Exact rules, estimated aggregates: the PolicyEngine tax–benefit microsimulation in an open model suite*

Vahid Ahmadi†

November 2026

Abstract

This paper documents the PolicyEngine tax–benefit microsimulation as the lead member of the *PolicyEngine Macro* suite: the engine that encodes UK and US statutory tax and benefit rules as dated, parameterised code and resolves them at household resolution. Its central claim is a distinction the suite treats as load-bearing and refuses to blur. A household calculation is *exact arithmetic*: it is a deterministic function of inputs the analyst types, it consults no survey and no weights, and every figure in it can be reproduced by hand—a single UK adult earning £50,000 in 2026 has £37,430 of basic-rate income after the £12,570 personal allowance, pays £7,486 of income tax and £2,994 of National Insurance, and keeps £39,520; a five-point rise in the basic rate adds exactly £1,872. A population estimate is not of that kind: it is a weighted sum of exact household arithmetic over an *estimated* population, and it inherits sampling, imputation, reweighting and ageing error from the enhanced Family Resources Survey (53,508 UK households in the 2026 vintage of `enhanced_frs_2023_24`) or the US Current Population Survey. The capability registry states the two separately—"none for household arithmetic; survey/calibration uncertainty for population estimates"—and this paper explains why that sentence has a colon in it. Validation follows the same split: the rules are checked against implemented legislation rather than against a forecast, and the one committed external benchmark is the static costing of a 1p rise in the UK basic rate, £6.46bn in 2026 rising to £7.38bn by 2030, against HMRC's June 2025 ready reckoner of £6.9bn in 2026–27 rising to £8.2bn by 2028–29. That comparison is reported as a benchmark and not as a validation: the costing is *below* the official figure at every year that can be matched—−6.4 per cent in 2026–27, widening to −15.6 per cent on the interpolated 2028–29 comparison—the reckoner is post-behavioural where this costing is static, and the £7.38bn endpoint has no year-matched official counterpart and no scored-run artifact behind it anywhere in the project. The model cannot answer GDP, inflation, interest rates or macro feedback, and the paper is explicit about what the suite's bridges do and do not supply in their place: the route into the OBR emulator is a demand-side approximation that injects the whole costing as a held add-factor on nominal household disposable income, discarding the very incidence the microsimulation just computed, and it is not a general reform-incidence macro model. No confidence interval is reported for any population estimate anywhere in this paper, because the codebase does not produce one.

*__Acknowledgements:__ I thank Max Ghenis for his great suggestions and helpful discussions. This paper is a PolicyEngine working paper describing the PolicyEngine tax–benefit microsimulation *as it is used by* the PolicyEngine Macro suite; the country rule packages it documents are maintained upstream by the wider PolicyEngine project and its contributors, and this paper is a description of that work in use rather than an audit of it. Prepared with AI assistance (Claude, Anthropic). All remaining errors are my own.

†Research Associate at PolicyEngine. Email: vahid@policyengine.org.

Keywords: microsimulation, tax–benefit models, policy costings, survey microdata, open source, reproducibility

1 Introduction

Two questions follow a change to a tax or a benefit, and they are not the same question. The first is arithmetic: given this family, these earnings, these children, this year, what does the rule book say they pay and receive, before and after? The second is statistical: across a whole country, how much does the change raise, and who gains and loses? The first has an exact answer, because tax and benefit law is a computable function of a household's circumstances and the answer can be checked with a pencil. The second does not, because nobody observes the whole country: it is estimated from a survey that is sampled, imputed, reweighted and aged forward, and every one of those steps puts error into the headline number.

This paper documents the PolicyEngine tax–benefit microsimulation—the engine behind `https://policyengine.org`, and the lead member of the *PolicyEngine Macro* suite—as this suite uses it, and it treats the distinction between those two kinds of number as its subject.[1] The microsimulation is the suite's first model, and the capability registry records its status as "exact for implemented rules (household arithmetic); population estimates add survey error." It is also the only member covering both the United Kingdom and the United States, and the only one whose evidence base is implemented legislation rather than another institution's published output. Every bridge in the suite starts or ends here: it supplies the static costing that the OBR macroeconometric emulator turns into second-round demand effects (Ahmadi, 2026a), the tax functions that the overlapping-generations member estimates its household problem from (Ahmadi, 2026b), and the incidence engine through which macro paths from the other members are pushed back down to households.

The contribution is threefold. First, *a documented exactness boundary.* The suite's capability registry records this model's uncertainty as "none for household arithmetic; survey/calibration uncertainty for population estimates," and Section 5 sets out precisely why the two clauses are different in kind rather than in degree: the arithmetic inside a population aggregate is exactly as exact as a household calculation, and all of the uncertainty lives in the weights and in the imputed components of the inputs. Blurring the two—quoting a population costing with the confidence one is entitled to have in a household calculation—is the characteristic misuse of a microsimulation model, and the boundary is worth stating in a document rather than in a docstring.

Second, *a validation posture for a model that does not forecast.* There is no forecast error to report here, because nothing is being forecast: published rules are applied, and the check is whether they were transcribed correctly. Section 6 states what that check does and does not establish, and reports the single committed external benchmark—the static costing of 1p on the UK basic rate against HMRC's published ready reckoner—with its actual numbers and its actual gap, including the fact that it is one reform, in one country, on one tax, and that the official figure it is measured against is itself an estimate on a different data basis.

Third, *a declared account of what the model cannot see and what the suite substitutes.* The registry's `cannot_answer` list is short and absolute: GDP, inflation, interest rates, macro feedback. Section 7 describes the machinery that supplies those, and is equally explicit about that machinery's limits—in particular that the route into the OBR emulator injects an entire annual costing as a held add-factor on nominal household disposable income, flat within the year,

[1] `https://policyengine-macro.vercel.app/pe/`. PolicyEngine Macro is a suite of open economic models—microsimulation, macroeconometric, structural-VAR, heterogeneous-agent, overlapping-generations and ecological stock-flow—that score the same PolicyEngine reform object and report comparable real-world aggregates (PolicyEngine, 2025).

and is therefore a demand-side approximation rather than a general model of reform incidence. The macro model never learns which households paid.

Two boundaries on the paper itself should be stated at the outset. This is a description of the model *as the suite consumes it*: the country rule packages (`policyengine-uk`, `policyengine-us`) and the simulation framework beneath them (`policyengine-core`, forked from OpenFisca) are maintained upstream, and nothing here audits how completely they cover UK or US statute or how thorough their own test suites are. And every figure quoted below comes from an artifact committed to the PolicyEngine Macro repository—the capability registry, the integration adapters, the committed chart data behind the site's validation figures, or the sister working papers—with the source named at the point of use; no number in this paper was produced for it.

The rest of the paper proceeds as follows. Section 2 places the model in the microsimulation tradition and in the open-model literature. Section 3 sets out what the model computes: variables and dated parameters as a dependency graph, the country-specific entity structures, and the reform object. Section 4 describes the integration layer this suite drives it through. Section 5 is the paper's central section, on the boundary between exact arithmetic and estimated aggregates. Section 6 gives the validation posture and the ready-reckoner benchmark; Section 7 the limits and the bridges; Section 8 reproducibility, data vintage and the gated microdata; Section 9 limitations, and Section 10 concludes. Appendices give the curated reform-parameter catalogue, the worked household arithmetic, and the reform-object grammar with its refusals.

2 Related literature

The model sits in three literatures: the tax–benefit microsimulation tradition that begins with Orcutt (1957), the survey-data literature on under-reporting and calibration that determines how far a population estimate can be trusted, and the open-source and reproducibility movement that determines whether anyone outside the producing institution can check it.

2.1 Tax–benefit microsimulation

Orcutt (1957) proposed simulating a population as a collection of individual units whose behaviour and circumstances are represented explicitly, rather than as aggregate relationships between averages; the tax–benefit branch of that programme applies statutory rules to a representative sample of households and aggregates the results. The survey and handbook literature—Mitton et al. (2000), Bourguignon and Spadaro (2006), Figari et al. (2015)—sets out the standard architecture: a microdata file of households, a rule engine encoding the tax and transfer system for a given policy year, and a differencing procedure that runs the same data under baseline and reform parameters and reports the change in revenue, in net incomes, and in their distribution.

Every major fiscal institution operates a model of this class. In the United Kingdom, the Institute for Fiscal Studies' TAXBEN has been the standard academic instrument for four decades, and EUROMOD extended the design to a harmonised multi-country platform whose openness of documentation—if not of data—set the transparency benchmark for the field (Sutherland and Figari, 2013). In the United States, the NBER's TAXSIM has provided federal and state tax calculations to researchers since the 1970s (Feenberg and Coutts, 1993), and the Policy Simulation Library's Tax-Calculator provides an openly developed federal model (Policy Simulation Library, 2025). The revenue departments and fiscal watchdogs run their own: HMRC's costings,

from which the ready reckoner used in Section 6 is produced, rest on administrative Survey of Personal Incomes data rather than a household survey (HM Revenue and Customs, 2025).

PolicyEngine's distinguishing choices within this tradition are the ones that matter for this paper. The rule engine is open source and so is the model of statute: `policyengine-core` is a fork of the OpenFisca framework (OpenFisca, 2025), in which every quantity—an allowance, a taper, a means test—is a *variable* with a formula and a set of dated *parameters*, and a reform is a re-dating of a parameter rather than an edit to code. And the same object serves both levels: the identical reform dictionary that scores one typed household scores the whole survey, which is why the exactness question of Section 5 arises so sharply—the arithmetic is literally the same code in both cases, and only the inputs differ in status.

2.2 Survey microdata, under-reporting, and calibration

The credibility of a population estimate rests on the microdata, and the microdata literature is not reassuring about using a household survey raw. Meyer et al. (2015) document a broad and worsening under-reporting of transfer receipt in US household surveys, with major programmes capturing well under the administratively recorded totals; Brewer et al. (2017) show for the UK Family Resources Survey that the households reporting the lowest incomes are conspicuously better off on expenditure measures than their reported incomes imply, so the bottom of the reported income distribution is substantially a measurement artefact. Both findings bear directly on a distributional costing: the decile a household is assigned to, and the benefit entitlement computed for it, depend on income components the survey measures with error.

The standard responses are imputation and calibration. Reweighting a survey so that its weighted totals reproduce known administrative or national-accounts aggregates is the calibration-estimator problem of Deville and Särndal (1992), and static ageing—uprating incomes and re-weighting a survey of one year to represent a later policy year—is routine in the microsimulation literature (Figari et al., 2015). PolicyEngine's UK data pipeline does both: the *enhanced* Family Resources Survey (Department for Work and Pensions, 2025) imputes under-reported incomes and reweights the survey so aggregates better match administrative totals, and the per-year datasets used for population runs are built by uprating to the policy year. This is what makes population estimates usable; it is also, precisely, what makes them estimates. A reweighted, imputed, aged survey is a model of the population, and its errors are not the sampling error of the original design alone.

2.3 Open models and reproducibility

The third literature is about who can check the answer. Systematic replication attempts find that published economic results frequently cannot be reproduced from the authors' own materials (Chang and Li, 2015), and the best-known failure in the fiscal domain—the spreadsheet error in Reinhart and Rogoff's debt-threshold result, found only when the original file was obtained and re-run (Herndon et al., 2014)—is a standing argument that consequential numbers should be recomputable by outsiders. Official costings are the strongest case: they move budgets, and the models producing them are typically not runnable outside the producing department.

An open microsimulation answers part of that objection, and only part. The rules are public code and the household calculation is reproducible by anyone with a laptop and no credentials at all—which is an unusual property in fiscal analysis, and the reason Section 6's worked example can be checked by hand. The population side is not fully open in the same way:

the UK microdata is distributed through a gated repository requiring a token, so a reader can reproduce this paper's household arithmetic exactly and cannot reproduce its £6.46bn costing without being granted access. Section 8 states that asymmetry rather than eliding it. The same tension—rules open, microdata restricted—runs through the whole field, since the underlying surveys are released under licence by national statistical agencies.

Finally, the multi-model design this paper's model leads is itself a position in the modelling literature: rather than extend one model to answer every question, the suite scores the same reform through models of different classes and publishes a *verification gradient* recording how far each one's outputs can be audited against ground truth, in the tradition of institutional model suites (Burgess et al., 2013). The microsimulation occupies the top of that gradient—"checked against implemented legislation"—and the sister papers (Ahmadi, 2026a,b) document the members below it.

3 What the model computes

3.1 Statute as a dependency graph

The model represents a tax and benefit system as a directed graph of *variables*. Every quantity—an allowance, a rate, a taper, a means test, an entitlement, a liability—is a variable carrying a formula and a set of *dated parameters*, so that the rule in force on a given date is the one whose start date is the latest date not after it. Asking the model for a variable resolves the graph beneath it: a request for household net income is answered by evaluating the statutory chain that produces it, not by applying an approximating schedule fitted to that chain. This is the OpenFisca architecture that `policyengine-core` inherits (OpenFisca, 2025), and it is what makes a reform expressible as data: changing a rate is re-dating a parameter and re-resolving the same graph, with no code path specific to the reform.

Four packages are involved, and the division matters when reading provenance. `policyengine-core` is the simulation framework; `policyengine-uk` and `policyengine-us` encode the statutory rules of their respective countries; and `policyengine` is the analyst-facing package this suite consumes, adding dataset management, reform application, structured outputs and version-pinned release bundles. The suite's own integration layer, `policyengine-macro`, sits above all four and is the subject of Section 4.

3.2 Entities

Variables attach to *entities*, and the entity structures differ by country because the two systems assess entitlement on differently drawn units. The UK model has three: person, benefit unit, and household. The US model has six: person, tax unit, SPM unit, family, marital unit, and household. The asymmetry is not an implementation artefact—US programmes genuinely assess on different groupings of the same people, so a single household can be one tax unit for the federal income tax and a differently drawn SPM unit for poverty measurement—and it propagates into the adapters, which return a UK result with a `benunit` block and a US result with `tax_unit` and `spm_unit` blocks.

Country coverage as the suite advertises it is: for the UK, income tax, National Insurance, Universal Credit, Child Benefit and Pension Credit among others; for the US, federal and state income tax, payroll tax, SNAP, TANF, EITC, CTC, SSI and Social Security. These are the systems the suite exercises, not an exhaustive statement of upstream coverage.

3.3 The household calculation

Write a household as h—the list of people with their ages and incomes, plus whatever entity-level attributes the country requires—and the policy environment as a parameter vector θ_y in force for policy year y. The model computes

$$N(h, \theta_y) \;=\; \text{the resolved value of any requested variable,}$$

a deterministic function. Given the same h, the same y and the same package versions, it returns the same numbers every time; there is no sampling, no draw, no weight, and no estimated coefficient anywhere in the chain. The reform effect on that household is a difference of two such evaluations,

$$\Delta(h) \;=\; N(h, \theta'_y) - N(h, \theta_y), \tag{1}$$

where θ' is θ with the reform's parameters re-dated. This is what `pe_household_impact` computes, and (1) is exactly as exact as its two terms.

A worked instance, which Section 6 uses as the validation case and Appendix B sets out in full: a single UK adult aged 35 with employment income of £50,000 in 2026 has £37,430 of income above the £12,570 personal allowance, all within the basic-rate band, and therefore pays £7,486 of income tax at 20 per cent and £2,994 of employee National Insurance at 8 per cent, leaving £39,520 of household net income on the HBAI concept. Applying the reform `{"gov.hmrc.income_tax.rates.uk[0].rate": 0.25}` raises the basic rate by five points and adds five per cent of £37,430, or £1,872, to the tax bill, reducing net income by the same amount. Every one of those figures is reproducible with a pencil, which is the property the rest of this paper is about.

3.4 The reform object

A reform is a flat dictionary mapping a parameter path to a new value—for example `{"gov.hmrc.income_tax.rates.uk[0].rate": 0.21}` for a one-point rise in the UK basic rate, or `{"gov.irs.credits.ctc.amount.adult_dependent": 1000}` for a US credit change. The same object is accepted at household level, at population level, and by every other model in the suite that scores reforms at all, which is what makes cross-model scoring of a single reform meaningful.

A value may be a bare number, in which case it takes effect from the start of the scored year, or a `{date: value}` mapping with ISO dates for an explicit effective date. The suite's validator normalises both to the second form and refuses everything else with an actionable message rather than a library traceback (Appendix C). One refusal is substantive rather than cosmetic and is worth recording here: a date *range* key is rejected, because the reform API applies a value from an effective date and leaves it in force indefinitely—the end date cannot be expressed at all—so accepting a range would silently score a permanent reform while the caller believed they had scored a temporary one. The validator says so and points at the workaround: score the affected years individually and treat later years as baseline.

The suite ships a curated catalogue of thirteen well-known reform paths (nine UK, four US; Appendix A) whose baseline values, labels and units are resolved from the installed parameter tree at call time rather than transcribed, so a path that ceases to resolve upstream is returned marked `live: false` with an explicit error instead of a silently stale value. Three of the thirteen—the capital-gains paths—carry a standing note that the household calculator does not compute CGT, so those reforms are valid but will not move a household result and must be

scored at population level. That note is a small but exact illustration of this paper's theme: the model tells you where its own arithmetic does not reach.

4 The integration layer

The suite does not fork the microsimulation; it drives the released `policyengine` package through a thin adapter module (`policyengine_macro.core`) whose job is to validate inputs, run the model, and return a payload with declared provenance and a common cross-model score block. Because importing the package pulls in the full UK and US country models—roughly twenty seconds—the import is deliberately lazy, inside the adapters rather than at module level, so that argument validation fails fast even where the model is not installed.

4.1 Household adapters

`pe_household` evaluates one household under a single policy environment and returns the resolved person-, benefit-unit-/tax-unit- and household-level variables together with a headline summary; `pe_household_impact` runs the same household twice, with and without the reform, and returns baseline, reform, and their differences—the $\Delta(h)$ of equation (1). Neither touches any dataset, and neither requires a credential.

Two input guards are worth documenting because both were written in response to failure modes that produce a plausible wrong answer rather than an error. First, `country` is required and has no default on the household tools: the UK and US are different tax–benefit models, and a default would silently score the wrong country for a US household. (`pe_population_impact` retains a `country="uk"` default for backwards compatibility—an asymmetry the code declares in a comment and Section 9 restates as a limitation rather than a feature.) Second, the US household dictionary is validated against the key the model actually reads: PolicyEngine US takes the state as household-level `state_code_str` and falls back to California when it is absent, so a misspelled `state_code` previously returned a confident California answer for a Texas household. The adapter now refuses the wrong key by name, and refuses a state code that is not a two-letter US state or DC.

4.2 The population pipeline

Population scoring runs the same rule graph over representative household microdata. The pipeline is: obtain the dataset for the requested country, year and vintage (downloading or building it on first use); construct and run a baseline simulation; construct and run a counterfactual differing only in the policy; and measure the difference. The baseline simulation is cached in-process per `(country, year, dataset)`, so a session that scores several reforms against the same baseline pays for one baseline and one reform run each thereafter.

The measurement core is deliberately shared between two callers that mean different things. `pe_population_impact` scores a *reform*: policy differs between the two runs. `_pe_population_incidence` scores a *macro shock*: policy is identical on both sides and the counterfactual differs only by an overlay scaling the employment-income inputs, so the budgetary difference it reports is the automatic-stabiliser response to the shock and not the cost of anything. Both report the same outcome block, and the second is careful to say in its own headline that policy was unchanged. When the overlay is absent the shocked run is

construction-identical to the baseline and every delta is zero by design, which callers surface as a reported null rather than an error.

Three measurement choices in that shared core are stated here because they determine what the published aggregates mean.

The budget measure differs by country. For the UK the budgetary impact is the change in `gov_balance`—all modelled taxes, including capital gains tax and employer National Insurance, minus benefit spending—because that is the UK budget headline and it is not in the model's default output set, so the adapter requests it explicitly. For the US it is the change in `household_tax` minus the change in `household_benefits`. These are different concepts on different coverage, and the returned payload names the basis it used in a `budgetary_impact_basis` field rather than leaving the reader to assume they are comparable.

Decile effects are means, not means of ratios. Distributional output groups households by baseline income decile and reports the change in each decile's mean net income, with the relative figure computed as the change in the decile's mean expressed as a percentage of that mean. The obvious alternative—averaging each household's own percentage change—is dominated by households with near-zero baseline income, for whom a small absolute change is an enormous ratio, and is not used.

Winner and loser counts equal the sum of the rows. The headline counts accumulate exactly the integers displayed in the decile rows. Truncating the underlying floats independently at each level had let the headline disagree with the table beneath it.

4.3 Provenance and the common score block

Every adapter returns a provenance record naming the model identifier, the installed distribution and its version, the source repository, the data vintage, and the baseline definition, together with an instruction for reproducing the run—check out the recorded source at the recorded revision, install the recorded adapter version, and rerun the serialised inputs. For population runs the data vintage *is* the dataset name, which is how the registry's "recorded per run" vintage policy is implemented (Section 8).

Population scores additionally carry a `ScoreResult` block in the schema shared across the suite, so that the same reform scored through the microsimulation, the OBR emulator and the OLG model can be compared field by field. Three of its fields are load-bearing for this paper's argument. `assumptions` always contains "static microsimulation: no behavioural or macro feedback" and the dataset name with its household count. `caveats` always contains a statement that GDP, consumption and investment are out of scope for a static microsimulation and only the budgetary impact is filled—so a consumer reading the common schema cannot mistake an empty GDP field for a zero effect. And the revenue quantity's `uncertainty` field reads, verbatim, "not estimated in this result." That is the machine-readable form of Section 5's central admission.

4.4 Hosted surfaces

The model is reachable three ways: a hosted Model Context Protocol server exposing `calculate_household`, `household_reform_impact`, `population_reform_impact`, `list_reform_parameters`

and `score_reform` with `model="microsim"`; the `pe-macro` command-line interface mirroring the same tools; and the Python package directly. Runtime is sub-second for a household and minutes for a population run. On the hosted server the gated UK microdata credential is provisioned server-side, so a hosted population score requires nothing of the caller—which is convenient, and is also the reason Section 8 distinguishes between a result being *obtainable* and a result being *reproducible by the reader*.

5 Exact arithmetic and estimated aggregates

The suite's capability registry records this model's uncertainty in one sentence with a semicolon in it:

> *none for household arithmetic; survey/calibration uncertainty for population estimates.*

That sentence is the central claim of this paper, and the purpose of this section is to establish that its two clauses describe different kinds of object rather than two points on a scale of confidence. Getting this wrong in either direction has a cost. Treating a household calculation as an estimate invites spurious hedging about a number that can be checked with a pencil. Treating a population costing as exact—quoting £6.46 billion to the nearest ten million and defending the second decimal—attributes to a survey-based aggregate a precision that no part of its construction supports.

5.1 Why the household number is exact

Section 3.3 defined the household calculation as $N(h, \theta_y)$, a deterministic resolution of the statutory dependency graph. Three properties follow, and they are worth stating separately because each rules out a distinct source of error.

There is no sampling. The household h is not drawn from anything: it is the household the analyst described. Nothing is inferred about a wider population, so no sampling distribution exists to have a variance.

There are no weights and no imputation. Every input is supplied directly. The model does not fill in an unreported income component, does not adjust for the tendency of survey respondents to under-report a benefit, and does not scale the result to represent anyone else.

There are no estimated coefficients. The graph contains statutory rates, thresholds and formulae with effective dates, not fitted parameters. Nothing in the chain from £50,000 of employment income to £7,486 of income tax was estimated from data; every step is a rule.

What remains is not uncertainty in the statistical sense but two auditable questions. Do the encoded rules match the statute they claim to encode? And is the household described the household intended—the right entity structure, the right year, the right state? Both are answerable by inspection, neither is random, and neither is reduced by collecting more data. This is why the registry says "none" rather than a small number: a household result carries no error *distribution*, only the possibility of a mistake, which is a different thing and is addressed by testing rules against legislation (Section 6) rather than by reporting an interval.

5.2 Why the population number is not

A population estimate has the form

$$\widehat{R}(\theta) \;=\; \sum_{i=1}^{n} w_i \, b_i(x_i, \theta), \tag{2}$$

where i indexes the households in the microdata file, b_i is the household-level quantity being aggregated—a tax liability, a benefit entitlement, the government balance—computed by exactly the code of Section 3.3, x_i is that household's recorded circumstances, and w_i is its survey weight. The reform estimate is the difference of two such sums under θ' and θ.

The decisive observation is where the uncertainty is and is not. It is *not* in b_i: given x_i, the arithmetic in (2) is exactly as exact as a household calculation, because it is the same function. All of it lives in w_i and in x_i, and it arrives through four channels that compound rather than cancel.

Sampling. The Family Resources Survey and the Current Population Survey are samples. A different draw of households would give a different $\widehat{R}$, with a variance determined by the survey design—clustering and stratification included, which is why a naive independent-sample standard error would understate it even if one were computed.

Measurement and imputation. Survey respondents under-report income and transfer receipt, systematically and by large margins in the cases the literature has measured (Meyer et al., 2015; Brewer et al., 2017). The UK enhancement responds by imputing under-reported incomes—so some components of x_i are not reports at all but model outputs, and the tax computed on them inherits whatever error the imputation carries.

Calibration. The enhanced survey is reweighted so that its weighted aggregates better match administrative totals (Deville and Särndal, 1992). This makes the headline aggregates far more usable and simultaneously makes w_i a fitted object: the weights are chosen to hit targets, and a reform whose incidence is concentrated in a subpopulation the calibration did not target is scored on weights that were never asked to be right for it.

Ageing. The UK population runs used by this suite are built from `enhanced_frs_2023_24` and evaluated for a later policy year—2026 in every worked example here. Incomes are uprated and the file is aged forward; the resulting population is a projection of the survey, not an observation of the policy year.

Any of the four alone would make $\widehat{R}$ an estimate. Together they make it an estimate whose error is dominated by data construction rather than by the statutory arithmetic, and this ordering is the practically important part: improving the rule engine does not improve a population costing much, and improving the microdata does.

5.3 What follows, and what this paper does not claim

Three consequences are enforced structurally in the codebase rather than left to the reader's discipline.

Different access requirements. A household calculation needs no data and no credential; a population run needs a gated dataset whose name is recorded in the result's provenance. The boundary is visible in what you have to obtain in order to cross it.

Different declared status. The registry states the two clauses separately; the common score block attaches "static microsimulation: no behavioural or macro feedback" and the dataset name with its household count to every population score; and the suite's verification gradient classifies the model as "checked against implemented legislation" with the headline "household maths exact; population adds survey error."

Different rounding discipline in this paper. Household figures are quoted to the pound because they are exact to the pound. Population figures are quoted as the committed artifacts record them and are discussed to a tenth of a billion at best, with the ready-reckoner comparison of Section 6 framed in percentage deviations rather than as agreement to a decimal place.

And now the admission that the rest of the paper depends on. *No interval is reported for any population estimate here, because the codebase does not produce one.* The uncertainty is documented as a string; the score block's uncertainty field for the revenue quantity reads "not estimated in this result"; and nothing in the pipeline computes a replicate-weight or bootstrap standard error, propagates imputation variance, or reports a range around £6.46 billion. A reader who wants to know how wide that number's distribution is will not find the answer in this paper or in the artifacts behind it. Section 9 records this as the model's first limitation and the natural next piece of work; describing an uncertainty is not the same as quantifying it, and the distinction this section draws would be worth more if the second clause had a number in it.

There is one further point about the boundary itself. Nothing forces a caller to read the registry string: the adapters return the declaration, but a consumer that ignores it will receive a population figure formatted exactly like a household figure, in the same units, from the same package. The boundary is documented and structurally visible, not mechanically enforced.

6 Validation

6.1 The posture: no forecast error to report

Every other member of the PolicyEngine Macro suite is compared against something a modelling institution published: the OBR emulator against the *Economic and Fiscal Outlook* and HMRC's ready reckoner (Ahmadi, 2026a), the structural VAR against the paper it replicates, the FRB/US implementation against the Federal Reserve's own database and reference solver. The microsimulation is the exception, and the reason is structural: it is not forecasting anything. There is no forecast error to report because no forecast is made. Published rules are applied to described circumstances, and the question validation has to answer is whether the rules were transcribed and resolved correctly.

The suite's verification gradient therefore places this model in its own class—"checked against implemented legislation"—with the headline "household maths exact; population adds survey error." That is a stronger epistemic position than a calibrated counterfactual and a different one from a replication with a published anchor, and it comes with its own characteristic failure mode: a rule can be correctly implemented and simply absent, and no amount of arithmetic checking will reveal a provision nobody encoded.

6.2 Rules against statute, checked by hand

The primary check is that a result is reproducible without the model. For the single UK adult of Section 3.3, earning £50,000 in 2026:

- £50,000 less the £12,570 personal allowance leaves £37,430 of taxable income, all inside the basic-rate band;
- at 20 per cent that is £7,486 of income tax;
- employee National Insurance at 8 per cent on the corresponding earnings is £2,994;
- household net income on the HBAI concept is £39,520;
- raising the basic rate by five points adds five per cent of £37,430, which is £1,872, and reduces net income by the same amount.

Every figure is deterministic arithmetic from the implemented rules, and the reader can check the whole chain against the published allowance and rate schedule. What this establishes is bounded and worth stating precisely, in the same words the project's own validation page uses: it verifies *those rules*, not that every area of legislation has complete test coverage. One hand-checkable case demonstrates that the resolution machinery does what it claims for the provisions it exercises. It says nothing about the parts of the statute book it does not touch, and this paper makes no claim about upstream's coverage of them (Section 9).

6.3 The static costing against HMRC's ready reckoner

The one committed external benchmark for the population side is the canonical UK illustrative change: one penny on the basic rate of income tax. HMRC publishes the official ready reckoner—*Direct effects of illustrative tax changes*, produced on the OBR-certified costing basis—which in its June 2025 vintage puts a 1p change in the basic rate at £6.9 billion in 2026–27, rising to about £8.2 billion a year in 2027–28 and 2028–29 (HM Revenue and Customs, 2025). That vintage stops at 2028–29. The same reform scored through the microsimulation on the enhanced FRS is reported at £6.46 billion in 2026, rising to £7.38 billion by 2030—though the two endpoints do not have the same standing behind them, which Section 6.4 sets out before the gap is interpreted. Table 1 sets the two side by side.

The direction should be stated before the table is read, because an earlier version of this comparison got it wrong. *Every microsimulation year is below every HMRC year.* There is no year in which the two brackets overlap and no reading on which the costing sits "inside" the official range: it sits under it throughout, by 6.4 per cent in the one year that is genuinely matched and by more thereafter.

6.4 What stands behind the two endpoints

The provenance of the pair is weaker than a table of four significant figures suggests, and saying so is part of the benchmark.

The £6.46bn 2026 figure is the one the pipeline can in principle produce: `integration/tests/test_populati` scores exactly this reform through `pe_population_impact` for 2026 and asserts the result. That test is still skipped unless a Hugging Face token for the gated microdata is present, and it writes no artifact, so it does not run in ordinary CI. Its assertion band was $4.0 < x < 9.0$ £bn—wide

Table 1: Static costing of 1p on the UK basic rate, microsimulation versus HMRC's ready reckoner

£bn per year	Microsimulation	HMRC	Deviation
2026–27	6.46	6.9	−6.4%
2028–29†	6.92	8.2	−15.6%
2030 (end of scored window)‡	7.38	≈8.2	−10.0%

Notes: Reform `{"gov.hmrc.income_tax.rates.uk[0].rate": 0.21}`, scored on the enhanced FRS. Only the first row is year-matched and independently scored on both sides. †The 2028–29 microsimulation figure is a linear interpolation between the two scored endpoints, not an independently scored year. ‡The 2030 row has *no* year-matched counterpart: the June 2025 reckoner stops at 2028–29, so the ≈8.2 it is set against is a 2028–29 figure held flat two years forward by us, and the −10.0 per cent is a comparison across a two-year horizon gap rather than a deviation. Official figures: HMRC, *Direct effects of illustrative tax changes*, June 2025 vintage (HM Revenue and Customs, 2025). The 2026, 2028–29 and HMRC columns are carried by the committed chart data at `validation/figures/chart_data.json` (key `obr_reform`), rendered as the comparison figure on the microsimulation's overview page and the OBR emulator's validation page; the £7.38bn endpoint is not in that file, and Section 6.4 records what does and does not stand behind it.

enough to admit both HMRC figures and most of the plausible range besides, so it could not have distinguished this costing from the benchmark it is compared against. It now pins ±5 per cent of £6.46bn and names the surfaces that must move with it, which makes it a check on the published number rather than a smoke test against gross error—but only when it is run.

The £7.38bn 2030 endpoint has no scored run behind it anywhere in this project. In the repository the pair exists in exactly two forms: as prose, and as the endpoints of `np.linspace(6.46, 7.38, 5)` in the sister paper's plotting script (Ahmadi, 2026a), from which the interpolated middle row is read off. The OBR emulator's own repository freezes both as named constants and says so in terms—the microsimulation is not runnable there, so the number is carried, not computed. No committed file in this project records a 2030 score, and no environment used to prepare this paper could reproduce one: the `policyengine` package is not installed and the microdata is gated. A reader should treat 6.46 as a figure with a runnable but loosely gated check behind it, and 7.38 as a figure with none.

The costing is below the official reckoner at every year that can be compared, and the gap widens with the horizon. Four differences account for the direction and shape of that gap, and none of them is a defect to be tuned away:

1. **Data basis.** HMRC costs from administrative Survey of Personal Incomes data; this estimate comes from an enhanced household survey. Administrative data captures the top of the income distribution—where a rate change on a broad band still raises substantial revenue—more completely than a survey does, even an imputed and reweighted one (Section 5).

2. **Behavioural content.** The official figure is a post-behavioural *direct* effect embedding taxable-income elasticities for higher incomes (Saez et al., 2012); this one is static by construction, with no behavioural response at all. The two are not the same object.

3. **Baseline policy assumptions.** The two costings condition on different threshold-freeze paths and uprating assumptions over the window, which matters more in later years.

4. **Fiscal drag over the horizon.** The widening from −6.4 to −15.6 per cent is concentrated in the later years, consistent with survey microdata capturing the drift of incomes across thresholds less fully than administrative data. This is the most speculative of the four: it is inferred from the shape of two paths, one of which is an interpolation, and nothing here decomposes the gap.

6.5 What this benchmark does and does not establish

It is a single comparison, and this paper declines to call it a validation. It covers one reform, in one country, on one tax, in one direction, against a figure that is itself an estimate produced on a different data basis and on a different *basis* in the technical sense—HMRC's income-tax reckoner is post-behavioural, this costing is static—so the deviations in Table 1 measure the distance between two objects that are not the same kind of thing, not a distance from truth. Being 6.4 per cent under the official figure in the one matched year is reassuring about the order of magnitude and about the absence of gross error in the aggregation, uprating and weighting chain. It is not evidence that the model's costings are accurate to six per cent in general; the middle row is not an independently scored year; the last row is not a year-matched comparison; and the direction is one-sided, which is what a systematic basis difference looks like rather than what noise looks like.

Two further gaps should be named. There is no committed external benchmark for the US side of the model at all: everything in this section is UK. And there is no benchmark anywhere for the distributional output—the decile impacts and winner/loser counts that a costing is usually quoted alongside—which is checked for internal consistency (Section 4.2) but not against any published distributional analysis. Extending the benchmark set in both directions is the obvious next validation work, and Section 9 records it as such.

7 What it cannot answer, and what the suite supplies

7.1 The refusal list

The capability registry records four things this model cannot answer, and the list is absolute rather than aspirational: *GDP, inflation, interest rates, macro feedback*. Its horizon is a single policy year. It is static and partial-equilibrium by design—prices, wages and output do not move, and behavioural responses, where used at all, are optional and post-hoc rather than part of the equilibrium.

This is not a gap in the implementation; it is the model class. A tax–benefit microsimulation answers what the rules do to incomes, given everything else. What happens to everything else is a different question requiring a different model, and the rest of the suite exists to answer it. The point of stating the refusal list explicitly, and of having the score block assert that GDP, consumption and investment are out of scope rather than returning them empty, is that a consumer reading a costing must not read the absence of a macro effect as a zero macro effect.

7.2 The route into the OBR emulator

The suite's flagship bridge reproduces the OBR's own policy-costing workflow: static costing in, second-round effects out (Office for Budget Responsibility, 2014). Given a PolicyEngine reform and a scoring window, `obr_score_reform` runs one population microsimulation per year in the

window, collects the annual budgetary impacts in £bn, converts them to a quarterly path in £m by multiplying by 1000/4 and holding the value flat within each year, and injects that path into the OBR macroeconometric emulator as a held add-factor on nominal household disposable income—the declared injection point `HHDI_ADDFACTOR`, which preserves the emulator's endogenous income identity while propagating the shock along its own demand chain from disposable income through real disposable income and consumption to output (Ahmadi, 2026a).

For the worked 1p reform of Section 6.3, the microsimulation's annual costings of £6.46bn rising to £7.38bn become a quarterly shock of roughly −£1.62bn to −£1.85bn on nominal household disposable income, and the emulator returns a second-round GDP effect building from −0.02 per cent on impact to −0.06 per cent by the end of the scored macro window in 2027Q4 (Ahmadi, 2026a). The sign is right and the magnitude is small relative to the mechanical costing, which is what the official conventions imply for an income-tax measure.

7.3 The bridge's own limits

It would be easy to present that chain as the microsimulation acquiring a macroeconomy. It is not, and the limits belong in this paper rather than only in the emulator's, because they are limits on how far a costing produced *here* can be pushed.

It is a demand-side approximation, not a reform-incidence model. The entire distributional result—the decile impacts, the winners and losers, which households paid and which taxes moved—is discarded at the boundary. What crosses is one scalar per quarter. The macro model sees an aggregate income shock and nothing else, so two reforms raising identical revenue by entirely different means, from entirely different households, produce identical second-round paths. Any genuine model of reform incidence would need the distribution the microsimulation just computed; this route deliberately does not carry it.

The within-year profile is invented. The microsimulation produces annual numbers, so the quarterly path is flat within each year. The code describes this as deliberately crude and declared, which is the right description.

Income feedback is not recycled. The economy's effect on disposable income—income falls, so tax falls, so income falls by less—is not fed back into the costing. The static costing enters once and is not revised by what the macro model does to incomes.

Corporation tax is refused rather than mistranslated. Corporate taxes are not household-borne in the microsimulation, so a reform touching them is rejected by the bridge with an explicit error pointing at the emulator's direct corporation-tax lever. Refusing is the correct behaviour: routing a corporate measure through a household-income injection point would produce a number, and the number would be meaningless.

Supply-side channels are somewhere else. Participation, saving and capital deepening belong to the suite's overlapping-generations member (Ahmadi, 2026b). That division of labour is the point of a suite rather than an omission from one model.

The receiving model has its own posture. The emulator's impact multiplier is approximately one by construction under its demand closure, against the OBR's own published assumption of 0.3 for income tax and NICs and 0.6 for current spending (Ahmadi, 2026a). The second-round numbers are therefore to be read as sign and order of magnitude, not as point estimates—a caveat the suite prints beside the worked example rather than in a footnote.

In summary, the bridge supplies a *declared* second-round demand effect on top of an exact-arithmetic-over-estimated-population costing, and that the compounded object is weaker than either of its parts. The costing is the stronger half.

7.4 The reverse direction: the microsimulation as incidence engine

Traffic runs the other way too, and here the microsimulation is the receiver. A macro path from another member—a wage and hours path from FRB/US or the HANK model, a CPI path from the structural VAR, scenario income deltas from the ecological model—is carried into a population run as an overlay that scales the employment-income inputs of the counterfactual simulation only, through the engine's supported dynamic-simulation hook. Policy is identical on both sides, so what is measured is the automatic-stabiliser response: a negative earnings shock reduces tax revenue and raises means-tested benefit spending.

Three properties of that overlay are worth recording, because they are the kind of thing that is usually left implicit.

First, the overlay carries only the counterfactual/baseline *ratio*, never a level. The baseline run uses the stock inputs, which already embed the official forecast the macro member is calibrated to, so the static effect of a reform can never be counted twice; a no-op macro result produces no overlay at all and dynamic scoring reduces exactly to static scoring. The cached baseline never sees the overlay, which is the structural guarantee rather than a convention.

Second, the scaling is applied uniformly across households, which understates the concentration of real incidence: an aggregate labour-income fall is spread over everyone rather than landing on the people who actually lost jobs. This is stated in the caveats attached to the result.

Third, an implausible factor is refused rather than applied. A scaling factor outside $[0.5, 2.0]$ is treated as a degenerate solve or a mis-sized shock and raises, on the argument that the modeller should inspect the macro run rather than push its output through the household model.

7.5 An empirical finding worth preserving

The overlay mechanism above is the second design. The first overrode a derived uprating index in the reform dictionary, and it was silently dead. Per-year population datasets are pre-uprated at dataset *build* time, so simulation-time uprating parameters are never consulted for input variables: two population runs overriding the 2026 average-earnings index—one by a factor of 0.99 and one drastically, to 0.84—both returned exactly zero everywhere, in revenue, in winners, in losers, and in every decile. A mechanism that returns a plausible-looking zero is worse than one that errors, and the codebase now refuses user reforms under the affected parameter prefix outright rather than accepting a reform it knows will do nothing. It is recorded here because it is exactly the class of defect that a paper describing a model in use should preserve and a marketing page would not.

8 Reproducibility, data, and vintage

Reproducibility for this model is not one property but two, split along exactly the boundary of Section 5.

8.1 Household results: reproducible by anyone

A household calculation needs no dataset and no credential. Install the package, describe the people, pick a year, and the result is deterministic. This is an unusually strong reproducibility property for fiscal analysis, and it is why the worked example of Section 6 can be offered as a check the reader performs rather than a claim the reader accepts. The only version dependence is the country package: statutory rules change, and a result is reproducible against the rules as encoded in the version recorded in the run's provenance.

8.2 Population results: reproducible with access

Population runs require microdata, and the two countries differ. The US population dataset is Current Population Survey based and public. The UK population dataset is the enhanced Family Resources Survey—by default the `enhanced_frs_2023_24` vintage—distributed through a gated Hugging Face repository; the first run downloads roughly 125MB and requires a `HUGGING_FACE_TOKEN` from an account granted access. The adapter fails with an explicit message naming the missing credential rather than an opaque download error, and the hosted server provisions the credential server-side so hosted callers never see it.

That gate is a real limit on this paper's own reproducibility, and it should be stated plainly: a reader can reproduce every household figure here exactly, and cannot reproduce the £6.46 billion costing without being granted access to the underlying microdata. The rules are open; the microdata is licensed. This is the ordinary condition of the field—national statistical agencies release household surveys under conditions—but it means "open model" and "openly checkable result" are not the same claim, and only the first is fully true of the population side.

8.3 Measured cost of a population run

The resource envelope is recorded in the integration code from measurement rather than estimate. On the UK 2026 enhanced FRS—53,508 households—one simulation run takes about six seconds, peaks at roughly 1.8GB of resident memory, and leaves about 92MB of derived per-year data on disk. Because the baseline simulation is cached in-process per country, year and dataset, a session scoring several reforms pays for one baseline and roughly six seconds per additional reform. The registry summarises the whole picture as "sub-second household; minutes population," the minutes being dominated by the first dataset build and download rather than by the arithmetic.

8.4 Vintage, and two declared inconsistencies

The registry records this model's data vintage as "country package and dataset dependent; recorded per run," which is the only accurate general statement: there is no single vintage, because the statutory rules travel with the installed country package and the microdata travels with the dataset. Every population result therefore carries its dataset name in provenance, and every score block repeats it in its assumptions with the household count.

Two vintage inconsistencies are declared rather than smoothed over.

The UK dataset lags upstream. Population runs here use `enhanced_frs_2023_24`. Upstream now publishes an FRS 2024–25 vintage, which this integration has not adopted. A costing produced today is therefore computed on a survey one release behind the newest available, and the ageing described in Section 5 carries it a year further.

Statute vintages diverge inside the suite. This integration tracks the latest released country model, while the overlapping-generations member is pinned to `policyengine-uk==2.88.0`. The two members can therefore disagree about baseline statute, which means a reform scored through both is not, strictly, the same baseline in each. The inconsistency is declared on the model's validation page and repeated here because a reader comparing two members' answers deserves to know that some of the difference is a version pin.

8.5 Provenance as the reproduction instruction

Every result returned by the integration layer carries the model identifier, the installed distribution and version, the source repository URL, the data vintage, the baseline definition, and an instruction: check out the recorded source at the recorded revision, install the recorded adapter version, and rerun the serialised inputs. That is the operational meaning of reproducibility here—not that a number can be re-derived from a description in a paper, but that the run can be reconstructed from the record the run itself emitted.

9 Limitations

The PolicyEngine Macro suite grades its members on a *verification gradient*: how far a model's outputs can be audited against ground truth should temper the weight placed on its answers. This model sits at the top of that gradient—it is checked against implemented legislation rather than against another model's published output—which makes it the suite's most trustworthy member and does not make it trustworthy for everything. In the order of how much they should change a reader's behaviour:

1. **Population uncertainty is described, not quantified.** This is the first limitation because it is the one this paper most wants a reader to carry away. The registry names survey and calibration uncertainty; the score block's revenue quantity says "not estimated in this result"; and nothing in the pipeline computes a replicate-weight or bootstrap standard error, propagates imputation variance, or attaches a range to a costing. Every population number in this paper is a point estimate whose dispersion is unknown to its producer. Quantifying it—starting with design-consistent variance from the survey's own replicate weights—is the most valuable single piece of outstanding work on the model.

2. **Static by construction, one year at a time.** No behavioural response is included in the default path; behavioural adjustments, where used at all, are optional and post-hoc rather than equilibrium objects. There is no general equilibrium: prices, wages and output do not move. The horizon is a single policy year, and a multi-year window is several independent annual scores rather than a dynamic path. GDP, inflation, interest rates and macro feedback are outside the model entirely.

3. **Statutory coverage is not audited here.** This paper documents the model as the suite uses it. It does not measure how completely the upstream country packages cover UK or

US statute, does not report their test coverage, and offers no estimate of how much of the tax and transfer system is encoded. The hand-checkable example of Section 6 verifies the rules it exercises and nothing more, and a provision nobody has implemented will not announce itself.

4. **Known partial coverage inside the household calculator.** Capital gains tax is not computed by the household calculator, so three of the thirteen curated reform paths are valid reforms that leave a household result unmoved and must be scored at population level. The catalogue carries that note on each affected path. It is a documented hole rather than a hidden one, but it is a hole, and it is a reminder that "exact" in Section 5 means exact *for what is computed*.

5. **One external benchmark, one country, one tax.** The ready-reckoner comparison of Section 6.3 is the only committed check against an outside number. It is UK-only; there is no committed US benchmark. It is aggregate-only; there is no committed check of the decile impacts or the winner and loser counts against any published distributional analysis. And it is measured against another estimate produced on administrative data with embedded behavioural elasticities, so the deviation is a distance between two estimates rather than an error.

6. **The benchmark's own numbers have thin provenance.** Table 1's 2028–29 figure of £6.92bn is a linear interpolation between the two endpoints, not an independently scored year, and its −15.6 per cent deviation inherits that construction. The 2030 row is not year-matched: the June 2025 reckoner stops at 2028–29. And the £7.38bn endpoint itself has no scored-run artifact anywhere in the project—the pair survives as prose and as `np.linspace(6.46, 7.38, 5)` in a plotting script—while the £6.46bn figure's only executable check is skipped without a microdata token. That check passed anything between £4bn and £9bn until August 2026 and now pins ±5 per cent of £6.46bn, so the 2026 endpoint is at last gated against something it could fail—when it is run. Section 6.4 sets this out. Producing and committing a scored 2030 run remains the second-most valuable piece of outstanding work after quantifying the interval.

7. **Data vintage lag and intra-suite statute divergence.** UK population runs use `enhanced_frs_2023_24` while upstream publishes a newer FRS vintage this integration has not adopted; and this integration tracks the latest country model while the OLG member is pinned to `policyengine-uk==2.88.0`, so two members of the same suite can disagree about baseline statute.

8. **Population results are not reader-reproducible.** The UK microdata is gated. Household results are reproducible by anyone; population results are reproducible only by someone granted access. The asymmetry is inherent to licensed survey data and is not resolved by the model being open source.

9. **The macro bridge is an approximation, and the compounded object is weaker than its parts.** The route into the OBR emulator discards the distributional detail the microsimulation computed, injects the whole costing at one point as a held add-factor on nominal household disposable income, interpolates it flat within each year, does not recycle income feedback, and delivers its second-round numbers from a model whose impact

multiplier is approximately one by construction against an official assumption of 0.3 for income tax. Read those numbers as sign and order of magnitude (Section 7.3).

10. **Uniform incidence in the reverse bridge.** When a macro path is carried down into a population run, employment income is scaled uniformly across households, which understates how concentrated real incidence is—a shock that in reality falls on people who lose jobs is spread over everyone who has one.

11. **A declared interface asymmetry.** The household adapters require the country argument explicitly, on the reasoning that defaulting would silently score the wrong country; the population adapter retains a UK default for backwards compatibility. Two entry points to the same model therefore treat the same mistake differently, and the safer behaviour is the newer one.

12. **The boundary is documented, not enforced.** Nothing prevents a consumer from taking a population figure, ignoring the uncertainty declaration that accompanies it, and quoting it with the confidence appropriate to a household calculation. The registry, the provenance record and the score block all state the distinction; none of them can compel it to be read.

10 Conclusion

This paper has described the PolicyEngine tax–benefit microsimulation as the PolicyEngine Macro suite uses it: statute encoded as a graph of dated, parameterised variables, resolved at household resolution for the United Kingdom and the United States, driven through an integration layer that validates the reform object, records provenance, and returns a score in a schema shared with the suite's macro members.

Its central claim is a boundary. A household calculation is exact arithmetic—no sampling, no weights, no imputation, no estimated coefficients—and can be reproduced with a pencil, which is why the registry records its uncertainty as "none." A population estimate is a weighted sum of that same exact arithmetic over an *estimated* population, and inherits sampling, measurement, calibration and ageing error from the survey beneath it, which is why the registry records survey and calibration uncertainty in the same sentence and after a semicolon. The two are different in kind, not in degree, and the practical consequence is asymmetric: improving the rule engine barely moves a population costing, while improving the microdata does.

The validation posture follows the same split. There is no forecast error to report because nothing is forecast; the rules are checked against implemented legislation, and one worked case—£50,000 of employment income yielding £7,486 of income tax, £2,994 of National Insurance and £39,520 of net income in 2026—is offered as a check the reader performs rather than a claim the reader accepts. The single committed external benchmark, 1p on the UK basic rate at £6.46bn in 2026 rising to £7.38bn by 2030 against HMRC's £6.9bn rising to £8.2bn, comes in *below* the official figure at every year that can be matched—by 6.4 per cent in the first scored year, widening thereafter—for reasons that are understood and are not defects to be tuned away: administrative versus survey data, a post-behavioural official basis against a static one, threshold-freeze assumptions. It is one reform, one country, one tax, one direction, measured against another estimate of a different object, with only its first year both year-matched and gated at all; the paper declines to call it a validation.

What the model cannot do is short and absolute: GDP, inflation, interest rates, macro feedback. The suite supplies those through bridges, and the paper has been as explicit about the bridges' limits as about the model's. The route into the OBR emulator discards the distribution the microsimulation computed and injects a single scalar per quarter as a held add-factor on nominal household disposable income; it is a demand-side approximation, not a general model of reform incidence, and its output should be read as sign and order of magnitude.

The clearest piece of outstanding work is the one this paper's central section had to admit. The uncertainty on a population estimate is documented as a sentence and not as a number: no interval, no standard error, no propagated imputation variance. A distinction between exact arithmetic and estimated aggregates is worth more when the second half has a range attached to it, and computing design-consistent variance from the survey's own replicate weights is the natural first step. Beyond that: adopting the newer FRS vintage, building a committed US benchmark to match the UK one, checking distributional output against published analyses rather than only for internal consistency, and carrying incidence rather than a scalar across the macro bridge.

References

Ahmadi, V. (2026a). An open OBR macro model for UK fiscal scoring. Policyengine macro working paper, PolicyEngine.

Ahmadi, V. (2026b). An overlapping-generations model for PolicyEngine reform scoring. Policyengine macro working paper, PolicyEngine.

Bourguignon, F. and Spadaro, A. (2006). Microsimulation as a tool for evaluating redistribution policies. *Journal of Economic Inequality*, 4(1):77–106.

Brewer, M., Etheridge, B., and O'Dea, C. (2017). Why are households that report the lowest incomes so well-off? *The Economic Journal*, 127(605):F24–F49.

Burgess, S., Fernández-Corugedo, E., Groth, C., Harrison, R., Monti, F., Theodoridis, K., and Waldron, M. (2013). The Bank of England's forecasting platform: COMPASS, MAPS, EASE and the suite of models. *Bank of England Working Paper*, No. 471. Bank of England, London.

Chang, A. C. and Li, P. (2015). Is economics research replicable? sixty published papers from thirteen journals say "usually not". *Finance and Economics Discussion Series*, 2015-083. Board of Governors of the Federal Reserve System.

Department for Work and Pensions (2025). Family resources survey. Statistical collection, GOV.UK.

Deville, J.-C. and Särndal, C.-E. (1992). Calibration estimators in survey sampling. *Journal of the American Statistical Association*, 87(418):376–382.

Feenberg, D. and Coutts, E. (1993). An introduction to the TAXSIM model. *Journal of Policy Analysis and Management*, 12(1):189–194.

Figari, F., Paulus, A., and Sutherland, H. (2015). Microsimulation and policy analysis. In Atkinson, A. B. and Bourguignon, F., editors, *Handbook of Income Distribution*, volume 2B, chapter 24, pages 2141–2221. Elsevier, Amsterdam.

Herndon, T., Ash, M., and Pollin, R. (2014). Does high public debt consistently stifle economic growth? A critique of Reinhart and Rogoff. *Cambridge Journal of Economics*, 38(2):257–279.

HM Revenue and Customs (2025). Direct effects of illustrative tax changes. Official statistics bulletin, GOV.UK. Ready reckoner of tax changes for 2026–27 to 2028–29, April 2026 implementation.

Meyer, B. D., Mok, W. K. C., and Sullivan, J. X. (2015). Household surveys in crisis. *Journal of Economic Perspectives*, 29(4):199–226.

Mitton, L., Sutherland, H., and Weeks, M., editors (2000). *Microsimulation Modelling for Policy Analysis: Challenges and Innovations*. Cambridge University Press, Cambridge.

Office for Budget Responsibility (2014). Policy costings and our forecast. Briefing Paper No. 6, Office for Budget Responsibility.

OpenFisca (2025). OpenFisca: A versatile microsimulation free software. https://openfisca.org. `policyengine-core` is a fork of the OpenFisca framework.

Orcutt, G. H. (1957). A new type of socio-economic system. *The Review of Economics and Statistics*, 39(2):116–123.

Policy Simulation Library (2025). Tax-calculator: An open-source microsimulation model of US federal income and payroll taxes. https://github.com/PSLmodels/Tax-Calculator.

PolicyEngine (2025). PolicyEngine: Open-source tax and benefit microsimulation. https://policyengine.org. Software and documentation; UK model at https://github.com/PolicyEngine/policyengine-uk, US model at https://github.com/PolicyEngine/policyengine-us.

Saez, E., Slemrod, J., and Giertz, S. H. (2012). The elasticity of taxable income with respect to marginal tax rates: A critical review. *Journal of Economic Literature*, 50(1):3–50.

Sutherland, H. and Figari, F. (2013). EUROMOD: The European Union tax-benefit microsimulation model. *International Journal of Microsimulation*, 6(1):4–26.

A The curated reform-parameter catalogue

Table 2 lists the thirteen reform paths the suite curates and exposes through its `list_reform_parameters` tool. The paths are curated by hand and every one has been verified to resolve through a household run; the baseline values, labels and units are resolved from the installed parameter tree at call time rather than transcribed here, so a path that ceases to resolve upstream is returned marked as not live with an explicit error rather than a silently stale number. The catalogue is a convenience surface, not a statement of the reformable parameter space, which is the whole parameter tree.

Table 2: The curated reform-parameter catalogue

Country	Path	Description	Unit
UK	`gov.hmrc.income_tax.rates.uk[0].rate`	Income tax basic rate (England, Wales, NI)	decimal
UK	`gov.hmrc.income_tax.rates.uk[1].rate`	Income tax higher rate	decimal
UK	`gov.hmrc.income_tax.allowances.personal_allowance.amount`	Income tax personal allowance	GBP per
UK	`gov.dwp.universal_credit.means_test.reduction_rate`	Universal Credit taper (earnings reduction) rate	decimal
UK	`gov.hmrc.national_insurance.class_1.rates.employee.main`	Employee National Insurance main rate	decimal
UK	`gov.hmrc.child_benefit.amount.eldest`	Child Benefit weekly amount, eldest child	GBP per
UK	`gov.hmrc.cgt.basic_rate`†	CGT rate for basic-rate taxpayers	decimal
UK	`gov.hmrc.cgt.higher_rate`†	CGT rate for higher and additional-rate taxpayers	decimal
UK	`gov.hmrc.cgt.annual_exempt_amount`†	CGT annual exempt amount	GBP per
US	`gov.irs.credits.ctc.amount.base[0].amount`	Child Tax Credit base amount per child	USD per
US	`gov.irs.credits.ctc.amount.adult_dependent`	CTC amount for adult dependents	USD per
US	`gov.irs.income.bracket.rates.2`	Federal income tax rate, bracket 2	decimal
US	`gov.irs.deductions.standard.amount.JOINT`	Standard deduction for joint filers	USD per

Notes: †These are valid reform paths, but the household calculator does not compute capital gains tax, so a household result will not move under them; they must be scored at population level. The catalogue carries that note on each affected entry. Source: `PE_PARAMETERS` in `integration/src/policyengine_macro/core.py`.

B The worked household arithmetic

The validation case of Section 6 in full. A single adult aged 35, resident in England, with employment income of £50,000, evaluated for policy year 2026 with no reform:

	£
Employment income	50,000
less personal allowance	−12,570
Taxable income (all within the basic-rate band)	37,430
Income tax at 20%	7,486
Employee National Insurance at 8%	2,994
Household net income (HBAI concept)	39,520
Reform: basic rate 20% → 25%	
Additional income tax, 5% × 37,430	1,872
Change in household net income	−1,872

Every line is deterministic arithmetic over the implemented rules, reproducible from the published allowance and rate schedule without running the model. The reform is expressed as `{"gov.hmrc.income_tax.rates.uk[0].rate": 0.25}`, and the corresponding one-point version, `{"...rate": 0.21}`, is the reform costed at population level in Section 6.3. Source: the worked example published on the model's code and validation pages.

C The reform-object grammar and its refusals

A reform is a non-empty dictionary whose keys are parameter-path strings. A value is either a number, applied from the start of the scored year, or a `{date: value}` mapping with ISO `YYYY-MM-DD` keys and numeric values. The suite's validator normalises both forms to the second and rejects everything else with a message naming the supported shapes, so that a malformed reform never reaches the model as a partially interpreted object. The refusals, and their reasons:

- *An empty or non-dictionary reform* is rejected: there is no default reform, and an empty one would score the baseline against itself.
- *A non-string or empty key* is rejected: keys are parameter paths.
- *A value that is neither a number nor a mapping* is rejected with its offending type named.
- *An empty value mapping* is rejected rather than treated as a no-op.
- *A non-ISO date key* is rejected; effective dates must be `YYYY-MM-DD`.
- *A date-range key* is rejected with an explanation, and this is the substantive one. The reform API applies a value from an effective date and leaves it in force indefinitely—there is no end date to set—so a range key cannot be honoured. Accepting it would score a permanent reform while the caller believed they had scored a temporary one. The validator instead directs the caller to use the start date alone for a permanent reform, or to score each affected year individually and treat later years as baseline for a time-limited one.

Source: `validate_reform` in `integration/src/policyengine_macro/core.py`.