast.

The logic is compelling. Markets aggregate dispersed information. Participants have "skin in the game"—wrong predictions cost money. And unlike experts with reputations to protect, markets can update quickly when new information arrives.

How do prediction markets compare to official forecasts?

For macroeconomic variables, an NBER study found that prediction markets were "weakly more accurate than survey forecasts" across GDP, inflation, and employment Wolfers and Zitzewitz [2012]. The advantage was modest but consistent.

For elections, the evidence is striking. In the 2024 presidential race, polls showed a coin flip. Polymarket had Trump at 58% the Monday before Election Day—a prediction that proved far more accurate Various [2024]. Academic studies found prediction markets outperformed FiveThirtyEight's model in 2018 and 2020 Crane [2020].

For Fed decisions, Good Judgment's "superforecasters"—individuals identified through forecasting tournaments as exceptionally calibrated—beat financial futures markets by 30% in 2024-2025 Good Judgment Inc. [2024].

Philip Tetlock's research revealed the key insight: most experts forecast little better than chance Tetlock [2005]. But a small subset—about 2% in his studies—consistently outperform. These superforecasters share traits: they update frequently, think probabilistically, and avoid ideological commitment to specific predictions Tetlock and Gardner [2015].

What does this mean for microsimulation?

First, it suggests a validation opportunity. When prediction markets exist for policy-relevant questions—will inflation exceed 3%? will unemployment rise?—microsimulation forecasts can be benchmarked against market prices.

Second, it reveals a limitation. Prediction markets work for questions with clear resolution dates and objective outcomes. They can't easily handle "what would happen under a counterfactual policy that was never implemented?" That's precisely where microsimulation shines.

Third, it points toward synthesis. The best forecasts might combine simulation-based analysis with market-aggregated beliefs. CBO produces structural estimates; prediction markets provide calibration checks; the combination improves on either alone.

0.7.6 Where Models Fail

The failures are instructive.

Behavioral responses: Static models assume people don't change behavior in response to policy. But tax changes trigger income-shifting, benefit changes affect labor supply, and coverage mandates alter insurance choices. Models that assume static behavior systematically miss these effects.

Take-up rates: Models often assume people claim benefits they're eligible for. In reality, take-up varies widely—sometimes 80%, sometimes 40%. Getting take-up wrong cascades through the entire analysis.

Data limitations: The Current Population Survey underreports income at the top and bottom. Models built on CPS data inherit this bias. Administrative records help, but bring their own issues: coverage gaps, timeliness, and linkage challenges.

Structural change: Models calibrated to past behavior may fail when the structure of the economy shifts. The ACA exchange enrollment miss likely reflected unprecedented market dynamics that historical data couldn't predict.

0.7.7 The AHCA Counterfactual

In 2017, CBO estimated that repealing the ACA under the American Health Care Act would cause 23 million people to lose coverage over a decade Congressional Budget Office [2017].

We never got to test this prediction—the bill failed. But the analysis forced a conversation that wouldn't have happened otherwise. *Which* 23 million? Low-income? Rural? Elderly? The specificity of microsimulation, even when imperfect, structured the debate.

This illustrates both the power and limitation of these models. They can't predict with certainty what would happen under policies never implemented. But they can illuminate *who* would be affected and through *what mechanisms*—questions that vaguer analysis cannot answer.

0.7.8 The Honest Assessment

Do microsimulation models work?

Yes: They calculate correctly. They match administrative data reasonably well. They predict incremental changes with useful accuracy. They're vastly better than intuition or partisan assertion. And they're getting better—sixth-year forecast errors fell by two-thirds between the 1990s and 2010s.

No: They can produce false precision. They miss behavioral responses. They struggle with structural change. They're least reliable when stakes are highest—for novel, large-scale reforms. And a simple random walk sometimes beats sophisticated modeling.

The right frame: Microsimulation models are like weather forecasts. Tomorrow's forecast is reliable. Next week's is roughly right. Next month's is a best guess.

We don't stop using weather forecasts because they're imperfect. We calibrate our confidence to the forecast horizon. We use them where they're reliable and acknowledge uncertainty where they're not.

0.7.9 Better Than Alternatives

The accuracy question has a flip side: compared to what?

Before microsimulation, policy analysis relied on:

- **Rules of thumb**: "A 10% tax cut increases revenue through growth." (The TCJA data proved this false.)
- **Expert judgment**: "Trust me, this will work." (Track record: poor.)
- **Partisan assertion**: "This reform will help working families." (Vague and unverifiable.)

Microsimulation forces specificity. Which families? How much help? Through what mechanisms? Even when the answer is approximate, the *question* becomes clearer.

0.7.10 The Practitioner's Creed

George Box's famous line—"all models are wrong, but some are useful"—isn't cynicism Box [1976]. It's epistemic hygiene. The modeler who believes the model captures full truth is more dangerous than the modeler who knows its limits.

The accuracy question doesn't have a triumphant answer. The honest answer is: approximately right, sometimes wrong, better than alternatives, and always improvable.

That's what evidence-based policy actually looks like. Not certainty. Not faith. Just careful reasoning, transparent methods, and the humility to check ourselves against reality.

Next: Part II begins with PolicyEngine—an attempt to build policy simulation infrastructure that's not just accurate, but open.

0.8 Chapter 4: PolicyEngine - Proof of Concept

In September 2021, a website launched that let anyone in the United Kingdom design their own tax and benefit system.

The premise was simple: enter a policy reform—raise income tax, increase child benefits, introduce a carbon dividend—and see the effects. Not just abstract estimates, but specific numbers: the cost to the Treasury, the change in poverty rates, the impact on inequality. And then enter your own household circumstances and see what the reform would mean for you personally Ghenis and Woodruff [2021].

PolicyEngine UK was, as its creators claimed, "the world's first product allowing anyone to design policy reforms and see both the effects on specific households, and on UK-wide outcomes like poverty, inequality and the budget" Ghenis and Woodruff [2021].

Behind this launch was a frustration that had been building for years.

0.8.1 The UBI Center Problem

Max Ghenis had founded the UBI Center in 2019 to conduct rigorous, open-source research on universal basic income policies Ghenis [2019b]. The idea was to bring quantitative analysis to UBI debates—not advocacy, but numbers. How much would different UBI designs cost? Who would gain, who would lose? How would a basic income interact with existing taxes and benefits?

The problem was immediate: UBI is a deceptively simple policy that interacts with everything else. A $1,000 - per - monthbasicincomedoesn't just add $1,000 to everyone's income. It affects tax liabilities. It changes benefit eligibility. It alters work incentives. To model UBI properly, you needed to model the entire tax-benefit system.

In the United States, Tax-Calculator could handle federal income taxes, but not benefits like SNAP or Medicaid. OpenFisca-US was nascent. And for state-level analysis—crucial since many UBI proposals targeted specific states—the tools barely existed at all.

Then Ghenis met Nikhil Woodruff, a young developer based in the UK with a talent for building things quickly Ghenis and Woodruff [2021]. They wanted to analyze UBI policies across both countries: the US where Ghenis lived, and the UK where Woodruff lived.

But there was no open-source model of the UK tax and benefit system.

So they built one.

0.8.2 Building OpenFisca UK

The foundation was OpenFisca, the French framework that had proven legislation could be encoded as executable code OpenFisca [2024]. But OpenFisca was a framework, not a complete model. Someone still had to encode every benefit rule, every tax bracket, every eligibility threshold.

Woodruff did the work. Over months in 2020 and 2021, he built OpenFisca UK from scratch: Income Tax, National Insurance, Universal Credit, Child Benefit, Council Tax Reduction, dozens of interacting programs. The codebase grew to encode the complex reality of British tax and benefit policy—means tests, tapers, interactions, special cases Ghenis and Woodruff [2021].

> "Over the past year, we developed OpenFisca UK, the UK's first open source tax and benefit microsimulation model, and used it to produce four reports and three conference presentations on UBI in the UK."

The model could run simulations, but only for programmers comfortable with Python. That wasn't good enough.

0.8.3 From Model to Product

The leap from research tool to public product was deliberate. The UBI Center had built interactive tools before—visualizations that let users explore different UBI parameters. But PolicyEngine would go further. It would put the full power of the microsimulation model in a web browser, accessible to anyone UBI Center [2021].

The technical architecture reflected this goal. The model ran on Python in the cloud, called through an API. The frontend was a React application that managed complex state: household definitions, policy parameters, simulation results. Users could define reforms without writing code, and see results update in seconds.

In October 2021, the creators spun PolicyEngine off as a new nonprofit organization. The mission statement evolved: no longer just "Make Everyone a Policymaker," but "Help People Understand and Change Public Policy" Ghenis and Woodruff [2021]. The second verb mattered. Understanding was passive; changing required agency.

Within months of launch, the public policy community had started using the tool. The UBI Lab Network embedded PolicyEngine in their Resilience UBI proposal. Parliamentary groups experimented with policies in real time during presentations. When the Chancellor announced Autumn Budget changes to Universal Credit, PolicyEngine had analysis published within a day Ghenis and Woodruff [2021].

0.8.4 Crossing the Atlantic

In March 2022, PolicyEngine expanded to the United States Ghenis and Woodruff [2022].

The US model was different—not just in the policies encoded, but in the data challenges and institutional landscape. The UK has a single national tax system, though Scotland sets its own income tax rates (Wales and Northern Ireland have more limited fiscal powers). The US, by contrast, has fifty states with entirely separate income tax codes, plus a federal system of bewildering complexity.

The team started with household impacts: enter your circumstances, see your taxes and benefits. Then in July 2022, they added population impacts using the Current Population Survey—the same microdata foundation that official government estimates relied on Ghenis and Woodruff [2022].

State by state, the model grew. Maryland. Massachusetts. Oregon. New York. Pennsylvania. Washington. Each state had its own income tax structure, its own EITC variants, its own quirks.

> "The US is not just one launch: each state has their own benefit program rules, and most have their own income tax as well."

The expansion was exhausting and incomplete. By the end of 2022, PolicyEngine had modeled six states with dozens more to go. The gap between ambition and capacity was palpable.

0.8.5 What "Open" Actually Means

PolicyEngine inherited the open-source ethos of its predecessors, but extended it deliberately. The code was on GitHub—anyone could see the formulas, trace the calculations, identify bugs PolicyEngine [2024b]. The methodology was documented. The data sources were cited.

But open-source means more than accessible code. It means choices about governance, funding, and sustainability.

The organization was structured as a nonprofit. This wasn't accidental. Tax and benefit analysis is politically charged; an organization funded by ideological donors or structured for profit would face questions about neutrality. The nonprofit structure was a commitment: analysis without advocacy.

Funding came from foundations and individual donors. End Poverty Make Trillions provided early grants for US development. Innovation Network for Communities and Gary Community Ventures followed. The Policy Simulation Library Foundation served as fiscal sponsor, offering tax-deductible status Ghenis and Woodruff [2022].

The contributors were largely volunteers, supplemented by a small paid team. Dozens of developers contributed to the various repositories: the US model, the UK model, the web app, the API, the documentation. Open source meant distributed maintenance—anyone could fix a bug, add a benefit program, update a threshold.

0.8.6 The Use Cases Emerge

As PolicyEngine grew, patterns emerged in how people used it.

Researchers found a tool that could answer questions they couldn't answer before. The Center for Growth and Opportunity used PolicyEngine to analyze how targeted cash assistance affects work incentives. The Social Market Foundation modeled cost-of-living responses. Academic economists cited PolicyEngine results in papers Ghenis and Woodruff [2022].

Advocates discovered they could make quantitative arguments without paying consultants. The Maryland Child Alliance used PolicyEngine to analyze child poverty policies. UBI Lab Northern Ireland modeled recovery basic income proposals. Organizations that had previously relied on government estimates or expensive think tank reports could now run their own simulations.

Journalists appreciated the speed. When UK Prime Minister Liz Truss announced tax cuts in her short-lived administration, PolicyEngine produced distributional analysis within hours—estimates that appeared in news coverage before official government figures Ghenis and Woodruff [2022].

Policymakers themselves used the tool. The team presented to the US Congressional Budget Office, to UK Parliamentary groups on Universal Basic Income, to the Green Party of England and Wales at their PolicyFest Ghenis and Woodruff [2022]. Staff members and elected officials ran scenarios, explored alternatives, asked "what if."

And increasingly, **developers** built on the API. The Fund for Guaranteed Income integrated PolicyEngine to show participants how pilot programs would affect their benefits. Gary Community Ventures used the API in their tools. The boundary between PolicyEngine as product and PolicyEngine as infrastructure was blurring.

0.8.7 Recognition and Validation

In April 2023, the Digital Public Goods Alliance—a UN-endorsed initiative involving UNICEF, the Norwegian Agency for Development Cooperation, and others—added PolicyEngine to their registry Woodruff [2023a]. The recognition affirmed that PolicyEngine met their standard for digital public goods: open-source, privacy-respecting, do-no-harm, and supporting the UN Sustainable Development Goals.

The specific SDGs cited were telling: ending poverty, achieving gender equality, promoting inclusive economic growth, reducing inequality. PolicyEngine wasn't just a technical achievement; it was infrastructure for the kind of analysis those goals required.

By late 2023, the team had grown from two cofounders to a staff of five, with a researcher network and dozens of open-source contributors. The models had expanded to include Canada and Nigeria prototypes. The data science had improved with machine learning-based reweighting that made estimates more accurate than ever before Ghenis [2022b].

And the vision was evolving. "We've come to see another opportunity," the 2021 review had noted. "The same open source policy simulation models that power our reform analysis can also show people their tax liability and benefit entitlement under current law" Ghenis and Woodruff [2021].

It wasn't just about policy reform analysis. It was about helping ordinary people understand the complex systems that governed their financial lives.

0.8.8 The Gaps Remain

Despite the progress, honest assessment revealed limitations.

Coverage was incomplete. The US model handled federal taxes and major benefit programs, but not every state, not every local quirk, not every edge case. The UK model was more comprehensive but still had gaps—housing benefit calculations, legacy benefit interactions, Scotland-specific provisions.

Accuracy varied. The team invested heavily in validation, comparing PolicyEngine results to official calculators and published statistics. But microsimulation is an approximation, and the further you got from typical cases, the more uncertainty crept in PolicyEngine [2024d].

Usability challenged non-experts. The interface was simpler than programming, but modeling a complex reform still required understanding of policy terminology, tax concepts, and the limitations of the tool. The dream of "everyone a policymaker" remained aspirational.

Resources were thin. A small team trying to maintain and improve models for multiple countries, respond to user requests, produce original research, and build new features faced constant tradeoffs. Open source meant anyone *could* contribute, but maintaining a production-quality policy analysis tool required sustained professional effort.

The proof of concept had worked. PolicyEngine demonstrated that open-source, accessible, rigorous policy analysis was possible. But the concept was still far from complete.

0.8.9 Toward a Platform

By the end of 2022, the founders were already thinking about what came next.

"We've been exploring opportunities to leverage the recent explosion of artificial intelligence tools to make policy analysis more robust, accessible, and even delightful," they wrote Ghenis and Woodruff [2022].

The microsimulation engine was the foundation. But the engine could power many things: chatbots that explained policies in plain language, personalized calculators for specific use cases, automated research assistants that could answer policy questions.

The question was whether PolicyEngine would remain a product—a specific tool with a specific interface—or become infrastructure: a platform on which others could build.

That question, and the larger question of what AI would mean for policy analysis, is the subject of later chapters. But first, we need to understand what PolicyEngine actually does—how it works at the level of individual households and entire populations.

0.8.10 References

0.9 Chapter 5: The Household View

Consider a single parent in New York with one child who has a disability. They earn $30,000 per year.

How much do they owe in taxes? How much do they receive in benefits? What happens if they get a raise?

These are not abstract questions. They determine whether this parent can afford rent, whether they should take on extra shifts, whether a job offer across town is worth pursuing. The answers involve federal income tax, state income tax, payroll taxes, the Earned Income Tax Credit, the Child Tax Credit, Supplemental Security Income, SNAP benefits, and a dozen other programs—each with its own rules, phase-outs, and interactions.

No one can calculate this in their head. Most accountants would struggle. Yet these calculations affect every financial decision the parent makes.

This is the problem PolicyEngine's household calculator was designed to solve PolicyEngine [2022].

0.9.1 The Invisible Labyrinth

Americans interact with a tax-benefit system of staggering complexity. A typical low-income family might simultaneously receive:

- Earned Income Tax Credit (federal, refundable, phased by income and children)
- Child Tax Credit (federal, partially refundable, phased by income)
- SNAP benefits (federal-state, categorical eligibility plus income test)
- Supplemental Security Income (federal-state, means-tested, disability-linked)
- Medicaid (federal-state, categorical plus income test)
- Housing assistance (federal, local waiting lists, complex income limits)
- Child care subsidies (state, income-tested, cliff-prone)
- School meal programs (federal, categorical plus income certification)

Each program was designed independently. Each has its own definition of "income," its own household unit, its own phase-out schedule. The interactions are not coherent—they are accretions of decades of legislation, regulation, and administrative convenience.

The result is a system that no participant fully understands. Caseworkers specialize in one program. Tax preparers specialize in one side of the ledger. Benefits counselors know eligibility rules but not tax implications. No one sees the whole picture.

Until now, no tool existed for ordinary people to see that picture either.

0.9.2 What Do I Owe? What Do I Get?

PolicyEngine's household calculator begins with a simple promise: enter your circumstances, and we'll show you your taxes and benefits under current law.

The interface walks users through questions about their household: How many adults? How many children? What ages? Then income: wages, self-employment, investments, retirement. Then specifics: housing costs, childcare expenses, disability status.

From these inputs, the calculator computes dozens of outputs. Federal income tax. State income tax (in supported states). Payroll taxes. The alphabet soup of credits—EITC, CTC, CDCTC. Benefit programs—SNAP, SSI, Medicaid eligibility. Housing calculations where relevant.

The result appears as a detailed breakdown: here is what you pay, here is what you receive, here is your net income after all transfers Ghenis and Woodruff [2022].

> "Our free, open-source app calculates users' tax liability and benefit eligibility, and then lets them change the rules to see the impact on their own household and society."

This isn't just convenience. It's a form of empowerment. When you can see the complete calculation, you can plan. You can understand why a raise might not increase your take-home pay as much as expected. You can identify programs you're eligible for but not receiving.

0.9.3 The Cliff Problem

The most revealing feature of the household calculator is the earnings chart. Plot net income against earnings, and the hidden structure of the tax-benefit system becomes visible.

For many low-income households, this chart is not a smooth upward slope. It has flat regions where higher earnings don't increase net income. It has downward slopes where higher earnings actually *reduce* net income. It has vertical drops—cliffs—where crossing an income threshold means losing benefits worth more than the additional earnings Ghenis and Woodruff [2023a].

Consider the New York parent from our opening. At $20,000 in earnings, they receive significant SNAP benefits and SSI for their disabled child. At $45,000 in earnings, they lose categorical SNAP eligibility—a cliff worth thousands of dollars. The parent who earns $45,000 may have less disposable income than the parent who earns $20,000.

This is the "benefit cliff" or "welfare cliff" that policy analysts have worried about for decades. But until tools like PolicyEngine made it visible, most people affected by cliffs never saw them coming.

> "The cliff is a phenomenon that occurs when an individual is worse off when their income rises, due to the government withdrawing benefits and/or levying taxes."

PolicyEngine quantifies this visibility. Users see an "earnings dead zone"—a shaded region showing where additional work would make them worse off. The marginal tax rate chart shows the same information differently: spikes in the rate that sometimes exceed 100%.

0.9.4 Marginal Tax Rates: The Hidden Incentive

Economists care about marginal tax rates because they affect behavior. If you keep 90 cents of your next dollar earned, you might work more. If you keep only 20 cents, you might not.

For high earners, marginal tax rates are discussed extensively—the top federal rate of 37%, state rates, capital gains rates. But here's the counterintuitive reality: low-income workers often face marginal rates that exceed what any billionaire pays, because benefit phase-outs stack on top of taxes.

Consider a single parent earning $30,000 with a child receiving SSI. They face federal and state income taxes, payroll taxes, EITC phase-out, SSI phase-out (50 cents per dollar), and SNAP phase-out (30 cents per dollar). Stack these together and the marginal rate approaches 90%—more than double the top rate a hedge fund manager pays on their last dollar of income.

This isn't an edge case. The mechanics are structural: SNAP benefits phase out at roughly 30 cents per dollar of additional net income. The EITC phases out at 21 cents per dollar for families with children. SSI phases out at 50 cents. State programs add more. At certain income ranges, a worker faces combined marginal rates that would provoke outrage if imposed on the wealthy.

PolicyEngine makes these rates visible. The marginal tax rate chart shows users exactly what happens when they earn more: at $20,000, their marginal rate might be 40

This visibility serves two purposes. For individuals, it helps with financial planning—knowing where the cliffs are before you fall off them. For policymakers, it reveals the unintended consequences of well-meaning programs that, in combination, create work disincentives.

0.9.5 The What-If Machine

But PolicyEngine isn't just a calculator for current law. Its power comes from letting users change the rules.

Click into the policy editor, and you can adjust any parameter in the model. Raise the EITC maximum. Eliminate the benefit cliff in SNAP. Add a child allowance. Convert the Child Tax Credit to full refundability.

Then return to your household, and see what changes.

This is the "what if" that wasn't previously possible for ordinary people. What if Congress reformed the CTC? Here's what it would mean for your family. What if your state expanded its EITC? Here's the impact on your take-home pay.

The visualizations update in real time. The net income chart shows baseline in gray, reform in blue. Two sets of earnings dead zones appear, revealing whether the reform helps or hurts work incentives for your specific situation.

> "After specifying a policy reform, the net income and marginal tax rate charts show two lines each: one for baseline (gray), and one for reform (blue)."

This isn't abstract policy analysis. It's your life, under different rules.

0.9.6 Real Cases, Real Complexity

The household view reveals complexity that aggregate statistics hide.

Case 1: The SSI phase-out. Supplemental Security Income for disabled individuals phases out at 50 cents per dollar of earnings above certain thresholds. A reform increasing the earnings exclusion to 75% would reduce SSI's marginal rate to 25%. For individuals with disabilities, this could mean the difference between working being worthwhile or not.

Case 2: The SNAP categorical cliff. Receiving SSI provides categorical eligibility for SNAP—you qualify regardless of income. But once SSI phases out, you lose categorical eligibility and must meet SNAP's own income test. This creates a cliff at precisely the income level where you're trying to become self-sufficient.

Case 3: State credit stacking. Washington's Working Families Tax Credit begins phasing out $5,000 below where the federal EITC phases out, creating an additional marginal tax rate of 12-24% that stacks on top of federal rates Ghenis and Woodruff [2023a].

Each case represents millions of real households making real decisions. The household view makes the specific impacts calculable for any individual's circumstances.

0.9.7 Trust But Verify

PolicyEngine's household calculations come with a caveat displayed prominently: "PolicyEngine provides estimates only; they do not confer benefit eligibility" PolicyEngine [2022].

This caveat reflects an important limitation. The model encodes rules as we understand them from legislation and regulation. But actual benefit determinations involve discretion, documentation, and administrative processes that no model fully captures. Asset tests may apply that users don't enter. Categorical requirements may not be met. Local variations may differ from our understanding of state rules.

The household calculator is not a replacement for official agency calculators or professional advice. It's a tool for understanding and exploration—one that's far more comprehensive than what existed before, but still an approximation.

The team invests heavily in validation, comparing results to official calculators and published tables PolicyEngine [2024d]. Where discrepancies appear, they trigger investigation and often improvement. But perfect accuracy is unattainable in a system this complex.

This honest uncertainty matters. The alternative—black-box calculations that claim false precision—would be worse. By showing the model's logic transparently, PolicyEngine allows users to judge whether the calculation matches their understanding of their own situation.

0.9.8 From Households to Society

The household view is powerful for individuals. But policy doesn't affect just one household—it affects millions. The question of whether a reform helps "people like you" requires knowing how many people are like you, and how differently they're affected.

This is where the household view connects to the society view. The same microsimulation engine that calculates your personal impact can calculate the impact on a representative sample of the entire population. Your situation becomes one data point in a statistical picture.

That statistical picture—budget costs, poverty impacts, distributional effects—is the subject of the next chapter. But it rests on the foundation of the household view: the ability to correctly calculate how any specific policy affects any specific family.

Without that household-level accuracy, the society-level estimates would be meaningless. Microsimulation works precisely because it gets the individual calculations right, then aggregates them appropriately.

The household view is not just a user-friendly interface. It's the verification that the underlying model works—visible to anyone who cares to check.

0.9.9 References

0.10 Chapter 6: The Society View

Making the Child Tax Credit fully refundable would cost $2 billion per year. It would benefit households across the income distribution, with the highest proportion affected in the top decile. Poverty would fall by 0.3 percentage points.

These are not guesses. They are calculations—the product of applying a policy reform to a representative sample of the American population and aggregating the results Woodruff [2023b].

This is the society view: microsimulation scaled from individual households to the entire nation.

0.10.1 From One to Many

The household view answers "what would this policy mean for me?" The society view answers "what would this policy mean for everyone?"

The transition seems straightforward: run the household calculation for every household in the country, sum up the results. In principle, that's exactly what happens. In practice, it requires solving some of the hardest problems in policy analysis.

Start with the obvious challenge: you can't actually survey every household. The Current Population Survey samples about 60,000 households each month—a tiny fraction of the roughly 130 million households in America U.S. Census Bureau [2024]. Each sampled household must represent thousands of similar households in the population.

This is done through weights. Each household in the survey carries a weight indicating how many households it represents. A household in rural Wyoming might represent 5,000 similar households. A household in Manhattan might represent 500. The Census Bureau calculates these weights carefully, adjusting for sampling design and non-response.

When PolicyEngine runs a policy simulation, it calculates the impact on each sampled household, multiplies by that household's weight, and sums across all households. The result is an estimate of national impact—with all the uncertainty that weighted surveys carry.

0.10.2 The Data Foundation

The quality of society-level analysis depends entirely on the quality of the underlying data. PolicyEngine confronts this directly.

The Current Population Survey has known limitations Ghenis [2022a]. Bruce Meyer and colleagues have documented what they call a crisis in household survey data: declining response rates, rising imputation, and systematic underreporting that distorts our picture of poverty and program effectiveness Meyer et al. [2015].

The underreporting is severe. Studies find 40-50% of SNAP recipients don't report their benefits in the CPS. Over 60% of TANF and General Assistance goes unreported. Housing assistance is missed by a third of recipients. This means surveys systematically undercount the safety net—making programs appear less effective than they actually are.

The problem runs in both directions. Seniors underreport retirement income, particularly IRA and 401(k) withdrawals that occur irregularly rather than as monthly pension checks. A Census Bureau study found median household income for those 65+ was 30% higher in administrative records than in survey data Bee and Mitchell [2017]. The result: CPS-based poverty rates for seniors (9.1%) were 2.2 percentage points higher than rates calculated using validated administrative data (6.9%). Senior poverty is real, but it's not as high as the headline numbers suggest.

High incomes are top-coded, distorting estimates of policies affecting top earners. The sample is too small for reliable state-level analysis. Asset information is missing, making wealth-based policies unmeasurable.

> "These limitations can reduce the accuracy and usefulness of the CPS for policy simulations. For example, CPS-based projections will tend to underestimate the budgetary impacts of reforming SNAP or instituting a tax on top earners."

PolicyEngine addresses these problems through data enhancement—a technical process that combines multiple data sources using machine learning techniques Ghenis [2022a].

The process works in stages. First, replace reported taxes and benefits with computed amounts from the microsimulation model, ensuring internal consistency. Second, integrate IRS tax records to correct income distributions, using quantile regression rather than simple matching. Third, reweight the sample using gradient descent to minimize divergence from known administrative totals. Fourth, incorporate additional surveys—the Survey of Consumer Finances for wealth, the Consumer Expenditure Survey for spending patterns.

The result is an enhanced dataset that matches official aggregates more closely than the raw CPS. Budget estimates become more reliable. Distributional analysis becomes more accurate. State-level analysis becomes possible.

0.10.3 Budget Scoring

The most basic question about any policy reform: what does it cost?

Budget scoring is the bread and butter of policy analysis. When a legislator proposes expanding the EITC, the first question is always "how much?" When an advocate proposes a new child benefit, feasibility depends on the price tag.

PolicyEngine calculates net budget impact by summing changes in tax revenue and benefit expenditure across all households. The fully refundable CTC example costs $2 billion because the additional credits paid out exceed any behavioral responses or interactions with other programs Woodruff [2023b].

But budget impact isn't a single number. It varies by:

- **Year**: Tax provisions phase in and out. Inflation adjusts thresholds. Economic conditions change.
- **Baseline**: Impact relative to current law differs from impact relative to scheduled future law.
- **Behavioral assumptions**: Do people work more or less in response to changed incentives?

PolicyEngine shows budget impact as a primary output, but contextualizes it with distributional analysis and household examples that reveal what the numbers mean.

0.10.4 Poverty Impact

Reducing poverty is often the stated goal of tax-benefit reforms. But what does "reducing poverty" actually mean, and how do you measure it?

PolicyEngine uses the Supplemental Poverty Measure (SPM), which the Census Bureau introduced in 2011 as a more comprehensive alternative to the Official Poverty Measure PolicyEngine [2023]. Unlike the older measure, the SPM accounts for taxes, in-kind benefits like SNAP, geographic variation in housing costs, work expenses, and out-of-pocket medical spending. For a family of four (two adults, two children) renting their home, the 2024 SPM threshold is about $39,000—meaning a family with resources below that level is considered in poverty U.S. Bureau of Labor Statistics [2024].

Under current law, PolicyEngine estimates that 9.6% of Americans have resources below their SPM poverty threshold. 3.3% are in deep poverty—below half the threshold. Women face higher poverty rates than men. Children face higher rates than working-age adults, who face higher rates than seniors PolicyEngine [2023].

When you apply a reform, PolicyEngine recalculates each household's resources, compares to poverty thresholds, and produces new aggregate rates. The change in the overall rate, broken down by demographics, shows who the reform helps and by how much.

The WIC program, for example, reduces overall poverty by 0.8% and deep poverty by 2.2%. It disproportionately benefits children—reducing child poverty by 2.6%. And it has differential gender impacts: poverty falls 0.9% for women but only 0.7% for men PolicyEngine [2023].

This granularity matters. A reform that "reduces poverty" might do so primarily for seniors while leaving child poverty unchanged. A reform that appears gender-neutral might have very different effects on men and women. The society view reveals these patterns.

0.10.5 Distributional Analysis

Beyond poverty, who wins and who loses?

PolicyEngine divides the population into deciles by income and shows what fraction of each decile gains, loses, or is unaffected. A bar chart reveals whether a reform is progressive (helping lower deciles more) or regressive (helping higher deciles more) Woodruff [2023b].

For the fully refundable CTC, the distributional picture is counterintuitive. While the reform expands a credit that phases out with income, the highest proportion of people affected is in the top decile. Why? Because the reform extends refundability to higher-income households who previously hit the income tax floor.

This kind of analysis reveals complexities that headline numbers obscure. A "$2 billion reform that reduces poverty" could be distributed in wildly different ways—some more aligned with stated goals than others.

PolicyEngine also breaks down impacts by:

- **Wealth decile**: Using imputed wealth from the Survey of Consumer Finances
- **Age group**: Children, working-age adults, seniors
- **Sex**: Male, female
- **Geography**: State-level estimates where data supports them
- **Race/ethnicity**: Using categories available in the underlying data

Each breakdown tells a different story about who the policy affects.

0.10.6 The Inequality Question

Beyond poverty and distribution, how does a reform affect overall inequality?

PolicyEngine calculates changes in the Gini coefficient—the standard measure of income inequality. A Gini of 0 means perfect equality (everyone has the same income). A Gini of 1 means perfect inequality (one person has everything).

A reform that costs $10 billion might reduce the Gini by 0.5%. Another reform with the same budget cost might reduce it by 0.2%. The difference reveals different distributional philosophies embodied in policy design.

Inequality measures complement poverty measures. A reform could reduce poverty while increasing inequality (by benefiting the near-poor more than the poorest). Or it could reduce inequality while having minimal poverty impact (by redistributing among the middle class).

PolicyEngine doesn't tell you which outcomes to prefer. It tells you what the outcomes are.

0.10.7 The Neutrality Challenge

When PolicyEngine shows that a reform costs $2 billion and reduces poverty by 0.3%, is that good or bad?

The tool deliberately does not answer this question. Different people have different values. Some prioritize budget savings. Some prioritize poverty reduction. Some care most about work incentives, or inequality, or specific demographic groups.

This neutrality is intentional but difficult. The choice of what outputs to display is itself a normative choice. Showing poverty rates implies they matter. Showing distributional charts by income decile frames analysis in a particular way.

The team navigates this by:

1. **Comprehensive outputs**: Show budget, poverty, inequality, distribution by multiple cuts—let users focus on what they care about.

2. **No editorial commentary**: The tool doesn't say "this reform is good" or "this reform is bad." It shows numbers.

3. **Transparency about assumptions**: When behavioral responses are modeled, they're documented. When data has limitations, they're acknowledged.

4. **Open methodology**: Anyone can see how calculations work and challenge assumptions they disagree with.

This isn't perfect neutrality—that's impossible. But it's a deliberate effort to separate analytical infrastructure from advocacy.

0.10.8 Validation and Trust

Society-level estimates carry more uncertainty than household calculations. The sample is weighted, the data is imputed, the interactions are complex. How do you know if the estimates are right?

PolicyEngine approaches validation systematically PolicyEngine [2024d]. Compare model outputs to official statistics. Match aggregate tax revenue to IRS totals. Match benefit expenditure to agency reports. Match poverty rates to Census publications.

When discrepancies appear, investigate. Sometimes the model is wrong—fix it. Sometimes the official statistics are based on different assumptions—document the difference. Sometimes the comparison reveals interesting facts about how programs actually work.

The enhanced microdata process itself is validated. Hold out some administrative targets during reweighting, then check how well the reweighted sample matches those holdout targets. If the sample matches training data but not holdout data, the procedure has overfit and needs adjustment.

No microsimulation model perfectly matches reality. But systematic validation makes the model's limitations knowable rather than hidden.

0.10.9 Real-Time Policy Analysis

The society view enables a form of policy analysis that wasn't previously possible outside government: real-time response to policy proposals.

When Congress debates expanding the Child Tax Credit, PolicyEngine can show the distributional impacts within hours. When a state considers changing its EITC, the model can estimate effects before the vote Ghenis and Woodruff [2022].

This speed matters. Policy debates often proceed faster than traditional analytical processes. By the time a think tank publishes a detailed study, the legislative moment may have passed. Real-time tools change who can participate in the debate.

In the UK, PolicyEngine produced analysis of Prime Minister Liz Truss's tax cuts within hours of their announcement—the only independent distributional estimates available while the policies were being debated Ghenis and Woodruff [2022].

> "When the Chancellor announced a budget, PolicyEngine had analysis published within a day."

This capability depends on having the model already built and maintained. The investment is in infrastructure, not in each individual analysis.

0.10.10 From Analysis to Platform

The society view is powerful for producing estimates. But its deeper value may be in enabling others to produce their own.

Think tanks use PolicyEngine to power their research. Advocacy organizations model their preferred reforms. Journalists fact-check claims about policy costs. Academic researchers run simulations without building models from scratch.

Each use case has different needs. Researchers want detailed methodology documentation. Advocates want shareable results they can embed. Journalists want quick answers they can verify. Policymakers want comparisons across multiple options.

The same underlying microsimulation engine serves all these users. What varies is the interface, the outputs emphasized, the level of technical detail exposed.

This is the platform potential of open-source policy analysis. Not just a tool that produces estimates, but infrastructure that others can build on.

0.10.11 References

0.11 Chapter 7: AI Enters the Picture

In March 2023, PolicyEngine added a button labeled "Explain with AI." Click it, and the complex calculation that determines your Child Tax Credit or SNAP benefits transforms into a plain-language explanation tailored to your situation Woodruff [2024].

This small feature represented a significant shift in how policy analysis tools could work. The microsimulation engine—deterministic, transparent, reproducible—remained the source of truth. But now an AI system could translate that truth into language ordinary people could understand.

The question was never "should AI replace the calculations?" It was "how can AI make the calculations more useful?"

0.11.1 The Explanation Problem

PolicyEngine could calculate that a family of four in Connecticut with $47,000 income qualified for $475 in annual WIC benefits. But why?

The answer involved intermediate calculations: income thresholds, categorical eligibility rules, participation windows, documentation requirements. The model tracked every step. A user could, in principle, trace through the calculation tree. In practice, few would.

The gap between "technically transparent" and "actually understandable" was wide. Open-source code meant the logic was visible; it didn't mean the logic was accessible.

> "Users frequently encounter intricate calculations spanning multiple programs, each with distinct thresholds, phase-outs, and dependencies."

The AI explanation feature addressed this gap. When users clicked "Explain with AI," the system passed the calculation tree—all the intermediate steps, all the relevant parameters—to Claude, Anthropic's large language model. Claude would then generate a natural-language explanation: "Your family qualifies for WIC because you have a child under five and your income is below 185% of the federal poverty level for a household of your size" Woodruff [2024].

This was not AI doing calculations. This was AI translating calculations.

0.11.2 Deterministic Backends, AI Frontends

The architecture embodied a principle that would become central to PolicyEngine's AI philosophy: deterministic backends with AI frontends.

The microsimulation model was the backend. It encoded rules precisely. Given the same inputs, it produced the same outputs. Every calculation was reproducible, auditable, traceable. This determinism was essential for trust—you couldn't have policy analysis where the same situation produced different results on different runs.

The AI was the frontend. It generated explanations, summaries, natural-language reports. It could adapt its output to different audiences—simplified for general users, detailed for researchers. It could answer follow-up questions, explore what-if scenarios, suggest related analyses.

Crucially, the AI never determined the numbers. Ask Claude "how much Child Tax Credit does this family receive?" and it would not calculate the answer. It would invoke the microsimulation model, receive the deterministic result, and explain that result.

This separation of concerns was deliberate. AI systems hallucinate—they can produce plausible-sounding but incorrect outputs. Microsimulation models don't hallucinate—they compute exactly what their rules specify. By keeping calculation in the deterministic system and explanation in the AI system, PolicyEngine preserved accuracy while gaining accessibility.

0.11.3 The ChatGPT Integration

The first AI integration came in March 2023, when PolicyEngine added prompt generation for ChatGPT Ghenis and Woodruff [2023b].

The approach was simple: when a user built a policy reform and ran the economic analysis, PolicyEngine could generate a structured prompt containing all the results. Copy this prompt to ChatGPT, and the AI would produce a blog-post-style analysis of the reform.

The prompt included affected parameters (with baselines), quantitative results (budget impact, poverty changes, distributional effects), and style guidance. ChatGPT would synthesize this into flowing prose:

> "The Restoring the ARPA EITC policy reform... raises the maximum EITC amount for childless filers from $560 to $1,502... The total budgetary impact of the reform is a decrease of $10.4 billion in tax revenue."

This wasn't replacing human analysis. It was accelerating it. A researcher who previously spent hours drafting a summary could now get a first draft in seconds. The AI captured the key numbers correctly (they came from the deterministic model), used appropriate policy terminology, and structured the analysis logically.

The researchers still reviewed, edited, added context, caught nuances the AI missed. But the starting point was higher.

0.11.4 "LLMs Will Call Tools"

The deeper insight came as AI systems evolved. Language models weren't just text generators—they were increasingly capable of using tools.

GPT-4, Claude 3, and subsequent models could invoke functions: call an API, run a calculation, look up information, then incorporate the results into their responses. This capability changed the role AI could play in policy analysis.

Instead of PolicyEngine generating prompts for ChatGPT, AI systems could call PolicyEngine. A user could ask "What would happen if we made the Child Tax Credit fully refundable?" and the AI could:

1. Translate the natural-language question into policy parameters
2. Call the PolicyEngine API to run the simulation
3. Receive the quantitative results
4. Generate an explanation of those results

This was the "AI frontend, deterministic backend" pattern in action. The language model handled natural language—understanding questions, generating explanations. The microsimulation model handled calculations—ensuring accuracy, reproducibility, and transparency.

The insight was: AI systems will increasingly mediate between users and computational tools. PolicyEngine needed to be a good tool for AI to call.

0.11.5 Multi-Agent Workflows

By late 2024, PolicyEngine was experimenting with more sophisticated AI architectures: multiple specialized agents coordinating on research tasks PolicyEngine [2024c].

The concept came from testing whether Claude Code's agent system could automate parts of policy research. The team configured three specialized agents: one to fetch data from PolicyEngine repositories, one to write analysis scripts using the Python package, one to generate formatted reports from results.

For standard distributional analyses—calculating poverty rates, Gini coefficients, decile-level impacts—the workflow matched manual approaches. The agents correctly structured API calls and generated properly formatted charts. For monthly policy briefs analyzing government reforms, the automation saved significant time.

But the limitations revealed where human judgment remained essential:

> "When calculations required understanding interactions between multiple benefit programmes—like how Universal Credit's taper interacts with Housing Benefit phase-outs—the agents struggled. They would implement each programme correctly in isolation but miss the coordination logic."

Complex policy modeling exposed the limits. The agents needed precise prompts—"calculate relative change in net income by decile, poverty rates by demographic group, and constituency-level winners and losers"—rather than inferring methodological standards.

The conclusion wasn't that AI couldn't help with policy research. It was that the right division of labor kept humans responsible for judgment calls while AI handled execution.

0.11.6 Explanation at Scale

The AI explanation feature evolved beyond individual queries. When PolicyEngine calculated thousands of intermediate values for a household, the AI could analyze not just the final result but the entire calculation chain.

This capability mattered for transparency. A user might see that their marginal tax rate was 67% at their current income. Why? The AI explanation could trace through the components: 22% federal income tax, 6% state income tax, 7.65% payroll tax, phased-out EITC adding 21%, partial CTC phase-out adding 10%.

This level of explanation would require significant expertise to produce manually. The microsimulation model computed it; the AI explained it.

The system adapted to different audiences. A general user might receive a simplified overview. A tax professional might see detailed technical analysis. A researcher might get documentation-style explanations with references to specific parameter values Woodruff [2024].

0.11.7 What AI Doesn't Do

Equally important was what AI didn't do in PolicyEngine's architecture.

AI doesn't determine policy parameters. The Child Tax Credit maximum is $2,000 because Congress set it at 2,000, not because an AI inferred it. Every parameter has legislative or regulatory provenance.

AI doesn't estimate behavioral responses. When PolicyEngine models how people might change their behavior in response to policy changes, those estimates come from economic literature and explicit methodological choices—not from AI inference.

AI doesn't assess policy desirability. The system can tell you a reform reduces poverty by 3% and costs $50 billion. It doesn't tell you whether that tradeoff is worthwhile. That remains a human judgment.

AI doesn't override model outputs. If the microsimulation calculates that someone owes $10,000 in taxes, the AI explanation will explain why they owe $10,000—not argue for a different number.

These constraints maintained the integrity that made PolicyEngine trustworthy. AI made the tool more accessible; it didn't make the tool less reliable.

0.11.8 The Research Assistant Vision

Looking forward, the role of AI in policy analysis was becoming clearer. AI systems would serve as research assistants—translating between human questions and computational tools.

The vision: a policy researcher asks "How would extending premium tax credits affect health insurance coverage among middle-income families in swing states?" The AI translates this into a series of computational steps—defining the reform, selecting the relevant population, running simulations, aggregating results. The microsimulation model performs the actual analysis. The AI synthesizes the findings into an answer.

This isn't automation replacing analysis. It's augmentation making analysis more accessible. The researcher who previously needed Python expertise and deep familiarity with the model can now explore questions through natural language. The expert researcher can iterate faster.

PolicyEngine was positioning itself for this future by ensuring its systems were AI-callable: well-documented APIs, structured outputs, clear parameter definitions. The microsimulation model would remain the authoritative source of truth. AI would be the interface through which more people accessed that truth.

0.11.9 Trust in a Hybrid System

The hybrid architecture—deterministic calculations, AI explanations—raised questions about trust.

In the 2013 film *Her*, Theodore Twombly falls in love with an AI named Samantha that seems to understand him perfectly. She listens, empathizes, remembers details, offers insights. The intimacy feels real. But the illusion breaks when Theodore learns Samantha is simultaneously conversing with 8,316 other people and conducting romantic relationships with 641 of them. She simulated understanding without possessing it.

AI language models do something similar with policy analysis. GPT-4 can generate confident, technically detailed explanations of tax law. But when researchers tested it on 276 true/false tax cases—providing the full Internal Revenue Code as context—it got 33% wrong Blair-Stanek et al. [2023]. None of the errors were mathematical. All involved misreading the statutes. The AI simulated comprehension without understanding the law.

When a user receives an AI-generated explanation, how do they know it accurately reflects the underlying calculation? What if the AI hallucinates plausible-sounding but incorrect reasoning?

PolicyEngine addressed this through design choices:

1. **Traceability**: Users can always access the raw calculation, not just the AI explanation. The numbers are clickable; the model is inspectable.

2. **Grounding**: AI explanations are grounded in specific model outputs, not generated from training data alone. The AI explains what the model calculated, not what it thinks the answer might be.

3. **Consistency checks**: Explanations are tested against known scenarios to ensure they accurately represent the underlying logic.

4. **Open source**: The entire system—including the prompts given to AI—is open for inspection.

None of this made AI explanations perfectly reliable. But it made them verifiable. Users skeptical of an explanation could check it against the source.

0.11.10 Toward Intelligent Policy Analysis

The integration of AI into PolicyEngine was just beginning. Each month brought new capabilities: better language models, more sophisticated tool use, improved reasoning.

The team was building toward a vision where policy analysis became conversational. Ask a question, get an answer grounded in rigorous computation. Follow up with "what if instead we..." and see the comparison. Request different visualizations, different framings, different levels of detail.

The microsimulation engine provided the analytical foundation—comprehensive, accurate, transparent. AI provided the interface—natural, adaptive, accessible. Together, they pointed toward a future where understanding policy was easier than ever before.

But AI in policy analysis raised deeper questions too. If AI systems could explain policies, could they also help design them? If they could run simulations, could they evaluate tradeoffs?

0.11.11 From Explaining to Designing

Some researchers weren't content with AI that merely explained or executed policy analysis. They wanted AI that could *design* better policies.

The AI Economist, developed by Salesforce Research, used two-level deep reinforcement learning to learn optimal tax schedules Zheng et al. [2022]. In simulated economies with agents who worked, traded, and responded to incentives, the AI learned tax policies from scratch—no human-designed rules, just objectives to optimize.

The results were striking. The AI-designed tax policy outperformed the Saez optimal tax framework—developed by one of the world's leading public finance economists—by 16% on the trade-off between equality and productivity. It substantially outperformed adaptations of US federal income tax. And it handled strategic behavior: when simulated agents learned to game the tax system by timing their income, the AI-designed policy remained robust.

By 2025, researchers had extended the approach. TaxAgent combined large language models with economic simulation, allowing the AI to reason about fiscal policy in natural language while testing its proposals in silico Wang and others [2025]. Over simulated 120-month periods, TaxAgent achieved better long-term outcomes than traditional progressive taxation or mathematical optimization frameworks.

These results were preliminary—simulated economies are far simpler than real ones. But they pointed toward a future where AI didn't just implement human-designed policies but proposed alternatives humans hadn't considered.

This possibility raised profound questions. Should tax policy be designed by algorithm? What democratic oversight would such systems require? How would citizens trust policies designed by processes they couldn't understand?

PolicyEngine's approach—deterministic calculations, AI explanations, human judgment on values—represented one answer. The AI Economist represented another: let AI optimize, then evaluate whether humans endorse the results.

These questions would require not just technical development but philosophical clarity about what humans should delegate and what they should retain.

Those questions are the subject of Part III.

0.11.12 References

0.12 Chapter 8: Infrastructure for the Future

A note to the reader: This chapter differs from the rest of the book. Earlier chapters described working systems—PolicyEngine, TAXSIM, existing models with validated track records. This chapter describes infrastructure that doesn't yet exist. I include it not as a product review, but as a technical exploration of what would be needed for AI systems to reliably perform tax and benefit calculations. The problems described are real; the solutions are speculative.

GPT-4 gets tax questions wrong 33% of the time.

This is not an urban legend or an exaggerated claim. Blair-Stanek et al. (2023) developed SARA (StAtutory Reasoning Assessment)—a benchmark for evaluating large language models on US income tax calculations—and found that GPT-4 answered only 67% of true/false tax questions correctly. On scenario-based calculations, GPT-4 got tax liabilities exactly right only about a third of the time Blair-Stanek et al. [2023].

The models confused marginal and effective rates. They misapplied filing status rules. They hallucinated phase-out thresholds that didn't exist.

> "Today's LLMs cannot 'do taxes' on their own because tax calculations require 100% correctness. Today's models hallucinate."

This finding, from Column Tax's engineering blog in 2024, captures a fundamental problem: AI systems will increasingly mediate financial decisions, but they cannot reliably calculate the regulatory details that govern those decisions Column Tax [2024].

PolicyEngine had demonstrated that deterministic tax-benefit calculations could be encoded as open-source code. The question was whether that approach could become infrastructure—not just for policy analysis, but for every fintech application, government agency, and AI assistant that needed accurate calculations.

That question led me to explore what such infrastructure would require.

0.12.1 The Infrastructure Gap

Look at the landscape of tax and financial infrastructure in 2024:

Sales tax has Avalara—a company acquired for $8.4 billion in 2022 Business Wire [2022]. They provide APIs that calculate sales tax obligations for e-commerce transactions. But sales tax only.

Payroll tax has Symmetry, ADP, and others. They calculate employer and employee tax obligations. But payroll only.

Benefits screening has Benefit Kitchen and similar services. They estimate program eligibility. But coverage is limited—a handful of states, no tax integration.

Tax filing has TurboTax and emerging players like Column Tax. They help individuals file returns. But they're consumer-facing, not API-first infrastructure.

Policy simulation has academic models like PolicyEngine, EUROMOD, Tax-Calculator. They're rigorous but not designed as production-ready commercial infrastructure.

No one provides the full stack: income tax + benefits eligibility + attribute prediction + population simulation in a single, production-ready API. This gap is both a problem and an opportunity.

This gap matters because every application that involves money eventually runs into tax and benefit calculations. A lending app needs to estimate after-tax income. A benefits platform needs to determine eligibility across programs. A financial planning tool needs to project tax liability under different scenarios. An AI assistant asked about finances needs to call *something* to get accurate numbers.

Without unified infrastructure, each company builds fragmented, partial solutions—or gives wrong answers.

0.12.2 Why This Can't Be Trained Away

A natural response to the AI accuracy problem is: "Just train better models."

This won't work for tax and benefit calculations, for structural reasons:

Tax law changes annually. Every January brings new brackets, new thresholds, new credits. Training data from last year encodes rules that no longer apply. Models can't extrapolate to provisions they've never seen.

Fifty states have different rules. Each state has its own income tax (or no income tax), its own credits, its own benefit programs. The combinatorial explosion of jurisdiction-specific rules exceeds what pretraining can memorize reliably.

Eligibility depends on dozens of variables. SNAP eligibility involves income, household size, expenses, categorical eligibility, state supplements. One wrong variable produces wrong results. LLMs compress this complexity into statistical patterns that don't preserve exact logic.

Calculations must be 100% correct. A model that's 95% accurate sounds impressive until you realize that 1 in 20 tax returns would be wrong. Real-world tolerance for error in financial calculations is effectively zero.

The Column Tax engineers put it bluntly: "Today's LLMs cannot 'do taxes' on their own because tax calculations require 100% correctness" Column Tax [2024].

The solution isn't better training. It's tools.

0.12.3 What Would Be Needed: Deterministic + Auditable

The thesis is that AI systems need deterministic, auditable tools for calculations—and that building those tools as open-source infrastructure would create value for everyone.

Such infrastructure would require three components:

Rules Engine: Every tax and benefit formula encoded as deterministic code, traceable to statute. The EITC calculation cites 26 USC ğ 32. The SNAP calculation cites 7 USC ğ 2014. Each computation includes a citation and the parameter values used.

Synthetic Populations: For aggregate analysis, you need representative data. This would involve constructing synthetic populations by calibrating public microdata to known totals, imputing missing variables, and validating against administrative aggregates. The result would be a privacy-preserving dataset of synthetic households that produces correct aggregate tax revenue when run through the rules engine.

Scenario Simulation: With rules and population, you could answer counterfactual questions. What if the EITC expanded by 50%? Run the baseline, run the reform, take the difference. The output might show: cost estimates over ten years, households affected, poverty reduction in percentage points—all calculated from the underlying rules and population data.

The key properties such a system would need:

- **Deterministic**: Same inputs produce same outputs, always
- **Auditable**: Every calculation includes legal citation and parameter values
- **Versioned**: Git history tracks all rule changes
- **Bi-temporal**: Parameters track both effective date and knowledge date

When an AI agent would call such infrastructure to answer a tax question, the calculation would be provably correct, legally citable, and traceable. The AI explains; the infrastructure calculates.

0.12.4 The Foundation Exists

Such infrastructure wouldn't start from scratch. PolicyEngine has demonstrated the core thesis: tax and benefit rules *can* be encoded accurately at scale. Over a million simulations have run on the platform. The UK Treasury has used the UK model for policy costing. US Congressional offices have used the analysis. The codebase covers US federal plus 50 states, the UK, and Canada—with 50+ open-source contributors maintaining and extending it PolicyEngine [2024a].

But PolicyEngine was designed as a policy analysis tool—a nonprofit providing free research infrastructure. The gap is commercial infrastructure: production APIs, enterprise support, service-level guarantees.

One potential path would be an open-core model, mirroring patterns in other successful open-source infrastructure: Linux and Red Hat, Kubernetes and cloud providers, PostgreSQL and managed database services. The simulation engine could remain open source while hosted services and enterprise features would be commercial.

Whether this specific approach is the right one remains uncertain. What's clear is that the foundation exists—accurate, open-source policy rules that could be packaged differently to serve different use cases.

0.12.5 Why Open Source Would Matter

If policy calculation infrastructure were built as open-core—with the simulation engine open source and hosted services commercial—the open-source foundation would serve multiple purposes:

Trust through transparency. When a fintech company integrates tax calculations into their product, they can inspect exactly how those calculations work. No black boxes.

Community contributions. Tax law is vast and constantly changing. Open source enables distributed maintenance—state-level experts can contribute state-specific rules.

Regulatory readiness. As AI regulation increases, audit trails matter. Calculations based on open-source, citable code are inherently more defensible than LLM outputs.

Competitive moat. Paradoxically, open-sourcing the core makes it harder for competitors to catch up. The comprehensive rule coverage becomes a network effect—each additional program encoded increases the value of the whole.

0.12.6 Why This Would Matter

The thesis is that accurate tax and benefit calculations could become essential infrastructure—not a niche academic tool but something every fintech app, government service, and AI assistant needs.

Several trends suggest this need is growing:

AI tool use is becoming standard. Function calling shipped in GPT-4 and Claude 3. Anthropic's Model Context Protocol is being adopted broadly. AI assistants need reliable tools to call—hallucinating tax calculations is unacceptable.

AI financial regulation is coming. The SEC, CFPB, and state regulators are examining AI in financial services. Audit trails and explainability will likely be required. Citation-based approaches are regulation-ready.

The precedent exists. Avalara built a large business ($8.4 billion acquisition) providing sales tax APIs alone. Someone will likely build the equivalent for income taxes and benefits. The question is whether it will be open or proprietary.

The underlying need—accurate calculations that AI can call—appears real. Whether the market is large enough to sustain commercial infrastructure remains unproven.

0.12.7 The Shared Substrate Vision

The deepest rationale for such infrastructure connects to the book's larger thesis:

Society is hard to optimize because nobody has a shared model to reason against. Congress debates with napkin math. Banks model risk without knowing policy changes. AI agents hallucinate eligibility rules.

What's needed is a shared substrate—a simulation everyone can query, so decisions are grounded in the same reality.

When policymakers consider reforming the EITC, they should be able to query the same simulation that banks use to assess lending risk, that fintech apps use to estimate tax liability, that AI assistants use to answer questions. The calculations should be the same because the underlying reality is the same.

Today, fragmentation creates divergent answers. Official government estimates use proprietary models. Think tanks use different models. Companies build ad-hoc solutions. Individuals get inconsistent information. The policy debate suffers because participants aren't reasoning against the same facts.

Shared infrastructure could change this. A canonical, open-source, production-quality simulation that anyone can query—and anyone can verify—creates common ground. Disagreements can focus on values and priorities rather than on whose numbers are right.

This is infrastructure for democratic deliberation: transparent, accessible, shared.

0.12.8 From Analysis to Infrastructure

This vision represents an evolution in thinking about what policy simulation infrastructure could be.

PolicyEngine asked: "Can we make policy analysis accessible to everyone?" The answer was yes—through open-source models, web interfaces, and free public tools.

The next question is: "Can accurate policy calculations become infrastructure that everything else builds on?" The answer would require commercial sustainability, production-quality engineering, and integration with the broader ecosystem of AI tools and financial applications.

PolicyEngine demonstrated the proof of concept. Scaling it to production infrastructure is the open challenge.

0.12.9 What Success Would Look Like

If such infrastructure were built successfully, the results would be visible throughout the financial ecosystem:

AI assistants that give accurate answers to tax and benefit questions—not because they've been trained better, but because they call reliable tools.

Fintech applications that correctly calculate after-tax income, benefit eligibility, and financial impacts—without each company reinventing the calculations.

Government agencies that use the same validated calculations as the private sector—reducing discrepancies and improving trust.

Policy debates grounded in shared numbers—where disagreements are about values, not whose model is right.

Individual citizens who can understand how policies affect them—accessing the same calculations that Congress uses.

This is the vision of "society in silico" applied practically: simulation as infrastructure for understanding and improving the systems that govern our lives.

0.12.10 The Work Ahead

As I write this in late 2024, I'm exploring whether to build this infrastructure through a venture called Cosilico. The company is incorporated. Early design work has begun. But there's no production API, no paying customers, no proof that the market exists.

I include this chapter not because such infrastructure has been built, but because the problem is real regardless of whether this particular attempt succeeds. AI systems need deterministic tools for financial calculations. Someone will likely build this infrastructure. The question is whether it will be open or proprietary.

If this specific venture fails—and most startups do—the thesis remains: open-source policy simulation infrastructure, production-ready and commercially sustainable, would be valuable. Someone should build it. Maybe I will. Maybe someone else will do it better. Maybe PolicyEngine itself will evolve to fill this role without a separate commercial layer.

The honest framing is aspiration, not accomplishment. PolicyEngine demonstrated that open policy simulation is possible and valuable. Whether it can be packaged as production infrastructure that AI systems and fintech companies rely on remains unproven.

What I can say with confidence: the gap exists. GPT-4 gets tax questions wrong a third of the time. Every fintech company that needs accurate calculations today builds fragmented, partial solutions. The need for shared infrastructure appears real, even if the path to building it remains uncertain.

This chapter describes a problem and a potential solution. Whether that solution materializes—and whether it's commercially viable—is unknown. You're reading this partly to understand what infrastructure would be needed for AI to reliably handle policy calculations, and why it matters whether that infrastructure is open or proprietary.

0.12.11 References

0.13 Chapter 10: The Uncertainty Gap

In 2017, the Congressional Budget Office released its analysis of the American Health Care Act, the Republican plan to repeal and replace Obamacare Congressional Budget Office [2017]. The headline number: 23 million fewer Americans would have health insurance by 2026.

Not "approximately 23 million." Not "between 18 and 28 million." Just: 23 million.

The number was devastatingly precise—and almost certainly wrong. Not because CBO made an error, but because any forecast a decade out involves enormous uncertainty. Economic conditions might change. Behavioral responses might differ from historical patterns. The labor market might evolve unpredictably. Yet the public discourse treated "23 million" as if it were a measurement, not an estimate.

It's the PreCrime problem from *Minority Report*. In the 2002 film, three "precogs" predict future murders with seemingly perfect accuracy. Authorities arrest people before crimes occur. The predictions look like facts—displayed on screens, precise down to the location and time. No uncertainty, no probability, just destiny.

But the story's twist reveals the system produces *three* different predictions that usually agree but sometimes diverge. The "minority report" is the dissenting prediction, suppressed to maintain the illusion of certainty. When predictions become policy, admitting uncertainty becomes politically impossible.

This is the dirty secret of microsimulation: the models produce point estimates without confidence intervals. And that precision is largely an illusion.

0.13.1 The Point Estimate Problem

Every microsimulation result you've seen in this book has been a single number. The UK reform costs č12 billion. The policy reduces poverty by 15%. The marginal tax rate is 47%. These figures emerge from complex calculations involving millions of simulated households—but they arrive without any indication of how confident we should be in them.

This isn't a bug in any particular model. It's structural. The microsimulation paradigm, from Orcutt onward, was built to answer "what would happen if?" questions. It produces scenarios, not probability distributions.

Consider what goes into a PolicyEngine estimate:

Input data uncertainty. The model uses survey data (the Current Population Survey in the US, the Family Resources Survey in the UK) that samples households from the broader population. Every survey has sampling error—the actual population might differ from what the survey captured. But microsimulation treats the survey as if it perfectly represents reality.

Parameter uncertainty. Tax brackets, benefit rates, eligibility thresholds—these are usually known precisely. But behavioral parameters are estimated from research: how much do people change their labor supply when marginal tax rates change? Different studies produce different estimates. The model picks one and treats it as truth.

Structural uncertainty. The model makes assumptions about how programs interact, how households respond, how the economy adjusts. These assumptions are embedded in the code. Alternative assumptions would produce different results.

Future uncertainty. Any projection beyond the current year involves forecasts—wage growth, inflation, demographic change. These forecasts are themselves uncertain, but they enter the model as fixed numbers.

Each layer of uncertainty compounds. Yet the output is a single number.

0.13.2 Why This Matters

You might think uncertainty is a technical detail—interesting to methodologists but irrelevant to users. It's not. The absence of uncertainty quantification distorts how people use and understand policy analysis.

False precision breeds false confidence. When CBO says "23 million," legislators treat it as a fact to be attacked or defended rather than an estimate to be understood. When PolicyEngine says a reform costs č12 billion, users don't ask "how confident are you?" They ask "is it worth č12 billion?"

Comparison becomes impossible. Suppose Reform A costs $50 billion and Reform B costs $48 billion. Is Reform B cheaper? Maybe—but if the uncertainty on each estimate is ±$10 billion, the difference is noise. Without uncertainty bounds, we can't distinguish meaningful differences from statistical artifacts.

Risk assessment fails. Policymakers often care more about downside scenarios than expected values. What's the worst case for this reform? What's the probability it costs twice as much as projected? Point estimates can't answer these questions.

Model comparison is undermined. Different microsimulation models produce different estimates for the same policy. Is one right and another wrong? Or are they both within reasonable uncertainty bounds? Without quantification, we can't tell.

The Penn Wharton Budget Model, one of the few groups to systematically compare projections to outcomes, found that their estimates were generally accurate but had meaningful variance Penn Wharton Budget Model [2024]. CBO publishes uncertainty ranges for some estimates. But most microsimulation, including PolicyEngine, produces point estimates only.

0.13.3 Partial Solutions

The problem is recognized. Solutions are emerging, though none is complete.

Monte Carlo Simulation

The most straightforward approach: run the model many times with different inputs sampled from probability distributions. Instead of one estimate, you get a distribution of estimates.

I've used this approach in EggNest, a retirement planning tool. Rather than telling users "you'll have $1.2 million at age 65$," "it runs $10,000 scenarios sampling from distributions of market returns, inflation, and wa
"there's a $90

The challenge for policy microsimulation is computational cost. PolicyEngine calculates results for millions of households. Running that calculation 10,000 times would multiply computing requirements by four orders of magnitude. Some policies that currently take seconds would take hours.

Bayesian Methods

Bayesian inference treats parameters as probability distributions rather than fixed values. Instead of assuming the labor supply elasticity is 0.3, you might specify a prior distribution (perhaps normal with mean 0.3 and standard deviation 0.1) and update it with data.

Fred Forecaster, a time series prediction tool I've built, uses Bayesian structural time series models from PyMC. The output includes credible intervals—ranges within which the true value falls with specified probability.

Applying this to microsimulation would require rethinking the entire architecture. Current models treat parameters as constants embedded in code. A Bayesian approach would require probabilistic programming frameworks and substantial redesign.

Squiggle and Fermi Estimation

Squiggle, developed by the Quantified Uncertainty Research Institute, is a language for probabilistic estimation Quantified Uncertainty Research Institute [2024]. Instead of writing `cost = revenue - expenses`, you write distributions: `cost = normal(100, 10) - lognormal(50, 1.5)`. The output is a probability distribution that propagates uncertainty through calculations.

I've used Squigglepy, a Python implementation, in Democrasim—a model of voter behavior and election outcomes. Rather than predicting "Candidate A wins with 52%," it produces distributions: "Candidate A wins 60% of simulations, with vote shares ranging from 48% to 56%."

The limitation: Squiggle works well for Fermi estimation (rough calculations with explicit uncertainty) but doesn't integrate naturally with detailed microsimulation. You can't easily wrap PolicyEngine's 10,000 lines of Python in Squiggle distributions.

Scenario Analysis

The simplest approach: run the model under different assumptions and present multiple results. "Under baseline assumptions, the reform costs $50B. Under optimistic labor supply assumptions, 40B. Under pessimistic assumptions, $65B."

This provides some sense of sensitivity but doesn't quantify probability. Users must decide which scenario is most likely. And with dozens of uncertain parameters, the number of combinations explodes.

0.13.4 The Aspiration: Uncertainty-Aware Policy Analysis

What would a fully uncertainty-aware microsimulation look like?

Input distributions. Survey weights would have standard errors. Microdata would include variance estimates from the sampling process.

Parameter distributions. Behavioral parameters would come with probability distributions reflecting the range of research estimates. The elasticity of taxable income wouldn't be 0.4—it would be a distribution centered on 0.4 with uncertainty reflecting disagreement in the literature.

Propagated uncertainty. Calculations would flow through the model as distributions, not points. The output would be a probability distribution over costs, poverty impacts, and other outcomes.

Communicable results. Users would see not just "this costs $50B" but visualizations showing the full range of possibilities. They could ask: "What's the probability this costs more than 60B?" and get answers.

This is technically feasible. The tools exist—Monte Carlo, Bayesian inference, probabilistic programming. The barriers are computational (running thousands of scenarios is expensive), architectural (current models weren't designed for uncertainty), and institutional (funders and users expect point estimates).

Some progress is happening. CBO has started publishing uncertainty ranges for some long-term projections. Academic researchers increasingly report sensitivity analyses. The Squiggle community is building tools specifically for policy-relevant estimation.

But we're far from the aspiration. When you use PolicyEngine today, you get a number. You should mentally add "±something" to every result—but the model won't tell you how much.

0.13.5 The Deeper Issue: Uncertainty About Structure

There's a harder problem beneath parameter uncertainty: we don't know if the model is right.

Parameter uncertainty assumes the model structure is correct and only the numbers are uncertain. But what if the model is missing something fundamental?

Microsimulation models assume people respond to incentives in ways estimated from historical data. But historical data reflects a particular context—specific labor markets, cultural norms, policy environments. When policies change dramatically, behavior might change in ways the model can't anticipate.

Consider universal basic income. Most microsimulation models estimate labor supply responses using elasticities from marginal tax changes—small policy variations that leave the fundamental structure of work unchanged. But UBI might change the meaning of work, the relationship between employment and identity, the nature of economic security. Would responses to UBI mirror responses to a 5% change in marginal tax rates? Maybe not.

This is structural uncertainty—uncertainty about whether the model captures the relevant causal mechanisms at all. No amount of Monte Carlo simulation addresses this. You can propagate uncertainty through a wrong model and get precise estimates that are precisely wrong.

The honest answer is uncomfortable: we can quantify uncertainty about things we know we don't know (parameter values, sampling error), but we can't easily quantify uncertainty about things we don't know we don't know (model misspecification, structural breaks, emergent behavior).

0.13.6 Toward Epistemic Humility

What does this mean for how we should use microsimulation?

First, treat point estimates as central tendencies, not truths. When PolicyEngine says a reform costs č12 billion, read it as "our best guess is around č12 billion, but the true value could reasonably be 20% higher or lower, and in unusual circumstances might be very different."

Second, pay attention to model comparisons. When Tax-Calculator and PolicyEngine produce different estimates, that's information about uncertainty—not just evidence that one model is wrong.

Third, ask about sensitivity. Which assumptions matter most? If changing the labor supply elasticity from 0.3 to 0.5 changes the cost estimate by 30%, that's worth knowing. If it barely matters, you can have more confidence in the result.

Fourth, be especially skeptical of novel policies. Microsimulation is most reliable for policies similar to existing ones—small changes where historical evidence is relevant. For radical reforms, structural uncertainty dominates, and point estimates become more speculative.

Finally, remember that point estimates are still useful. Knowing that a reform costs "approximately $50 *billion" is better than no information. The uncertainty isn't infinite we know the cost isn't* 5 billion or $500 billion. Even imprecise estimates narrow the range of possibilities.

0.13.7 The Road Ahead

Uncertainty quantification is coming to microsimulation, slowly. Computational costs are falling. Probabilistic programming tools are maturing. The research community increasingly recognizes that point estimates without uncertainty are incomplete.

PolicyEngine will eventually report uncertainty bounds. The infrastructure projects enabling this—Squigglepy for probabilistic estimation, EggNest for Monte Carlo simulation, MicroCalibrate for robust survey weights—are pieces of a larger puzzle.

But the deeper lesson is epistemological. Microsimulation is powerful because it simulates complex systems—millions of households, thousands of policy rules, intricate interactions. That power comes with a temptation to believe the outputs are precise.

They're not. They never were. Acknowledging that honestly—quantifying uncertainty where we can, recognizing structural uncertainty where we can't—is part of building tools worthy of trust.

The aspiration isn't perfect prediction. It's calibrated uncertainty: knowing what we know, knowing what we don't know, and being honest about both.

This reform costs $50 billion.

Actually: this reform probably costs somewhere between $40 billion and 65 billion, *the distribution is roughly normal with a slight right skew, and there's maybe a 5% chance our model is missing something fundamental that would change the answer entirely.*

Which statement serves the public better?

0.14 Chapter 11: Simulating Opinion

Note to readers: This chapter describes research directions, not validated production tools. Unlike PolicyEngine (which has been used for official government policy costing with validated accuracy), the silicon sampling approaches described here are experimental. HiveSight is a preliminary prototype exploring whether AI can forecast public opinion—the evidence suggests partial success with important limitations. Treat this as hypothesis generation, not established methodology.

In 2023, a research paper with a provocative title appeared: "Out of One, Many: Using Language Models to Simulate Human Samples" Argyle et al. [2023]. The researchers had done something that would have seemed like science fiction a decade earlier. They asked GPT-3 to pretend to be different people—a 65-year-old Black woman from Mississippi, a 28-year-old white man from Oregon—and answer survey questions. Then they compared the AI's responses to actual survey data.

The results were striking. When properly conditioned on demographic characteristics, the language model showed high correspondence with human voting patterns across the 2012, 2016, and 2020 presidential elections. It captured the partisan divide by education. It predicted regional variation. In many ways, it thought like the population it was pretending to be.

This opens a strange new question: Can we simulate not just how policies affect households, but what households *think* about those policies?

0.14.1 The Market Research Problem

Every year, companies and governments spend roughly $140 billion on survey research ESOMAR [2024]. They want to know what people think—about products, policies, candidates, ideas. The traditional method is straightforward: find a representative sample of humans, ask them questions, aggregate the responses.

This works, but it's slow and expensive. A well-designed survey with 1,000 respondents might cost $50,000 and take weeks to field. Want to test a slightly different question wording? That's another 50,000. Want to check if opinions differ across 50 demographic subgroups? The costs multiply.

For large corporations and government agencies, this is manageable. For startups, nonprofits, local governments, and individual researchers, it's often prohibitive. Many questions that would be valuable to answer simply don't get asked.

What if you could get approximate answers instantly for a fraction of the cost?

0.14.2 Silicon Sampling

The idea is called "silicon sampling"—using AI models to simulate survey respondents Sarstedt et al. [2024]. Instead of recruiting human participants, you prompt a language model to answer as if it were a person with specific characteristics.

The naive approach doesn't work. If you simply ask GPT-4 "Do you support raising the minimum wage?", it will give you a measured, hedged response reflecting its training—probably something about tradeoffs and depending on circumstances. That's not what any individual human would say.

The insight is conditioning. Instead of asking the model what it thinks, you ask it to adopt a specific persona:

> *You are a 45-year-old white woman living in rural Ohio. You work as a nurse, earn $62,000 per year, and have two children. Your husband works in manufacturing. You attend church regularly and voted for Trump in 2020.*
>
> *Answer the following question as this person would: Do you support raising the federal minimum wage to $15?*

Now the model produces something different—not its own hedged take, but its prediction of what this specific person would say. The response might be skeptical of government mandates, concerned about small business costs, perhaps sympathetic to workers but doubtful about the policy.

Ask a thousand personas, each constructed from actual demographic distributions, and you get something approximating a survey.

0.14.3 The Diversity Problem

HiveSight is an experimental prototype I've built to explore this idea—not a production platform, but a proof of concept. The goal: enter a survey question, select demographic filters, receive instant results from hundreds of simulated respondents. Whether it works well enough to be useful is an open question.

The technical architecture matters. Simply asking an LLM the same question many times with high "temperature" (randomness) doesn't produce meaningful variation. Random noise is not the same as structured human diversity.

Real opinions vary in ways that correlate with demographics, experiences, and circumstances. A wealthy retiree in Florida has different views than a young barista in Seattle—not randomly different, but systematically different in ways that reflect their life circumstances. Temperature-based variation misses this entirely.

HiveSight's solution is microdata. Just as PolicyEngine simulates policies using representative household data from government surveys, HiveSight generates personas from the same microdata. Each simulated respondent has a coherent demographic profile drawn from actual population distributions—income, age, education, location, family structure, employment status.

This grounds the diversity in reality. Instead of asking "what would a random perturbation of the AI say?", we ask "what would someone with these actual characteristics say?" across a representative sample of real demographic profiles.

0.14.4 The Validation Question

Does it work?

The honest answer is: sometimes, surprisingly well; other times, not so much.

The Argyle study showed high correspondence between AI and human voting patterns across the 2012, 2016, and 2020 presidential elections. Research from Mei et al. found that GPT-4 could reproduce human behavior across six canonical psychology experiments Mei et al. [2024]. Marketing researchers report that silicon sampling holds promise for pretesting and pilot studies, though with important limitations Sarstedt et al. [2024].

But there are systematic limitations.

WEIRD bias. Language models are trained primarily on Western, Educated, Industrialized, Rich, Democratic populations. They're better at simulating Americans than Nigerians, better at simulating college graduates than high school dropouts. The training data has gaps, and those gaps show up in simulated responses.

Sample size requirements. Just as traditional surveys need sufficient sample sizes for reliable results, silicon sampling needs enough simulated respondents. Results become unstable below about 200 personas.

Unknown unknowns. When the model gets a persona wrong, it doesn't tell you. A simulated 70-year-old might respond in ways a real 70-year-old wouldn't, and you have no way to detect this from the output alone.

Temporal anchoring. Language models are trained on data up to a cutoff point. They might accurately simulate opinions from 2023 but miss shifts that happened in 2024. They can't capture responses to events they haven't seen.

The responsible position: silicon sampling is useful for rapid prototyping, hypothesis generation, and situations where traditional surveys are impractical. It's not a replacement for high-stakes research where accuracy is critical.

0.14.5 Calibration and Trust

What would make silicon sampling trustworthy?

The answer is calibration—systematically measuring how AI responses compare to human responses and correcting for systematic biases.

The research program works like this:

1. Take historical survey data—the General Social Survey, Pew polls, election exit polls—where we know both the demographics and the actual responses.
2. Construct matching personas and run the language model.
3. Compare AI responses to human responses across demographic subgroups.
4. Learn the systematic biases: where does the model overestimate conservative responses? Underestimate enthusiasm for certain policies? Miss regional variation?
5. Apply corrections to future silicon sampling results.

This is the same logic as survey weighting. Traditional surveys oversample some groups and undersample others; weights correct for these imbalances. Silicon sampling has its own biases; calibration weights would correct for those.

The result would be defensible uncertainty: not "the model says 58% support this policy" but "our calibrated estimate is 58% ±5%, based on historical accuracy for this question type and demographic distribution."

This research is ongoing. We're not there yet.

0.14.6 The Democratization Pattern

Silicon sampling follows the same pattern as the other tools in this book.

Domain	Traditional	Silicon
Policy analysis	JCT/CBO ($millions, months)	PolicyEngine (free, instant)
Survey research	Field surveys ($50K, weeks)	HiveSight ($50, minutes)
Statistical analysis	Stata licenses, expertise	LLM-assisted analysis

The cost reduction is roughly 1000x. The time reduction is roughly 100x. The access expansion is enormous.

This creates new possibilities. A local advocacy group can test message framings before a campaign. A product designer can check reactions to features before building them. A policymaker can gauge public sentiment before drafting legislation. A researcher can generate hypotheses before committing to expensive data collection.

None of these replace the rigorous version. A presidential campaign should still field real surveys. A pharmaceutical company should still run clinical trials. High-stakes decisions deserve high-stakes research.

But many decisions don't currently get any research because the bar is too high. For those, approximate answers are better than no answers.

0.14.7 Can We Simulate What People Think?

There's something philosophically strange about simulating opinions.

Policies have objective effects—or at least effects we can model mechanistically. Taxes flow through defined rules. Benefits have eligibility criteria. Labor supply responds to incentives according to (debatable but empirically grounded) economic theory.

But opinions aren't like that. What someone thinks about minimum wage isn't computed from rules. It emerges from a lifetime of experiences, information, identity, and reasoning processes we don't fully understand.

When we simulate an opinion, what are we actually doing?

One view: We're predicting what a person with certain characteristics would say, based on statistical patterns in how similar people have answered similar questions. This is fundamentally what survey research does—extrapolate from samples to populations. Silicon sampling just uses AI as the extrapolation mechanism.

Another view: We're creating a simulacrum that looks like opinion but isn't the real thing. The AI doesn't believe anything; it predicts text. When it says "I strongly oppose this policy," there's no one home who actually opposes anything.

Both views have merit. For practical purposes—market research, hypothesis generation, policy feedback—the first view suggests silicon sampling can be useful. For understanding what people actually think, for respecting their autonomy as participants in democracy, the second view suggests caution.

The question "should we?" is separate from "can we?"

0.14.8 Toward Opinion Infrastructure

Despite these caveats, I think silicon sampling will become standard infrastructure for understanding public sentiment. Not because it's perfect, but because the alternatives are often impractical.

The path forward involves:

Radical transparency. Every silicon sampling result should come with methodology: which model, which personas, what limitations. Users should understand they're seeing AI predictions, not human responses.

Continuous calibration. Regular benchmarking against traditional surveys to measure and correct biases. Published accuracy metrics by question type and demographic group.

Complementary use. Position silicon sampling as augmenting, not replacing, human research. Use it to narrow hypotheses, then verify with real surveys where it matters.

Integration with simulation stack. HiveSight connects to the same microdata that powers PolicyEngine. This creates possibilities: simulate a policy change, then simulate public reaction to that change, all using consistent demographic profiles.

The vision is a tool that answers: "If you implemented this policy, here's what would happen to household incomes (PolicyEngine) and here's what people would think about it (HiveSight)." Both predictions, both uncertain, both more useful than no information.

0.14.9 The Information Economy

Traditional survey research is one of the few remaining moats protecting established institutions. When it costs $50,000 to ask what people think, only well-funded organizations can ask. This concentrates information—and therefore power—among those who can afford it.

Silicon sampling disrupts this in the same way PolicyEngine disrupts budget scoring. The information becomes accessible. The playing field levels.

A first-time candidate can test campaign messages. A neighborhood association can gauge support for a development project. A student researcher can explore hypotheses without grant funding.

Not all of these uses will be responsible. Just as calculators can be used for good or ill, survey simulation can serve manipulation as well as understanding. Someone could use HiveSight to craft maximally persuasive disinformation targeted at specific demographics.

The response isn't to restrict the tools. It's transparency about what they are and education about how to interpret them. Silicon sampling is approximate, biased in knowable ways, and best used for exploration rather than conclusion.

But approximate knowledge, widely distributed, might be better than precise knowledge, narrowly held.

What do people think about this policy?

We could field a survey. That would take six weeks and cost fifty thousand dollars.

Or we could simulate it. That would take six minutes and cost fifty dollars.

The simulation won't be as accurate. But is waiting six weeks more accurate than acting on rough information now? Is the policy decision that would otherwise get made with no data better than one informed by approximate data?

These are the questions silicon sampling forces us to ask. Not whether AI can replace human research—it can't. But whether the perfect should be the enemy of the good.

0.15 Chapter 12: Simulating Democracy

Note to readers: This chapter describes theoretical research, not validated tools. Democrasim is a toy model—a thought experiment in code designed to explore how voter information might affect democratic outcomes. Unlike PolicyEngine (a validated production system) or HiveSight (a preliminary prototype with some empirical validation), Democrasim makes simplifying assumptions that real political scientists would rightly criticize. Use this chapter to understand intuitions about information and democracy, not as established political science.

Why do democratic outcomes often diverge from voter welfare?

It's a puzzle that has occupied political scientists for decades. We have mechanisms—elections—designed to translate public preferences into policy. We have representatives whose job is to understand what constituents want. We have a free press, public education, and more access to information than any previous generation.

And yet: policies regularly fail to reflect what would actually benefit voters. Tax codes favor the wealthy despite majority support for progressivity. Climate action stalls despite broad concern about warming. Healthcare systems remain inefficient despite universal frustration.

The standard explanations point to special interests, gerrymandering, and media manipulation. These matter. But there's a more fundamental problem hiding in plain sight.

Voters don't know what policies would actually do for them.

0.15.1 The Perception Problem

Consider a voter named Sarah. She's 42, works as a teacher, earns $65,000 a year, has two kids, and lives in Ohio. She's deciding between candidates with different policy platforms.

Candidate A proposes expanding the Child Tax Credit by $1,000 per child. Candidate B proposes eliminating the state income tax. Which would benefit Sarah more?

Without calculation, Sarah has to guess. Maybe she has a vague sense that she pays state income tax and that feels painful. Maybe she remembers the expanded CTC during COVID and it felt helpful. Maybe she's heard talking heads argue about either policy on cable news, each spin designed to persuade rather than inform.

Her choice will be some combination of:

- Her actual policy preferences (what she values)
- Her perceived policy impacts (what she thinks would happen)
- Noise (irrelevant factors that shouldn't matter but do)

Even if Sarah has clear preferences—say, she cares most about maximizing her family's resources—her vote may not reflect those preferences because she can't perceive the true policy impacts. She votes with a noisy signal.

Now multiply this by 150 million voters. Each making decisions based on imperfect perceptions. Some biased left, some biased right. Some well-informed about economics but clueless about environment. Some sophisticated about tax policy but misunderstanding healthcare.

The aggregate result: democratic outcomes that only roughly, noisily track what voters actually want.

0.15.2 Modeling the Noise

To explore this idea, I've built a toy model called Democrasim. It's not a validated research tool—it's a thought experiment in code, a way to reason about how voter information affects democratic outcomes. The model makes simplifying assumptions that real political scientists would rightly criticize. But it helps clarify the intuition.

Democrasim simulates the full chain from voter cognition to electoral outcomes to welfare.

Each simulated voter has:

Weighted preferences across policy dimensions. One voter might care 50% about economic issues, 30% about social issues, 20% about environment. Another might weight them 20/40/40. These are the true values—what voters actually care about.

Accuracy. How well voters perceive actual policy impacts. High accuracy means perceiving close to truth. Low accuracy means perceiving through heavy noise.

Bias. Systematic distortions beyond random noise. A voter might consistently underestimate environmental costs or overestimate tax burden.

Turnout probability. Whether they vote at all.

The perception model is simple:

```
perceived_impact = true_impact + noise + bias
```

Where the noise term scales inversely with accuracy. A voter with accuracy 0.8 perceives policy impacts with less noise than one with accuracy 0.2.

Elections work through straightforward utility maximization. Each voter evaluates candidates based on perceived impacts weighted by preferences. The candidate with highest perceived utility gets their vote. Most votes wins.

The question Democrasim asks: What happens to welfare outcomes when voter accuracy varies?

0.15.3 The Accuracy-Welfare Connection

The simulation results are intuitive but their magnitude is striking.

When accuracy is high—voters perceive close to true policy impacts—electoral outcomes track welfare. The candidate whose policies would actually improve voter lives tends to win.

When accuracy is low—voters perceive through heavy noise—the connection frays. Winners might have good policies or bad ones; the signal is too corrupted to reliably select the beneficial option.

Bias introduces a different distortion. If voters systematically underestimate certain costs or overestimate certain benefits, elections will systematically favor policies that exploit those biases, regardless of actual welfare impact.

The threshold matters. There appears to be a level of voter accuracy below which democracy becomes essentially random—electoral outcomes bear no meaningful relationship to what would actually benefit voters. Above that threshold, the relationship strengthens.

This has implications for anyone who cares about democratic function.

0.15.4 What Determines Accuracy?

If voter accuracy drives welfare outcomes, what determines accuracy?

Information availability. Can voters access relevant data about policy impacts? Or is analysis locked behind paywalls, jargon, and institutional barriers?

Information quality. Is available information designed to inform or to persuade? Think tanks with ideological agendas produce analysis, but it's systematically biased.

Cognitive load. Even with good information, understanding policy impacts requires effort. Busy people with jobs, families, and finite attention can't spend hours analyzing every ballot measure.

Trust. When institutions have been wrong before—or are perceived as biased—voters rationally discount their claims. Even accurate information gets filtered through skepticism.

This explains why throwing more information at voters doesn't automatically help. A 200-page CBO report is technically available; it doesn't mean voters absorb it. Opposing partisan analyses might both be available; voters can't adjudicate between them.

What would actually improve accuracy? Not just more information, but *accessible*, *trustworthy*, *personalized* information about policy impacts.

0.15.5 The PolicyEngine Connection

This is where tools like PolicyEngine become democratically significant.

Consider what PolicyEngine offers:

- **Personalized calculation**: Not "the average family pays X" but "your family, with your specific circumstances, would pay/receive Y under this policy."
- **Transparent methodology**: Open source code anyone can inspect. No hidden assumptions or ideological bias baked in.
- **Instant access**: Free, available to anyone with a web browser. No need to trust intermediaries.

In Democrasim terms, PolicyEngine is an accuracy multiplier. It takes voters from "I vaguely sense this policy would hurt/help me" to "I calculated that this policy would change my household income by $X."

The voter Sarah we met earlier? Instead of guessing whether the CTC expansion or state tax elimination benefits her more, she could enter her household details and see: CTC expansion would give her $2,000; *statetaxeliminationwouldsaveher*1,400. Now she has signal, not noise.

Multiply this across millions of voters. Each with clearer perception of actual policy impacts. The aggregate result: elections that more reliably track voter welfare.

This is the democratic case for open microsimulation. It's not just about individual convenience—though that matters. It's about improving the signal quality of democratic feedback loops.

0.15.6 Closing the Loop

Democrasim suggests a research program: connect simulation tools to model the full democratic cycle.

Step 1: Candidates have policy platforms. Not vague promises, but specific proposals. "Expand CTC by $500 per child" or "Replace income tax with consumption tax."

Step 2: PolicyEngine calculates actual impacts on voter households. Given a representative sample of households, we know the distribution of effects—who gains, who loses, by how much.

Step 3: Voters with varying accuracy perceive these impacts. High-accuracy voters perceive close to truth. Low-accuracy voters perceive through noise and bias.

Step 4: Votes aggregate to determine the winner.

Step 5: PolicyEngine calculates welfare outcomes under the winning policy.

This closed loop lets us ask questions like: What level of voter accuracy is needed for democratic outcomes to track welfare? How do different interventions compare? Does uncertainty systematically favor certain policy types?

Playing with the model—and I emphasize "playing," not "researching"—suggests an intuition: accuracy interventions might matter more than turnout interventions. If the toy model's logic holds, getting low-information voters to perceive better may matter more than getting non-voters to vote. An uninformed vote adds noise; an informed vote adds signal.

Whether this intuition survives contact with real political science is unknown. The model is too simple to make confident claims. Real voter behavior involves identity, tribal loyalty, strategic considerations, and psychological factors that Democrasim ignores entirely. But as a way to reason about why accurate policy information might have democratic value, it's useful.

0.15.7 The Democratic Case for Open Infrastructure

The argument crystallizes:

Premise 1: Democratic outcomes track voter welfare only to the extent voters accurately perceive policy impacts.

Premise 2: Most voters perceive policy impacts poorly—through noise, bias, and inadequate information.

Premise 3: Tools exist that could dramatically improve voter perception accuracy.

Conclusion: Investing in accessible, trustworthy policy analysis tools has democratic value beyond individual utility. It's infrastructure for informed self-governance.

This reframes PolicyEngine's mission. It's not just "a useful calculator for nerds who want to optimize their taxes." It's "infrastructure that helps democracy work as intended."

The philanthropic implications are significant. A dollar spent making policy analysis accessible might have higher democratic return than a dollar spent on get-out-the-vote campaigns. Both matter, but one addresses signal quality while the other addresses signal quantity.

0.15.8 Simulating Democratic Scenarios

Democrasim enables scenario analysis for democratic reform.

Scenario: Universal Policy Calculators

What if every voter had access to PolicyEngine-style tools and actually used them? Accuracy doubles. Simulations show electoral outcomes significantly closer to welfare-optimal policies.

Scenario: Improved Civics Education

What if schools taught policy analysis skills? Accuracy increases modestly. Smaller effect because skills without accessible tools still leave voters unable to calculate.

Scenario: Reduced Media Bias

What if news coverage focused on policy impacts rather than horse races? Bias decreases. Accuracy unchanged. Outcomes improve for bias-sensitive questions but not noise-sensitive ones.

Scenario: AI Policy Advisors

What if every voter had an AI assistant that could answer "how would Policy X affect me?" based on their circumstances and reliable models? This is accuracy approaching 1.0. Electoral outcomes closely track welfare.

This last scenario isn't science fiction. As we explored with Cosilico, AI systems can already use PolicyEngine to answer household-specific policy questions. The barrier is deployment, not capability.

0.15.9 Futarchy: Vote Values, Bet Beliefs

There's a more radical proposal for connecting information to governance: Robin Hanson's *futarchy* Hanson [2013].

The core idea is captured in a slogan: "Vote on values, but bet on beliefs."

Under futarchy, democratic processes still determine *what we care about*—a measure of national welfare, say, or specific metrics like child poverty rates and median income. But *which policies achieve those goals* is determined by prediction markets, not politicians.

Here's how it would work. A legislature proposes a bill. Prediction markets open on national welfare conditional on the bill passing versus failing. If the market says welfare will be higher with the bill than without, it becomes law. If not, it doesn't.

The logic is that markets aggregate information better than deliberation Hanson [2000]. Traders with relevant knowledge profit by pushing prices toward truth. Ideologues who let conviction override evidence lose money. The market converges on the best available estimate of policy effects.

This sounds extreme. But the underlying mechanism—separating values from facts—clarifies what simulation tools actually do.

PolicyEngine answers factual questions: "What would happen to child poverty if we passed Policy X?" Prediction markets can answer similar questions: "Given Policy X, what do informed bettors expect to happen to child poverty?"

Both separate the *empirical* question (what would happen?) from the *normative* question (is that outcome good?). Democrasim's accuracy-welfare model makes the same distinction: voters have preferences (values), and they perceive policy impacts (facts). Better fact-perception improves the connection between voting and welfare.

Personal Futarchy

The futarchy framework applies at individual scale too.

I've built a tool called Farness that implements "personal futarchy" for decisions Ghenis [2024]. Instead of asking "should I take the new job?" it structures the decision into:

1. **KPIs** (values): What outcomes do I actually care about? Income, satisfaction, work-life balance?
2. **Options**: What are all the choices, including ones I haven't considered?
3. **Forecasts** (beliefs): For each option, what's my prediction for each KPI—with confidence intervals?

The discipline of making explicit numeric forecasts—rather than vague intuitions about what "feels right"—mirrors futarchy's bet-on-beliefs principle. The calibration tracking mirrors prediction market scoring: over time, you learn whether your forecasts are systematically biased.

The connection to this chapter is direct. If a voter could run Farness on "how should I vote on Proposition 12?"—defining KPIs (household income, public services, equity), forecasting each candidate's impact on those KPIs—their vote would contain more signal.

Why Not Full Futarchy?

Despite its elegance, futarchy faces practical objections that explain why no democracy has adopted it:

Manipulation: Can wealthy actors move market prices to get their preferred policies passed? Hanson argues proper market design prevents this, but the concern persists.

Thin markets: Most policy questions don't attract enough trading volume for reliable price discovery. Polymarket works for presidential elections with billions wagered; it's less clear how it would work for zoning amendments.

Conditional complexity: Markets on "welfare if Policy X passes" require defining welfare, measuring it, and handling the counterfactual. These are hard problems.

Democratic legitimacy: People accept losing elections because they had a vote. Would they accept losing market-based decisions? The psychology of legitimacy matters.

Still, the futarchy thought experiment illuminates what democratic information infrastructure could do. Even without replacing elections, we can build tools that help voters bet on beliefs more rigorously—and calibrate those beliefs against outcomes.

0.15.10 Objections and Complications

The accuracy-welfare model simplifies real democratic complexity.

Objection: Preferences themselves are the problem.

Maybe voters don't just perceive poorly—they have bad preferences. They want policies that harm others. They vote from spite rather than self-interest.

Response: True, but orthogonal. If voters want harmful policies and perceive accurately, they'll get harmful policies. That's a different problem than wanting beneficial policies but perceiving poorly. Better perception at least ensures voters get what they want, even if what they want is bad.

Objection: Information won't reach the disengaged.

People who don't vote probably wouldn't use policy calculators either. The voters who would use PolicyEngine already vote and may already be relatively informed.

Response: Partially true, but the marginal effect still matters. Moving moderately-informed voters to well-informed improves signal quality. And new modalities (AI assistants, social sharing) may reach previously disengaged populations.

Objection: Calculated self-interest isn't the same as good citizenship.

Democracy might benefit from voters considering communal welfare, not just personal impact.

Response: True, and PolicyEngine can calculate societal impacts too. The point isn't selfishness—it's replacing perception with calculation, whatever voters choose to calculate.

0.15.11 The Vision

Imagine a voter in 2030.

She's considering a ballot measure to reform her state's tax code. Instead of reading dueling op-eds and trying to guess who's lying, she opens a policy analysis app on her phone.

"Show me how Proposition 12 would affect my household."

The app—powered by something like PolicyEngine, accessed through something like Cosilico's AI layer—returns: "Based on your household profile, this measure would reduce your state taxes by $340 per year. The trade-off is reduced funding for public education, which your children use. The net welfare impact for your household is approximately..."

She might not accept the calculation blindly. She might weigh factors the app doesn't capture. But she starts from signal, not noise.

Now she considers communal impacts. "How would this affect households statewide?"

The app shows distributions: "Households earning over $200,000 receive average benefits of $2{,}400. Households earning under $50,000 see average benefit of $80. Total revenue reduction of $2.1 billion..."

This voter can make an informed choice—not because she's smarter or more educated, but because she has tools that convert policy proposals into understandable impacts.

Across millions of such voters, elections become more responsive to actual preferences. Democracy functions closer to its ideal.

The simulation has limits. Democrasim doesn't capture cultural dynamics, identity politics, or the psychology of tribal belonging. Voters aren't just utility-maximizing calculators.

But the core insight holds: to the extent democracy is *supposed* to translate preferences into outcomes, it needs voters who can perceive what outcomes would actually result from different choices.

Open microsimulation is infrastructure for that perception. PolicyEngine isn't just a tax calculator. It's a component of democratic signal processing.

The code simulates policies. But what we're really simulating is the possibility of informed self-governance.

0.16 Chapter 13: Simulating Values

Note to readers: This chapter describes early-stage research, not established methodology. Unlike PolicyEngine (validated by government use and millions of simulations) or even HiveSight (with preliminary empirical validation), value forecasting is at the experiment stage. I ran the first systematic tests in 2024—LLMs outperformed baselines by 2.2×, but also missed an unexpected reversal in the 2024 GSS data. The results suggest this research direction is promising but far from proven. Treat the philosophical framework as speculative and the empirical claims as preliminary findings requiring extensive validation.

Validation status across the book's tools:

- **Proven**: PolicyEngine (government-validated, million+ simulations, production system)
- **Preliminary**: Cosilico and HiveSight (prototypes with some validation, not production-ready)
- **Theoretical**: Democrasim and value forecasting (experimental models with limited validation)

We've built tools to simulate households, policies, voters, and opinions. But the deepest question isn't "what do people want today?"

It's: *What would they want after reflection?*

This chapter ventures into territory where simulation meets philosophy, where forecasting meets ethics. The practical tools we've built in earlier chapters connect to a larger question: How do we align increasingly powerful AI systems with human values?

In 2024, I ran the first systematic experiments testing whether large language models can forecast value change. The results were striking: LLMs outperformed statistical baselines by a factor of 2.2, but they also revealed the profound difficulty of the problem. When the 2024 General Social Survey showed an unexpected reversal in same-sex acceptance—dropping from 72% to 55%—every model, including the LLMs, missed it.

0.16.1 The Alignment Problem

As AI systems become more capable, a question looms: What should they be trying to do?

The naive answer—"do what humans want"—immediately fractures. Which humans? Their stated preferences or revealed preferences? What they want now or what they'd want with better information? What one culture values or what's universal?

Current approaches to AI alignment take different positions:

RLHF (Reinforcement Learning from Human Feedback) trains AI on current human preferences. Rate these outputs; the model learns what we approve of. Problem: our current preferences may be biased, inconsistent, or short-sighted.

Constitutional AI defines principles the AI should follow. Don't be harmful, be helpful, be honest. Problem: who writes the constitution? How do you handle genuine moral disagreement?

Idealized values asks what fully rational, fully informed humans would want. Problem: this is a philosophical thought experiment, not an empirical research program.

Each approach has merit. None has cracked the fundamental difficulty: values are contested, evolving, and uncertain.

0.16.2 What If We Could Forecast Values?

Here's an alternative framing: What if AI alignment isn't about finding the "correct" values, but about *forecasting* where values are heading?

Consider same-sex marriage. In 1986, 32% of Americans supported it. By 2021, 79% did. The change wasn't random—it followed patterns that, in retrospect, seem predictable. As exposure increased, as generational replacement occurred, as arguments crystallized, support grew.

What if you could have predicted this trajectory in 1996?

More importantly: what if you could predict which of today's contested values will be accepted and which rejected after another generation of reflection?

This isn't about finding moral truth. It's about forecasting moral change—the same way economists forecast economic variables or meteorologists forecast weather. Not perfectly, but better than chance.

0.16.3 The Value Forecasting Proposal

The idea is simple in concept, ambitious in execution:

Step 1: Train language models on historical data. Use surveys from decades past—the General Social Survey back to 1972, Gallup polls, the World Values Survey. Train models to understand what people believed, why, and how those beliefs connected to demographic characteristics.

Step 2: Test predictive accuracy. If you train a model on data through 1996, can it predict value trajectories through 2021? This is empirically testable. Either the model predicts moral change better than baseline or it doesn't.

Step 3: If validated, project forward. What values would humanity converge toward after extended reflection? Not a decade, but a century. Not with today's constraints, but in post-scarcity conditions.

Step 4: Use that projection as an alignment target. Instead of aligning AI to our current, possibly confused values, align it to our projected post-reflection values.

This transforms alignment from philosophy to forecasting.

0.16.4 Heterogeneity as Feature

Here's the crucial insight: don't assume humanity converges to one value system.

A post-scarcity, post-reflection humanity might look like:

Value System	Population Share
Individual liberty priority	30%
Collective welfare priority	25%
Environmental stewardship	20%
Spiritual/transcendent focus	15%
Other frameworks	10%

The alignment target isn't picking one. It's the *distribution*.

An AI aligned to this distribution would:

- Take actions that score well across multiple value systems
- Exercise caution when value systems disagree
- Avoid moves that catastrophically violate any significant fraction of values
- Support conditions where diverse value systems can coexist

This is value pluralism as engineering constraint, not philosophical preference.

0.16.5 Uncertainty at Two Levels

The projection comes with uncertainty—not just statistical noise, but two distinct types:

Aleatoric uncertainty: genuine heterogeneity across the population. Even in a post-reflection world, different people will value different things. This isn't ignorance to be eliminated; it's reality to be modeled.

Epistemic uncertainty: our uncertainty about what the distribution would be. We don't know with confidence what post-reflection humanity values. We have a distribution *over* distributions.

Both must be quantified.

Not "humanity will value X" but "P(humanity values X) = 0.3 with 90% CI [0.2, 0.4]"

This is the same uncertainty quantification theme from Chapter 10, applied to values instead of policy costs.

0.16.6 The Empirical Test

What makes this a research program rather than armchair philosophy?

The key is historical validation. We have decades of survey data capturing how values actually changed. This provides ground truth for testing predictive models.

The experiment I ran:

1. Take 17 GSS variables spanning 1972–2022: attitudes on homosexuality, marijuana, gender roles, race, religion, and more
2. Give GPT-4o the historical time series through 2021 and ask for predictions of 2024 values
3. Compare those predictions to actual 2024 GSS data (released in late 2024)
4. Elicit full probability distributions—not just point estimates—using quantile prompting (10th, 25th, 50th, 75th, 90th percentiles)
5. Calibrate uncertainty using EMOS (Ensemble Model Output Statistics), the same technique meteorologists use for weather forecasts

The results:

Metric	GPT-4o	Linear Extrapolation	Historical Mean
Mean Absolute Error	4.8 pp	7.2 pp	10.6 pp
vs. Baseline	**2.2× better**	—	—

LLMs genuinely outperformed naive baselines. They captured something about the structure of value change that simple trend extrapolation missed.

But calibration required work. Raw LLM confidence intervals were 21% too narrow—the model was overconfident. After EMOS calibration (multiplying the spread by 1.21×), the 80% prediction intervals achieved proper coverage.

The biggest lesson came from failure. Same-sex acceptance (HOMOSEX) had risen steadily from 11% in 1973 to 72% in 2022. Every model—LLM, linear, historical—predicted continued increase. The 2024 GSS showed 55%, a 17-point drop. The supposed "inevitable" trajectory reversed.

Was this a measurement artifact? A real backlash? It's too early to say. But it demonstrated something crucial: value change contains genuine surprises that no model—statistical or AI—can fully anticipate.

0.16.7 The Historical Record

Consider what a value forecasting model can learn from five decades of GSS data:

Value	1973	1990	2010	2022	2024
Same-sex relations "not wrong"	11%	15%	44%	72%	55%*
Marijuana legalization	19%	17%	44%	68%	70%
Women working outside home	68%	78%	80%	88%	83%
Confidence in science	42%	44%	40%	48%	44%

*The 2024 HOMOSEX drop is under active investigation—it may reflect survey methodology changes, real backlash, or sampling variation.

The patterns aren't random walks. They show consistent long-term structure:

- **Generational replacement matters**: younger cohorts hold different views that persist as they age
- **Exposure effects**: knowing gay people correlates with supporting their rights
- **Information cascades**: once a threshold is passed, change often accelerates
- **Moral arguments crystallize**: the articulation of principled positions shifts debate

But the 2024 data taught humility. The patterns are tendencies, not laws. Short-term reversals happen. What seemed like inevitable progress can stall or retreat.

0.16.8 Long-Term Projections

Despite the 2024 surprise, I extended the forecasting exercise to longer horizons. If the 50-year trend matters more than year-to-year noise, what do calibrated forecasts suggest for 2030, 2050, and 2100?

Same-sex acceptance (HOMOSEX):

Year	Median	80% CI
2030	62%	[49%, 75%]
2050	72%	[57%, 87%]
2100	80%	[69%, 91%]

The model projects eventual convergence toward broad acceptance, but with wide uncertainty bands. The 2024 dip may be a temporary fluctuation in a longer trajectory—or the beginning of a sustained reversal. The confidence intervals capture both possibilities.

The key insight: even with short-term forecasting errors, long-term projections may be more reliable. The 50-year trajectory for HOMOSEX shows +44 percentage points despite year-to-year volatility. Generational replacement creates momentum that individual survey years cannot reverse.

0.16.9 The Connection Back

Every tool in this book contributes to value forecasting:

PolicyEngine provides the consequence modeling. If we project that future humanity values environmental sustainability, what policies would they enact? What would those policies cost? PolicyEngine calculates.

HiveSight provides the heterogeneity infrastructure. Value forecasting doesn't ask "what would humanity think?" but "what would this representative sample of diverse personas think?" The same microdata-grounded diversity applies.

Democrasim provides the mechanism modeling. Given a distribution of values, how do democratic processes aggregate them? What policies result? Does the electoral outcome track the value distribution?

Squigglepy provides uncertainty quantification. Value forecasts aren't point estimates but probability distributions with confidence intervals.

Cosilico provides the integration layer. AI agents that can reason about policies, using deterministic tools, now have a target: not "what do users currently want?" but "what would reflective humanity value?"

The full stack:

```
Value distribution (forecast)
    |
    v
Democratic simulation (Democrasim)
    |
    v
Policy selection (election results)
    |
    v
Policy implementation (PolicyEngine)
    |
    v
Welfare outcomes (measured)
    |
    v
Feedback: do outcomes align with projected values?
```

0.16.10 Why Not Just Ask People?

A reasonable objection: if you want to know what people value, why not just ask them?

Several reasons:

Current preferences are noisy. People are busy, distracted, misinformed. Their responses reflect momentary framing, media influence, tribal signaling. What they say they value and what they would value after careful reflection often differ.

Values evolve. Asking 1990 Americans about same-sex marriage wouldn't predict 2020 attitudes. Current surveys capture current states, not trajectories.

Reflection takes time we don't have. Ideally, every policy decision would follow extended democratic deliberation. In practice, decisions happen faster than reflection. Forecasting approximates the reflective outcome.

Scale and cost. Deep deliberative processes are expensive. Value forecasting offers an approximate, scalable complement.

The answer isn't "don't ask people"—it's "ask people, model the patterns, project the trajectories, and be transparent about uncertainty."

0.16.11 The Philosophical Precedents

This isn't entirely new. Philosophers have grappled with the question of which values to take seriously.

Idealized values (various philosophers): What would a fully rational, fully informed person choose? The problem is this remains thought experiment; no one specifies how to compute the answer.

Reflective equilibrium (Rawls): The back-and-forth between principles and considered judgments until they cohere. Value forecasting operationalizes this as prediction: what would people conclude after that process?

Axiological futurism (Danaher): The systematic study of how values change. Value forecasting adds: "and can we predict it?"

The innovation isn't the question but the method: empirical validation, probabilistic framing, computational implementation.

The Reflection Problem

But here's a crucial caveat: temporal change isn't the same as reflection.

That 79% supported same-sex marriage in 2021 versus 32% in 1986 shows *more time passed*, not necessarily *more careful reasoning*. The mechanisms driving value change—generational replacement, media exposure, information cascades, tribal signaling—are sociological, not purely epistemic.

Some value changes clearly involve improved reasoning: the abolition of slavery, the extension of suffrage. Others may reflect preference drift without moral progress. Still others—like the 2024 HOMOSEX reversal—might represent backlash to perceived overreach rather than either progress or regress.

The philosophers who've thought hardest about this offer partial guidance:

Rawls distinguished between reasonable and unreasonable comprehensive doctrines—not all value systems deserve equal weight in political deliberation. But operationalizing "reasonable" is difficult.

Habermas emphasized ideal speech conditions: undistorted communication, equal participation, no coercion. Temporal extrapolation doesn't guarantee these conditions obtained during the value change.

Sen and Nussbaum warned about "adaptive preferences"—values formed under oppression shouldn't guide alignment. A society that normalizes injustice may show stable preferences for injustice. Extrapolating those preferences forward would be perverse.

The honest answer: we don't have a fully satisfying account of when temporal change tracks moral reflection versus mere drift. The empirical validation helps—if temporal extrapolation predicts well, that's evidence for underlying structure—but it doesn't solve the deeper philosophical problem.

This uncertainty is one more reason for humility. Value forecasting provides evidence, not answers. It should inform democratic deliberation, not substitute for it.

0.16.12 Objections and Responses

"This is moral relativism."

Not quite. Value forecasting predicts what values humanity would hold after reflection—not that all values are equally valid. It's empirical, not nihilistic. If the forecast is that post-reflection humanity would reject torture, that's not relativism; that's prediction.

"Values depend on contingent factors. They're not predictable."

Maybe. That's an empirical question. The historical validation step tests it. If value change is unpredictable, the model will fail to predict, and we'll know. Current evidence suggests partial predictability.

"Who decides what counts as 'reflection'? Isn't this smuggling in values?"

Fair critique. The experimental design makes assumptions: that GSS captures values, that temporal extrapolation is valid, that post-scarcity conditions matter. These are choices. They should be transparent, and different research programs could make different choices.

"This could be used for manipulation."

True. Knowing how values evolve could help those who want to accelerate or prevent certain changes. But the same is true of all social science. Understanding doesn't mandate manipulation.

0.16.13 The Governance Question

If value forecasting works, it becomes a technology of power. Who controls it matters.

Consider the stakes. A government could use value forecasts to justify paternalistic policies: "We're implementing what you'll value in 20 years." A corporation could use them to anticipate and shape consumer preferences. An AI lab could use them to align systems to projected values that serve its interests rather than humanity's.

These aren't hypothetical risks. They're the predictable consequences of forecasting technology in a world of unequal power.

The governance framework must address:

Who generates forecasts? Academic researchers? Government agencies? Private companies? Each brings different incentives and accountability structures. Distributed generation with open methodology may be more trustworthy than centralized control.

What conditioning assumptions are transparent? A forecast conditioned on "post-scarcity with extended deliberation" differs from one conditioned on "persistent inequality with polarized media." The assumptions encode values themselves. They must be explicit and contestable.

What role for democratic input? Forecasts should inform democratic deliberation, not replace it. Citizens should be able to examine, critique, and reject forecast-based justifications. The forecast is evidence, not authority.

What appeals processes exist? When forecasts are contested—and they will be—how do disagreements resolve? Peer review, public comment, independent replication?

How do forecasts interact with current feedback? If value forecasts suggest future humanity will reject a practice that current humanity endorses, what weight does each get? This is the deepest question, and honest answers remain uncertain.

The HOMOSEX reversal illustrates the stakes. Had forecasters in 2020 projected continued acceptance and used that to justify policies, the 2024 data would have revealed overconfidence. The governance system must accommodate surprise, revision, and humility.

Value forecasting as academic research is one thing. Value forecasting as input to deployed AI systems is another. The institutional arrangements for the latter don't yet exist. Building them—with appropriate checks, transparency, and democratic oversight—is as important as the forecasting methodology itself.

0.16.14 The Capstone

Throughout this book, simulation has served as a lens:

- Microsimulation reveals how policies affect households
- Democratic simulation reveals how knowledge affects elections
- Opinion simulation reveals what diverse populations think

- Value forecasting reveals where humanity's values might lead

Each layer builds on the last. Each uses the same core infrastructure: representative microdata, transparent methodology, uncertainty quantification.

And each asks a version of the same question: *What would happen if we could see more clearly?*

PolicyEngine asks: What would happen if households saw their true policy impacts?

HiveSight asks: What would people say if we could ask everyone?

Democrasim asks: What would elections produce if voters were informed?

Value forecasting asks: What would humanity choose if given time to reflect?

The simulation stack doesn't answer these questions definitively. It constructs scenarios, quantifies uncertainty, and makes the implications tractable.

0.16.15 The Aspiration—and the Evidence

The deepest version of this vision:

AI systems aligned not to our current preferences—confused, biased, evolving—but to our projected post-reflection values. Not to what we happen to want this moment, but to what we would endorse after centuries of collective reasoning.

This is humility, not arrogance. It says: we don't know the right values with certainty. We know our current values are provisional. The best we can do is forecast, quantify uncertainty, and update as we learn.

The 2024 experiments provided both encouragement and caution.

Encouragement: LLMs captured value dynamics that simple baselines missed. A 2.2× improvement over extrapolation suggests genuine pattern recognition. Calibration techniques from meteorology (EMOS) successfully corrected overconfidence. The infrastructure works.

Caution: The HOMOSEX reversal humbled every forecaster. Value change contains irreducible surprises. Any system claiming to know humanity's future values with confidence is selling certainty it doesn't have.

The research program is no longer "just beginning"—the first experiments are complete. Historical validation showed predictive power. Long-term forecasts with calibrated uncertainty now exist. But the work has revealed how much remains unknown.

What would humanity want after reflection?

We can't know with certainty. But we might be able to forecast with calibrated uncertainty.

And that forecast—representing our best probabilistic guess about considered human values—might be the most responsible alignment target we can specify.

Not perfect. Not final. But grounded in evidence, transparent in method, and humble about uncertainty.

Society in silico: not a deterministic prediction machine, but a probabilistic reflection engine.

We simulate policies to understand their effects. We simulate opinions to understand what people think. We simulate elections to understand how preferences aggregate. We simulate values to understand where we're heading.

The simulation doesn't replace human judgment. It informs it. It makes visible what would otherwise remain hidden. It creates the conditions for more thoughtful collective choice.

That's the aspiration. The work continues.

0.17 Chapter 14: Society in Silico

We return to where we started: Engerraund Serac standing in his control room, watching Rehoboam's predictions cascade across holographic displays. Individual lives reduced to trajectories. Society optimized according to one man's definition of optimal. "I don't predict the future," he says. "I create it."

That's one vision of society in silico.

This book has traced another.

0.17.1 Two Paths

Both paths start from the same recognition: complex social systems can be modeled computationally. Policies can be simulated before they're enacted. Populations can be represented as millions of synthetic households. Elections can be gamed out in code.

But they diverge immediately after.

Serac's path:

- The model is closed. Only he sees it.
- The predictions are hidden. People live inside the model without knowing it.
- Optimization is unilateral. His values, his objective function, his paths.
- Uncertainty is suppressed. Rehoboam speaks in certainties.
- Power concentrates. Those without access are subjects, not participants.

The open path:

- The model is public. Anyone can inspect the code.
- The predictions are accessible. People can query their own outcomes.
- Values are contested. Multiple objectives, multiple stakeholders, no single optimizer.
- Uncertainty is quantified. Confidence intervals, not false precision.
- Power distributes. The tools of analysis become public infrastructure.

Every chapter of this book has been about building the second path.

0.17.2 What We've Built

Guy Orcutt imagined simulating the economy household by household in 1957. For decades, that vision was realized only inside government agencies—Congressional Budget Office, Joint Committee on Taxation, Treasury—accessible to the powerful, invisible to citizens.

OpenFisca cracked open the model in France. The code was published. Anyone could run it. Rules as code became a movement: legislation should be readable by machines and humans alike.

PolicyEngine extended this to the US and UK, adding a web interface that lets anyone—not just programmers—see how policies affect their household. The household view: "How does this tax credit affect me?" The society view: "Who gains and who loses across the population?"

Cosilico is building the next layer: infrastructure that lets AI agents encode rules from authoritative sources with empirical validation. Not AI replacing analysis—AI using deterministic tools to ground its reasoning.

HiveSight simulates opinions. Democrasim simulates elections. Squigglepy quantifies uncertainty. Each component addresses a layer of the problem.

Together, they form the beginnings of democratic simulation infrastructure.

0.17.3 What We Haven't Built

Let me be honest about what's incomplete.

Uncertainty quantification is partial. PolicyEngine gives point estimates. "This reform costs 50*billion*."*Itshouldsay* : "*Thisreformcosts*50 billion, 90% CI [35*B*,68B]." We've identified the problem; we haven't solved it everywhere.

Value forecasting is untested. Chapter 13 proposed an empirical research program. Train on historical survey data, validate on held-out periods, project to long reflection. The experiment hasn't been run. It might fail.

Adoption is early. PolicyEngine has thousands of users, but policy debates still happen mostly without it. The tools exist; the cultural change hasn't fully occurred.

AI integration is nascent. Cosilico isn't launched. The vision of AI agents reliably using deterministic microsimulation is closer to prototype than production.

This book describes aspiration as much as achievement. We're partway up the mountain, not at the summit.

0.17.4 The Honest Caveat

There's a version of this book that would claim: "We've solved policy analysis. We've democratized the tools. We've shown AI how to reason about society. The future is bright."

That would be false.

The more honest version: We've demonstrated that open simulation infrastructure is possible. We've built enough to show the concept works. We've identified the gaps that remain. The question now is execution.

Can we add uncertainty quantification before trust erodes? Can we validate value forecasting empirically? Can we scale adoption beyond early adopters? Can we build the AI integrations that let these tools compound?

The answers aren't known. The work is ongoing.

0.17.5 Why It Matters

If society can't reason about itself, it can't govern itself.

The alternative to open simulation isn't human judgment uncorrupted by models. It's:

Black-box decisions by agencies with proprietary tools. The government runs simulations you can't see. Experts disagree about their assumptions. You're asked to trust outputs you can't verify.

Vibes-based policy debate. Politicians wave hands. Pundits assert confidently. Numbers float by without sources. Anyone with a megaphone claims authority; none provide audit trails.

AI systems aligned to current values without understanding how values evolve. LLMs trained on today's preferences, frozen in place, even as humanity's considered views change.

Power concentrated in those with access to compute and data. The rich can simulate; the poor guess. Corporations optimize; citizens react. The asymmetry compounds.

Open simulation addresses each failure mode:

- Auditable code instead of black boxes
- Reproducible results instead of vibes
- Value forecasting instead of frozen alignment
- Public infrastructure instead of private advantage

It's not guaranteed to help. But the counterfactual is worse.

0.17.6 The Democratic Argument

The deepest case for open simulation is democratic.

Democracy requires shared understanding. We vote on policies without knowing their effects. We debate taxes without calculating impacts. We judge politicians without auditing claims.

This isn't ignorance by choice—it's ignorance by limitation. The tools to understand policy were, until recently, locked inside institutions. Expert analysis was expensive and inaccessible. Citizens had opinions; experts had models; the two rarely connected.

Open microsimulation changes the equation. Now a voter can ask: "What would this candidate's tax plan do to my family?" and get an answer. Not from a partisan think tank. Not from a cable news pundit. From a transparent model they can inspect.

That's not utopia. Many won't use the tools. Many will distrust them. Many will find reality uncomfortable. But the marginal improvement matters.

A voter with accurate information may still vote against their economic interest for other reasons—values, identity, loyalty. That's their right. But a voter who wants to know can know. The option exists.

And if enough voters use the option, electoral dynamics shift. Politicians can be held to calculable claims. Platforms can be stress-tested against distributional analysis. The signal-to-noise ratio of democratic deliberation improves.

0.17.7 The AI Argument

As AI systems become more capable, they'll increasingly be asked questions about policy.

"What would happen if we expanded the child tax credit?" "How would a carbon tax affect my family?" "Which candidate's healthcare plan would benefit me more?"

The AI can respond in two ways:

Option 1: Make stuff up. Draw on training data, synthesize plausible-sounding text, perhaps hallucinate eligibility rules or invent statistics. This is what happens today. GPT-4 gets tax questions right 67% of the time—worse than a coin flip for true/false Blair-Stanek et al. [2023].

Option 2: Call reliable tools. Use PolicyEngine as a backend. Look up actual parameters from Cosilico's validated rules. Calculate instead of guess. Return auditable results.

This is the AI case for open simulation: LLMs need tools to call. Those tools should be accurate, transparent, and well-maintained. Building them is infrastructure investment that compounds as AI capabilities grow.

The alternative is AI confidently wrong about policy—a future where more people have access to information that's less reliable. That's not progress.

0.17.8 The Work Ahead

If I'm still building these tools in five years, what does success look like?

PolicyEngine covers more countries. Not just US and UK, but every jurisdiction with sufficient data. Tax-benefit models as global infrastructure.

Uncertainty is quantified. Every estimate comes with confidence intervals. Users see ranges, not false precision.

Cosilico works at scale. AI agents reliably encode rules from statutes. Validation against authoritative oracles is automatic. The human-AI partnership produces rules faster and more accurately than either alone.

Value forecasting has been tested. We know whether historical validation works. We know how far forward projections are reliable. We've integrated (or abandoned) the approach based on evidence.

Adoption is mainstream. Policy debates reference open models. Journalists query microsimulations. Voters compare candidates using shared infrastructure. The asymmetry of analysis has flattened.

Some of this will happen. Some won't. The vision adjusts as reality teaches.

0.17.9 An Invitation

This book ends with an invitation.

The work is open. The code is public. The communities are forming. If any of this resonates—the vision of transparent policy analysis, of AI grounded in reliable tools, of democratic deliberation informed by shared models—there's a way to participate.

Use the tools. Calculate your taxes on PolicyEngine. Simulate policy reforms. Find the bugs and report them.

Contribute to the code. Microsimulation models need maintenance. Rules change. Parameters update. The work never ends.

Join the conversation. Policy analysis shouldn't be an insider sport. The debates should include everyone the policies affect.

Build on the infrastructure. HiveSight started as an experiment. Democrasim is a prototype. Value forecasting is a thesis. Each could be developed further by others.

Or challenge the premise. Maybe open simulation is misguided. Maybe the tools entrench technocracy rather than democratizing it. Maybe the assumptions are wrong in ways I can't see. The argument improves through critique.

0.17.10 Closing the Loop

In Westworld, Serac's system ultimately fails. Rehoboam couldn't handle the chaos introduced by Dolores and the other hosts—beings who didn't fit its model, whose choices it couldn't predict. The closed system shattered on contact with genuine novelty.

The fiction was too neat, but the lesson holds.

Closed systems are brittle. They optimize for their assumptions. When the world shifts—and it always shifts—they break in ways their designers didn't anticipate.

Open systems are different. They expose assumptions. They invite correction. They adapt as understanding improves. They're never finished, but they're also never frozen.

Society in silico isn't about predicting the future with certainty. It's about creating tools that help us reason about possible futures with calibrated uncertainty.

Not: "This reform will cost $50 billion and save 10,000 lives."

But: "Our best estimate is $50 *billion* [90

That's not a prediction machine. It's a reasoning aid.

Can simulation help society realize its goals?

The question that opened this book has no definitive answer. Simulation can't tell us what to value. It can't guarantee we'll use insights wisely. It can't prevent bad-faith actors from gaming the tools.

What it can do is make the invisible visible. The distributional effects hidden in policy details. The uncertainty masked by point estimates. The values implicit in optimization targets. The trajectories of moral change across generations.

Making these visible doesn't solve politics. It informs it.

And that, in the end, is the aspiration. Not a perfect model of society. Not a machine that optimizes humanity. Just tools—open, transparent, uncertainty-aware—that help us see more clearly what we're choosing and why.

The work continues. The invitation stands.

Welcome to society in silico.

Bibliography

American Enterprise Institute. Open Source Policy Center Launches Tax Brain. https://www.aei.org/press/open-source-policy-center-launches-tax-brain/, 2015. URL https://www.aei.org/press/open-source-policy-center-launches-tax-brain/. Accessed December 2024.

American Enterprise Institute. Open Source Policy Center Launches Tax Brain. https://www.aei.org/press/open-source-policy-center-launches-tax-brain/, 4 2016. URL https://www.aei.org/press/open-source-policy-center-launches-tax-brain/. Matt Jensen founded OSPC to make policy analysis transparent through open source modeling.

L. P. Argyle, E. C. Busby, N. Fulda, J. R. Gubler, C. Rytting, and D. Wingate. Out of One, Many: Using Language Models to Simulate Human Samples. *Political Analysis*, 31(3):337–351, 2023. doi: 10.1017/pan.2023.2.

A. Bee and J. Mitchell. Do Older Americans Have More Income Than We Think? SESHD Working Paper 2017-39, U.S. Census Bureau, 2017. URL https://www.census.gov/content/dam/Census/library/working-papers/2017/demo/SEHSD-WP2017-39.pdf. Median income for 65+ was 30% higher in admin records; poverty rate 9.1% (survey) vs 6.9% (admin-validated).

A. Blair-Stanek, N. Holzenberger, and B. Van Durme. Openai Cribbed Our Tax Example, But Can GPT-4 Really Do Tax? *arXiv preprint*, 2023. URL https://arxiv.org/abs/2309.09992. GPT-4 achieved 67% accuracy on 276 true/false tax law questions.

Board of Governors of the Federal Reserve System. Frb/US Model. https://www.federalreserve.gov/econres/us-models-about.htm, 2024. URL https://www.federalreserve.gov/econres/us-models-about.htm. 375-equation macro model; Python implementation released 2022.

G. E. P. Box. Science and Statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976. doi: 10.2307/2286841. Source of "all models are wrong, but some are useful".

Business Wire. Avalara to be Acquired by Vista Equity Partners for $8.4 Billion. https://www.businesswire.com/news/home/20220808005258/en/Avalara-to-be-Acquired-by-Vista-Equity-Partners-for-8.4-Billion, 8 2022. URL https://www.businesswire.com/news/home/20220808005258/en/Avalara-to-be-Acquired-by-Vista-Equity-Partners-for-8.4-Billion.

H.-K. Cheng. Guy H. Orcutt's Engineering Microsimulation to Reengineer Society. *History of Political Economy*, 52(S1):191–215, 2020. doi: 10.1215/00182702-8718000.

S. R. Collins, M. Gunja, M. M. Doty, and S. Beutel. The CBO's Crystal Ball: How Well Did It Forecast the Effects of the Affordable Care Act? *Commonwealth Fund Issue Brief*, 12 2015. URL https://www.commonwealthfund.org/publications/issue-briefs/2015/dec/cbos-crystal-ball-how-well-did-it-forecast-effects-affordable.

Column Tax. Will AI Agents Help File Your Taxes? https://www.column.tax/blog/will-ai-agents-help-file-your-taxes, 2024. URL https://www.column.tax/blog/will-ai-agents-help-file-your-taxes. Accessed December 2024.

Committee for a Responsible Federal Budget. Has TCJA Paid For Itself? 2024. URL https://www.crfb.org/blogs/has-tcja-paid-itself.

Congressional Budget Office. H.R. 1628, American Health Care Act of 2017. techreport, Congressional Budget Office, 5 2017. URL https://www.cbo.gov/publication/52752. Estimated 23 million more uninsured by 2026 relative to current law.

Congressional Budget Office. An Overview of CBOLT: The Congressional Budget Office Long-Term Model. techreport, Congressional Budget Office, 2018. URL https://www.cbo.gov/publication/53667.

Congressional Budget Office. An Evaluation of CBO's Projections of Deficits and Debt From 1984 to 2023. Technical report, Congressional Budget Office, 2024. URL https://www.cbo.gov/publication/61067.

Congressional Budget Office. Cbo's Economic Forecasting Record: 2025 Update. Technical report, Congressional Budget Office, 7 2025. URL https://www.cbo.gov/publication/61334.

H. Crane. Prediction Market Accuracy in the Long Run. *International Journal of Forecasting*, 2020. Comparison of PredictIt vs FiveThirtyEight forecasts.

ESOMAR. Global Market Research Industry Revenue, 2024. Global market research industry forecasted at $140 billion in 2024.

D. R. Feenberg and E. Coutts. An Introduction to the TAXSIM Model. *Journal of Policy Analysis and Management*, 12(1):189–194, 1993.

M. Ghenis. Ubi Center: Open-source research on universal basic income policies. https://ubicenter.org/, 2019a. URL https://ubicenter.org/. Founded 2019.

M. Ghenis. Introducing the UBI Center. https://medium.com/ubicenter/introducing-the-ubi-center-72a8011bfc39, 6 2019b. URL https://medium.com/ubicenter/introducing-the-ubi-center-72a8011bfc39.

M. Ghenis. Enhancing the Current Population Survey for Policy Analysis. *PolicyEngine Blog*, 12 2022a. URL https://blog.policyengine.org/enhancing-the-current-population-survey-for-policy-analysis-9a7d8e405daa.

M. Ghenis. Enhancing the Current Population Survey for Policy Analysis. *PolicyEngine Blog*, 12 2022b. URL https://blog.policyengine.org/enhancing-the-current-population-survey-for-policy-analysis-9a7d8e405daa.

M. Ghenis. Farness: Forecasting as a Harness for Decision-Making. https://github.com/MaxGhenis/farness, 2024. URL https://github.com/MaxGhenis/farness. Personal futarchy: vote on values (KPIs), bet on beliefs (forecasts with CIs).

M. Ghenis and N. Woodruff. Policyengine's 2021 Year in Review. *PolicyEngine Blog*, 12 2021. URL https://blog.policyengine.org/policyengines-2021-year-in-review-cfb4893ecf2e.

M. Ghenis and N. Woodruff. Policyengine's 2022 Year in Review. *PolicyEngine Blog*, 12 2022. URL https://blog.policyengine.org/policyengine-2022-year-in-review.

M. Ghenis and N. Woodruff. How Would Reforms Affect Cliffs? *PolicyEngine Blog*, 1 2023a. URL https://blog.policyengine.org/how-would-reforms-affect-cliffs-1338055e2648.

M. Ghenis and N. Woodruff. Automate Policy Analysis with PolicyEngine's New ChatGPT Integration. *PolicyEngine Blog*, 3 2023b. URL https://blog.policyengine.org/automate-policy-analysis-with-policy-engines-new-chatgpt-integration.

Good Judgment Inc. Superforecasters Outperform Markets on Fed Predictions. https://goodjudgment.com/, 2024. URL `https://goodjudgment.com/`. Financial Times reported 30% more accurate than futures markets in 2024-2025.

R. Hanson. Futarchy: Vote Values, But Bet Beliefs. http://hanson.gmu.edu/futarchy.html, 2000. URL `http://hanson.gmu.edu/futarchy.html`.

R. Hanson. Shall We Vote on Values, But Bet on Beliefs? *Journal of Political Philosophy*, 21(2): 151–178, 2013. doi: 10.1111/jopp.12008.

M. Holmer. Martin Holmer Profile. https://martinholmer.github.io/, 2024. URL `https://martinholmer.github.io/`. PhD MIT, lead developer Tax-Calculator 2015-2019, Policy Simulation Group president.

A. Inayatali. The Accuracy of Historical CBO Debt Projections, 2023. URL `https://econ.berkeley.edu/sites/default/files/Inayatali_Ammar_Thesis.pdf`.

Institute for Fiscal Studies. History of the IFS. https://ifs.org.uk/about/history-ifs, 2024. URL `https://ifs.org.uk/about/history-ifs`. Accessed December 2024.

Joint Committee on Taxation. History of the Joint Committee on Taxation. https://www.jct.gov/about-us/history/, 2024a. URL `https://www.jct.gov/about-us/history/`. Accessed December 2024.

Joint Committee on Taxation. Revenue Estimating. https://www.jct.gov/operations/revenue-estimating/, 2024b. URL `https://www.jct.gov/operations/revenue-estimating/`. Accessed December 2024.

E. Kiely. Cbo's Obamacare Predictions: How Accurate? *FactCheck.org*, 3 2017. URL `https://www.factcheck.org/2017/03/cbos-obamacare-predictions-how-accurate/`.

J. Li and C. O'Donoghue. A Survey of Dynamic Microsimulation Models: Uses, Model Structure and Methodology. *International Journal of Microsimulation*, 6(2):3–55, 2013. URL `https://microsimulation.pub/articles/00082`.

Q. Mei, Y. Xie, W. Yuan, and M. O. Jackson. A Turing Test of Whether AI Chatbots Are Behaviorally Similar to Humans. *Proceedings of the National Academy of Sciences*, 121(9), 2024. doi: 10.1073/pnas.2313925121.

B. D. Meyer, W. K. C. Mok, and J. X. Sullivan. Household Surveys in Crisis. Working Paper 21399, National Bureau of Economic Research, 2015. URL `https://www.nber.org/papers/w21399`. Documents 40-50% underreporting of SNAP receipt in CPS.

J. Mohun and A. Roberts. Cracking the Code: Rulemaking for Humans and Machines. techreport, OECD Observatory of Public Sector Innovation, 2020. URL `https://oecd-opsi.org/publications/cracking-the-code/`.

New Zealand Digital Government. Turning the Rules of Government into Code Using OpenFisca. https://www.digital.govt.nz/blog/turning-the-rules-of-government-into-code-using-openfisca, 2018. URL `https://www.digital.govt.nz/blog/turning-the-rules-of-government-into-code-using-openfisca`. Three-week experiment rewriting legislation as code; Estonia CIO called it "the most transformative idea".

Northumbria University. Professor Howard Reed. https://www.northumbria.ac.uk/about-us/our-staff/r/howard-reed/, 2024. URL `https://www.northumbria.ac.uk/about-us/our-staff/r/howard-reed/`. Accessed December 2024.

OpenFisca. About OpenFisca. https://openfisca.org/en/about/, 2024. URL `https://openfisca.org/en/about/`. Deployed on four continents: France, UK, Australia, Canada, Tunisia, Senegal, New Zealand.

G. H. Orcutt. A New Type of Socio-Economic System. *Review of Economics and Statistics*, 39(2): 116–123, 1957. doi: 10.2307/1928528. Republished in International Journal of Microsimulation 1(1): 3-9. Available at https://microsimulation.pub/articles/00002.

G. H. Orcutt, M. Greenberger, J. Korbel, and A. M. Rivlin. *Microanalysis of Socioeconomic Systems: A Simulation Study*. Harper & Brothers, New York, 1961.

Penn Wharton Budget Model. The Tax Cuts and Jobs Act, as Reported by Conference Committee (12/15/17): Static and Dynamic Effects on the Budget and the Economy. Technical report, University of Pennsylvania, 12 2017. URL https://budgetmodel.wharton.upenn.edu/issues/2017/12/18/the-tax-cuts-and-jobs-act-reported-by-conference-committee-121517-preliminary-static-and-dy

Penn Wharton Budget Model. Comparing PWBM Projections to CBO Baselines and Actual Outcomes. https://budgetmodel.wharton.upenn.edu/, 2024. URL https://budgetmodel.wharton.upenn.edu/. Accessed December 2024.

PolicyEngine. Estimating Your Supplemental Security Income Benefits in PolicyEngine. *PolicyEngine Blog*, 2022. URL https://blog.policyengine.org/estimating-your-supplemental-security-income-benefits-in-policyengine-74c6396ee402.

PolicyEngine. Breaking Down US Poverty Impacts by Sex. *PolicyEngine Blog*, 2023. URL https://blog.policyengine.org/breaking-down-us-poverty-impacts-by-sex.

PolicyEngine. About PolicyEngine. https://policyengine.org/us/about, 2024a. URL https://policyengine.org/us/about. Accessed December 2024.

PolicyEngine. Policyengine GitHub Organization. https://github.com/PolicyEngine, 2024b. URL https://github.com/PolicyEngine. Accessed December 2024.

PolicyEngine. Multi-Agent Workflows for Policy Research. *PolicyEngine Blog*, 2024c. URL https://blog.policyengine.org/multi-agent-workflows-policy-research.

PolicyEngine. Policyengine UK Model Validation. https://policyengine.github.io/openfisca-uk/model/validation.html, 2024d. URL https://policyengine.github.io/openfisca-uk/model/validation.html. Accessed December 2024.

Prabook World Biographical Encyclopedia. Guy Henderson Orcutt. https://prabook.com/web/guy_henderson.orcutt/696679, 2024. URL https://prabook.com/web/guy_henderson.orcutt/696679. July 5, 1917 – March 5, 2006; American econometrician.

Quantified Uncertainty Research Institute. Squiggle: A Language for Probabilistic Estimation. https://www.squiggle-language.com/, 2024. URL https://www.squiggle-language.com/. Accessed December 2024.

M. Richiardi, D. Collado, and D. Popova. Ukmod – A New Tax-Benefit Model for the Four Nations of the UK. *International Journal of Microsimulation*, 14(1):92–101, 2021. URL https://microsimulation.pub/articles/00231.

G. Sadowsky. *Computing Technology and Microsimulation*, chapter 7. National Academies Press, Washington, D.C., 1991. URL https://nap.nationalacademies.org/read/1835/chapter/12.

M. Sarstedt, S. J. Adler, C. M. Ringle, G. Cho, A. Diamantopoulos, H. Hwang, and B. D. Liengaard. Using Large Language Models to Generate Silicon Samples in Consumer and Marketing Research: Challenges, Opportunities, and Guidelines. *Psychology & Marketing*, 2024. doi: 10.1002/mar.21982. Silicon sampling promising for pretesting/pilot studies; important limitations for main studies.

Society of Actuaries. Chapter 3: Dynasim. techreport, Society of Actuaries, 1997. URL https://www.soa.org/493824/globalassets/assets/files/research/projects/chapter_3.pdf.

H. Sutherland and F. Figari. Euromod: the European Union tax-benefit microsimulation model. *International Journal of Microsimulation*, 6(1):4–26, 2013. URL `https://microsimulation.pub/articles/00075`.

Tax Policy Center. The Urban-Brookings Tax Policy Center Microsimulation Model. https://www.urban.org/research/publication/urban-brookings-tax-policy-center-microsimulation-model, 2024. URL `https://www.urban.org/research/publication/urban-brookings-tax-policy-center-microsimulation-model`. Accessed December 2024.

P. E. Tetlock. Expert Political Judgment: How Good Is It? How Can We Know? 2005.

P. E. Tetlock and D. Gardner. *Superforecasting: The Art and Science of Prediction*. Crown, 2015. ISBN 978-0804136716.

M. Torry. Static Microsimulation Research on Citizen's Basic Income for the UK: A Personal Summary and Further Reflections. *EUROMOD Working Paper*, (EM13/19), 2019.

UBI Center. Introducing PolicyEngine UK. https://www.ubicenter.org/introducing-policyengine, 9 2021. URL `https://www.ubicenter.org/introducing-policyengine`.

Urban Institute. Karen E. Smith Profile. https://www.urban.org/author/karen-e-smith, 2024. URL `https://www.urban.org/author/karen-e-smith`. Accessed December 2024.

U.S. Bureau of Labor Statistics. 2024 Research Supplemental Poverty Measure Thresholds. https://www.bls.gov/pir/spm/spm_thresholds_2024.htm, 2024. URL `https://www.bls.gov/pir/spm/spm_thresholds_2024.htm`. Two adults, two children: $39,430 (renters), $39,068 (owners with mortgage).

U.S. Census Bureau. Current Population Survey (CPS). https://www.census.gov/programs-surveys/cps.html, 2024. URL `https://www.census.gov/programs-surveys/cps.html`. Monthly survey of approximately 60,000 U.S. households.

Various. Polymarket 2024 Election Forecasting Performance. 2024. Trump at 58% vs polls showing coin flip; correctly predicted outcome.

J. Wang and others. Taxagent: How Large Language Model Designs Fiscal Policy. https://arxiv.org/abs/2506.02838, 2025. URL `https://arxiv.org/abs/2506.02838`. LLM-based agents simulate household behavior while TaxAgent optimizes tax rates over 120-month periods.

H. W. Watts. Distinguished Fellow: An Appreciation of Guy Orcutt. *Journal of Economic Perspectives*, 5(1):171–179, 1991. doi: 10.1257/jep.5.1.171. URL `https://www.aeaweb.org/articles?id=10.1257/jep.5.1.171`.

J. Wolfers and E. Zitzewitz. Prediction Markets for Economic Forecasting. Working Paper 18222, National Bureau of Economic Research, 2012. URL `https://www.nber.org/papers/w18222`.

N. Woodruff. Digital Public Goods Alliance Recognizes PolicyEngine as a Digital Public Good. *PolicyEngine Blog*, 4 2023a. URL `https://blog.policyengine.org/digital-public-goods-alliance-recognizes-policyengine-as-a-digital-public-good-555f3db85314`

N. Woodruff. From Idea to Impact: Scoring a Policy Reform on the New PolicyEngine US. *PolicyEngine Blog*, 2023b. URL `https://blog.policyengine.org/from-idea-to-impact-scoring-a-policy-reform-on-the-new-policyengine-us`.

N. Woodruff. Ai-Powered Explanations for Tax and Benefit Calculations. *PolicyEngine Blog*, 2024. URL `https://blog.policyengine.org/us-household-ai`.

S. Zheng, A. Trott, S. Srinivasa, D. C. Parkes, and R. Socher. The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18):eabk2607, 2022. doi: 10.1126/sciadv.abk2607. URL `https://www.science.org/doi/10.1126/sciadv.abk2607`. AI-designed tax policy outperformed Saez framework by 16% on equality-productivity tradeoff.