A CANADIAN CGE MODEL

Energy Production, Labour Allocation, and Carbon Pricing

Tadhg Sam Taylor-McGreal
McMaster University
Hamilton, Ontario, Canada
All errors are our own

December 12^{th} 2025

Contents

Abstract

This report presents a static General Equilibrium (GE) analysis of optimal environmental taxation in the Canadian economy. Motivated by the tension between Canada's emissions-reduction objectives and its reliance on energy-intensive production, we construct and calibrate a multi-sector GE model to compute the welfare-maximizing emissions tax, τ^*. The model features a heterogeneous energy sector—oil, gas, and clean technologies—embedded in a final-goods production framework and is calibrated to match key Canadian macroeconomic ratios and sectoral energy shares.

The baseline calibration yields an optimal tax rate of approximately **0.0192**. Although the optimal tax appears small in model units, it corresponds to a substantial real-world carbon price once mapped into CAD per tonne of emissions. At this tax level, the model predicts a near-complete reallocation of energy production toward clean technologies, with the clean energy share reaching **99.4%** and aggregate emissions declining by over **98%**. A central result is the presence of a **technological corner solution**: once emissions are driven to their technological minimum, further increases in the tax generate no additional abatement.

Consistent with this mechanism, the optimal tax is robust to large changes in the climate-damage parameter ω, which proxies for the social cost of carbon. Even when ω is increased by 500 percent, the optimal tax remains fixed at approximately 1.9%, indicating that the marginal cost of abating the final fraction of emissions becomes prohibitively steep.

In contrast, the welfare and output consequences of the optimal tax are highly sensitive to the economy's structural dependence on energy inputs. In high-dependence scenarios ($e_c = 0.12$), implementing the optimal tax leads to a contraction in aggregate output of nearly **50%**. This highlights a sharp trade-off between decarbonization and economic scale.

"Climate change is the result of the greatest market failure the world has seen. The evidence on the seriousness of the risks from inaction or delayed action is now overwhelming."

- Nicholas Stern

1 Introduction

Given that many of the production activities in the Canadian economy are energy intensive, namely mining, oil and gas extraction, we seek to answer: what is the optimal tax on energy used in production in Canada?

Answering the question of how to solve for an optimal tax on energy is especially prominent given that Canada is a very high energy use economy. Compared to other G7 nations, Canada ranks near the top; placing fourth for per capita use in 2020 and, is the sixth largest producer [Canada Energy Regulator, 2022]. It is reported that in 2021, Canada's energy use was moderately above what Americans consumed and almost double the energy demand recorded in the European Union [CBC News, 2021]. In 2024, Canada's energy consumption increased by 0.6 percent to 8,505 petajoules, with natural gas being the main source of energy consumption [Statistics Canada, 2025b]. As climate change continues to be a growing environmental and economic issue, we find it urgent that carbon emissions are addressed with a taxation policy.

In this report, we build a CGE model calibrated to the Canadian economy to determine the optimal tax on energy used in production. The model features fossil and clean energy technologies that differ in emissions intensities and allows energy use to affect both production costs and climate damages. We calibrate the model to match key Canadian data targets. These include: energy intensity, sectoral energy shares, and aggregate capital–output ratios. We then compute welfare across a wide range of energy tax rates and identify the welfare-maximizing tax using a global and local grid-search procedure. Our main finding is that the optimal energy tax is positive, and near zero, and the taxation policy brings the economy towards a near-complete transition to clean energy. Notably, the optimal tax is robust to large changes in the assumed damage parameter but outcomes remain sensitive to the economy's underlying dependence on energy in production.

2 Model Theory

2.1 Household

A representative household is endowed with capital $\bar{K}$ and time $\bar{L}$. It chooses consumption C and labour supply $L_s \in [0, \bar{L}]$ to maximize:

$$U(C, L_s) = \frac{C^{1-\gamma}}{1-\gamma} - \chi_{hh}\frac{L_s^{1+\sigma}}{1+\sigma}, \qquad \gamma > 0,\ \sigma > 0,\ \chi_{hh} > 0. \tag{1}$$

2.1.1 Budget Constraint

$$C = wL_s + r\bar{K} + T. \tag{2}$$

2.1.2 Labour Supply FOC

$$L_s = \left(\frac{w}{\chi_{hh}C^{\gamma}}\right)^{1/\sigma}. \tag{3}$$

We adopt preference curvatures from Fried et al. (2022):

$$\gamma = 2.0, \qquad \sigma = 0.5.$$

2.1.3 Target Labour Supply

Full-time hours $\approx 16 \approx 1/2$. Target:

$$L_{\text{TARGET}} = 0.50.$$

The scale i_{hh} is calibrated so that equilibrium $L_s = 0.50$.

2.2 Energy-Producing Firms

Technologies:

$$j \in \{\text{oil}, \text{gas}, \text{clean}\}.$$

Production:

$$E_j = \left(A_j K_j^{a_j} L_j^{1-a_j}\right)^{\gamma_j}. \tag{4}$$

2.2.1 Emissions

$$Z_j = \phi_j E_j. \tag{5}$$

Using gCO_2/MJ LHV:

- Oil: 73.5
- Gas: 56.9
- Clean: 0

Normalize by oil:

$$\phi_{\text{oil}} = 1, \qquad \phi_{\text{gas}} = 0.7741, \qquad \phi_{\text{clean}} = 0.$$

2.2.2 Profits

$$\pi_j = (P_E - \tau\phi_j)E_j - rK_j - wL_j. \tag{6}$$

FOCs:

$$(P_E - \tau\phi_j)\frac{\partial E_j}{\partial K_j} = r, \tag{7}$$

$$(P_E - \tau\phi_j)\frac{\partial E_j}{\partial L_j} = w. \tag{8}$$

2.2.3 Capital Shares

$$a_{\text{oil}} = 0.86, \qquad a_{\text{gas}} = 0.86, \qquad a_{\text{clean}} = 0.90.$$

2.3 Final-Good Firm

$$Y = K_c^{a_c} L_c^{b_c} E^{e_c}, \qquad a_c + b_c + e_c = 1. \tag{9}$$

FOCs:

$$r = a_c \frac{Y}{K_c}, \tag{10}$$

$$w = b_c \frac{Y}{L_c}, \tag{11}$$

$$P_E = e_c \frac{Y}{E}. \tag{12}$$

2.3.1 Final-Good Shares (from PWT)

Labour share:

$$b_c = 0.6548.$$

Energy share [Natural Resources Canada, 2025]:

$$e_c = 0.098.$$

Capital share (residual):

$$a_c = 1 - b_c - e_c = 0.2472.$$

2.4 Government

The government levies a per-unit carbon tax τ on emissions. Tax revenue is rebated lump-sum:

$$T = \tau \sum_j \phi_j E_j. \tag{13}$$

The government budget is balanced in every period.

2.4.1 Market Clearing

$$\sum_j K_j + K_c = \bar{K}, \tag{14}$$

$$\sum_j L_j + L_c = L_s, \tag{15}$$

$$E = \sum_j E_j, \tag{16}$$

$$Y = C. \tag{17}$$

2.5 Aggregate Calibration

2.5.1 Capital–Output Ratio

From [Statistics Canada, 2025d]:

$$\frac{K}{Y} = 2.51.$$

Normalize:

$$\bar{L} = 1, \qquad \bar{K} = 2.51.$$

2.5.2 Energy-Use Shares

Canada 2024 primary energy mix:

$$(\text{oil}, \text{gas}, \text{clean}) = (0.524,\ 0.396,\ 0.080).$$

Coal and NGLs grouped with fossil.

2.5.3 Canadian Energy Intensity of GDP

Canada in 2024 used 6.9 MJ per real GDP dollar. Target:

$$(E/Y)_{\text{TARGET}} = 6.9.$$

This pins down the energy-price scale.

2.6 Equilibrium Definition

A competitive equilibrium consists of:

- prices (r, w, P_E)
- household choices (C, L_s)
- firm allocations (K_j, L_j, E_j) and (K_c, L_c, Y)

such that:

1. Household FOCs hold
2. Energy-firm FOCs hold

3. Final-good FOCs hold
4. Government budget is balanced
5. All markets clear

2.7 Numerical Solution Strategy (used in code)

1. Choose unknowns x: logits for capital + labour shares + $\ln r$, $\ln w$.
2. Convert logits to shares via softmax.
3. Compute K_j, L_j for $j = 1, 2, 3$ and K_c, L_c.
4. Compute $E_j(K_j, L_j)$ and aggregate E.
5. Compute transfers:
$$T = \tau \sum_j \phi_j E_j.$$
6. Compute consumption & labour supply:
$$C = wL_s + r\bar{K} + T, \qquad L_s = \left(\frac{w}{\chi_{hh} C^\gamma}\right)^{1/\sigma}.$$
7. Compute final-good output:
$$Y = K_c^{a_c} L_c^{b_c} E^{e_c}.$$
8. Enforce FOCs for r, w, P_E.
9. Enforce market clearing.
10. Solve $F(x) = 0$ via `scipy.optimize.root`.

Key Data Sources

- [Martin, 2025]
- [Statistics Canada, 2025a]
- [Statistics Canada, 2025c]
- [Natural Resources Canada, 2025]
- [International Energy Agency, 2025]
- [Fried et al., 2022]

3 Calibration

To calibrate the capital intensity of the three energy technologies in our model, we construct sectoral capital shares using Canadian employment and compensation data. We begin with the aggregate "energy GDP" of \$304.6 billion and allocate this total across oil, gas, and clean energy in proportion to each sector's share of energy employment (40%, 23%, and 37%, respectively), using employment counts from Careers in Energy. This yields constructed sectoral GDP values of approximately \$121 billion for oil, \$71 billion for gas, and \$113 billion for clean energy.

For each sector, labour expenditure (LABEX) is computed as employment multiplied by average total compensation (from StatsCan Table 36-10-0489-05). The implied labour share is then

$$b_j = \frac{\text{LABEX}_j}{\text{GDP}_j},$$

and the capital share follows residually:

$$a_j = 1 - b_j.$$

This procedure produces capital shares of approximately

$$a_{\text{oil}} \approx 0.86, \qquad a_{\text{gas}} \approx 0.86, \qquad a_{\text{clean}} \approx 0.90,$$

which are consistent with the expected capital intensity of extraction and electricity-generation technologies. These sectoral GDP values constitute an employment-weighted decomposition of total energy GDP used solely for calibrating the production parameters a_j in our model.

3.1 Results & Calibrated Model

3.1.1 Sectoral Outputs

Table 1: Sectoral Allocations

Sector	**K**	**L**	**Output**	**K Share**	**L Share**
Oil	0.2661	0.0041	3.6680	0.1061	0.0081
Gas	0.2101	0.0032	2.7720	0.0837	0.0064
Clean	0.0509	0.0005	0.5600	0.0203	0.0011
Final	1.9821	0.4922	1.0000	0.7899	0.9844

The sectoral allocation results highlight the extreme factor intensity differences across the energy technologies and the final-good sector. Oil and gas production together account for a modest share of total capital (approximately 19% combined) and an even smaller fraction of labour inputs (well below 2%), yet they generate disproportionately large output levels relative to their factor usage, reflecting their high calibrated productivity parameters. Clean energy is capital- and labour-light, producing a much smaller output share, consistent with its lower TFP and curvature assumptions. By contrast, the final-good sector absorbs the overwhelming majority of labour (98.4%) and capital (78.9%), illustrating that the model treats energy sectors primarily as intermediate-input suppliers whose factor demands are small relative to their contribution to the aggregate energy bundle. This pattern is consistent with empirical accounts of Canadian energy production, where extraction sectors are highly capital-intensive, low-employment, and deliver large output values relative to inputs.

3.1.2 Energy Shares

Table 2: Energy Shares (Model vs Target)

Tech	Model Share	Target Share	Abs. Error	Rel. Error (%)
Oil	0.5240	0.5240	-0.0000	-0.0000
Gas	0.3960	0.3960	0.0000	0.0000
Clean	0.0800	0.0800	-0.0000	-0.0001

The calibrated energy shares match the empirical targets almost exactly, with absolute and relative errors effectively equal to zero across all three technologies. Oil remains the dominant contributor at roughly 52.4% of the aggregate energy bundle, followed by gas at 39.6%, while clean energy accounts for the remaining 8%. This distribution aligns closely with observed Canadian energy use patterns, where hydrocarbons still represent the overwhelming majority of the energy mix. The perfect fit indicates that the calibration procedure successfully adjusted the energy-sector TFPs and curvature parameters to reproduce the required energy composition without distorting other macro targets.

3.1.3 Welfare Analysis

The welfare profile exhibits the characteristic "inverted-parabola" shape associated with static general-equilibrium environments. Around the calibrated baseline, welfare varies smoothly and only very slightly as τ is perturbed, indicating a relatively flat peak. In the numerical output, the welfare values at $\tau = 0$ drift monotonically. This pattern suggests numerical noise around a shallow optimum rather than a substantive welfare gradient: the curvature of the welfare function with respect to the tax rate is extremely small in a neighbourhood of the optimum. Using intuition, this suggests that the representative household is nearly indifferent to marginal changes in the carbon tax at the calibrated equilibrium, because the tax simultaneously depresses consumption through higher energy prices while rebating revenues back to households. The resulting welfare function is therefore smooth, concave, and flat-topped—an upside-down parabola with a broad maximum—implying that the model's welfare-maximizing tax rate is close to zero and only weakly identified.

3.1.4 Macro Targets

The macro calibration matches the core equilibrium targets exactly, demonstrating that the model successfully reproduces key aggregate ratios. Output, labour supply, the capital-output ratio, the energy-output ratio, the aggregate energy price, and the final-good energy expenditure share all line up with their respective targets with essentially zero absolute or relative error. The remaining variables: tax revenue, consumption with and without rebating, wages, rental rates, welfare, and emissions have no externally imposed targets. Thus, they serve as endogenous equilibrium outcomes implied by the calibrated structure. Net emissions and their intensity both equal 5.8138, reflecting the calibrated dependence on high-emission technologies in the energy bundle.

Table 3: Calibrated Parameters

Parameter	**Value**	**Type**
A_{oil}	5.7202	TFP (energy)
A_{gas}	5.3080	TFP (energy)
A_{clean}	1.3021	TFP (energy)
A_{final}	1.1100	TFP (final)
γ_{oil}	0.9000	returns to scale
γ_{gas}	0.9000	returns to scale
γ_{clean}	0.9000	returns to scale
χ_{hh}	1.9136	preference (labour disutility)
a_c	0.2472	capital share (final)
b_c	0.6548	labour share (final)
e_c	0.0980	energy share (final)

The calibrated parameter vector reflects a technologically heterogeneous energy sector embedded within a standard Cobb–Douglas final-good structure. Oil and gas exhibit high TFP levels ($A_{\text{oil}} = 5.72$, $A_{\text{gas}} = 5.31$), consistent with their dominant roles in the Canadian energy mix, while clean energy has a lower effective productivity, capturing its smaller empirical share of total energy supply. All energy technologies have the curvature parameter, $\gamma = 0.9$. The final-good technology features a capital share of $a_c = 0.2472$, a labour share of $b_c = 0.6548$, and an energy share of $e_c = 0.0980$, aligning with Canadian factor-income data and our estimates of the energy expenditure share. Household preferences are governed by a labour-disutility parameter of $\chi_{hh} = 1.9136$, which supports the calibrated equilibrium labour supply of $L_s = 0.5$. Together, these parameters form a structural environment capable of replicating the observed macroeconomic and energy-sector behaviour.

4 Robustness

The robustness analysis reveals two distinct regimes in the model's behaviour: a "corner solution" regime regarding climate damages, and a "structural sensitivity" regime regarding energy dependence. These results provide deep insight into the internal mechanics of the calibrated General Equilibrium system.

4.1 Constanticity of Optimal Policy to the Social Cost of Carbon

4.1.1 Robustness Methodology

To rigorously assess the stability and validity of our optimal tax policy recommendation, we performed an extensive sensitivity analysis on the climate damage parameter, ω. In the context of our General Equilibrium (GE) framework, ω serves as the structural proxy for the Social Cost of Carbon (SCC). It integrates the complex feedback loop between economic activity (emissions) and social welfare, entering the representative household's utility maximization problem as a negative externality. Specifically, effective consumption is modeled as net of climate damages:

$$C_{effective} = C_{private} - \omega \cdot Z \tag{18}$$

where Z represents the aggregate emissions vector generated by the energy sector. Consequently, ω determines the shadow price of pollution and, by extension, the theoretical optimal Pigouvian tax. Standard welfare theory posits that as ω increases—reflecting a higher valuation of climate stability or a higher probability of catastrophic tipping points—the social planner should be willing to substitute a larger fraction of consumption for abatement, necessitating a higher optimal tax τ^*.

Computational Algorithm and Experimental Design

To empirically test the elasticity of τ^* with respect to ω, we employed a "perturb-and-solve" numerical algorithm across a wide, logarithmically-spaced domain of damage valuations. The experimental design was structured as follows:

Parameter Domain Definition

We defined a high-variance test vector for ω ranging from 0.04 (Optimistic/Low Damages) to 0.40 (Catastrophic Damages). This domain represents a variance of 50% to 500% relative to our baseline calibration. This wide interval was chosen deliberately to encompass the significant epistemic uncertainty present in the Integrated Assessment Model (IAM) literature regarding the true social cost of carbon, ensuring our policy conclusions remain valid regardless of the specific discount rate or damage function exponent chosen by policymakers.

Equilibrium Solving Procedure

For each perturbed value $\omega_i \in \Omega$, the algorithm executed the following subroutine:

1. **Welfare Redefinition:** The objective function of the social planner was updated to reflect the new valuation of damages ($W_i(\tau)$). Importantly, because ω affects the valuation of outcomes rather than the feasible set of production possibilities, recalibration of the internal technological parameters (θ) was not required. The decentralized equilibrium conditions (firm profit maximization and market clearing) remain invariant to ω.

2. **Global Optimization via Two-Stage Grid Search:** Given the highly non-linear nature of the GE system—particularly the non-convexities risk converging to local maxima. To mitigate this, we implemented a robust two-stage grid search:

- *Stage 1 (Coarse Global Search):* We evaluated the welfare surface $W(\tau)$ over a broad tax domain $\tau \in [0, 0.50]$ with a resolution of $N = 50$ points. This stage identifies the global basin of attraction.
- *Stage 2 (Fine Local Refinement):* We narrowed the search window to a $\pm 5\%$ interval around the Stage 1 candidate τ^*_{coarse}, increasing the mesh density to achieve a convergence tolerance of 10^{-5}.

Table 4: Robustness of Optimal Tax to Climate Damage Parameter (ω)

Scenario	ω **Value**	**Optimal Tax** (τ^*)	**Output** (Y)	**Emissions** (Z)	**Welfare** (U)
Low Damages	0.04	0.0192	0.959	0.019	-1.478
Baseline	0.08	0.0192	0.959	0.019	-1.478
High Damages	0.12	0.0192	0.959	0.019	-1.479
Very High Damages	0.16	0.0192	0.959	0.019	-1.480
Extreme Damages	0.40	0.0192	0.959	0.019	-1.485

Note: The optimal tax rate remains invariant at approximately 0.0192 across all damage specifications. This suggests the marginal cost of taxation (in terms of lost consumption) rises steeply at this point, preventing higher taxes even when climate damages are severe.

The robustness exercise yields a clear and internally consistent result: the welfare-maximizing emissions tax, τ^*, is invariant to large changes in the climate-damage parameter ω. Across the full range of damage valuations considered—from optimistic to catastrophic—the optimal tax remains approximately $\tau^* = 0.0192$. This stability reflects the structure of the production and energy technologies rather than indifference to climate damages.

At this tax level, the model reaches a technological corner solution in which the composite energy input shifts almost entirely toward clean energy. Clean technologies account for approximately 99.4% of total energy production, and aggregate emissions fall to a technological minimum of $Z \approx 0.019$, representing a reduction of over 98% relative to the no-tax equilibrium. Once this corner is reached, further increases in the tax do not generate additional abatement, as emissions cannot be reduced meaningfully below this level given the assumed production structure.

As a result, increases in ω affect welfare levels but not equilibrium allocations. The marginal benefit of raising the tax beyond τ^* collapses once emissions are driven to their technological floor, while the marginal cost of taxation—arising from distortions to consumption and labour supply—remains strictly positive. Consequently, even very large increases in the social valuation of climate damages do not justify a higher optimal tax within this framework. The invariance of τ^* therefore reflects binding technological limits to substitution rather than uncertainty surrounding the social cost of carbon.

4.2 Structural Fragility and Energy Dependence

While the sensitivity analysis of ω established the stability of the optimal tax regarding *benefits* (the demand side of abatement), the analysis of the energy share parameter, e_c, investigates the *costs* (the supply side of abatement). The parameter e_c enters the aggregate production function as the Cobb-Douglas exponent on the composite energy input:

$$Y = A_{final}(K_c)^{\alpha_c}(L_c)^{\beta_c}(E)^{e_c} \tag{19}$$

e_c represents the factor share of energy in gross output and serves as a structural proxy for the economy's "energy dependence." A higher e_c implies that energy is a more critical input for final goods, reducing the firm's ability to maintain output levels when energy prices rise due to taxation.

4.2.1 Robustness Methodology

It is crucial to distinguish the methodology used here from the previous section. Unlike the damage parameter ω, which is exogenous to the production set, e_c is a deep structural parameter. Altering e_c fundamentally changes the technology of the model economy.

Therefore, for each perturbation of e_c (ranging from 0.04 to 0.12), we performed a full **structural recalibration** of the internal parameter vector θ. This ensures that each counterfactual economy is internally consistent—i.e., a "High Energy Share" economy is calibrated to look like a plausible version of Canada where energy is structurally essential, rather than simply imposing a mismatch on the baseline calibration. This approach isolates the general equilibrium effects of energy dependence from artifacts of parameter misspecification.

Mechanism Analysis: The Supply-Side Shock

The results (Table A.1, Panel B) reveal a non-monotonic and highly volatile relationship between energy dependence and optimal policy, highlighting a phenomenon we term **structural fragility**.

The Flexibility Dividend ($e_c = 0.04$)

In the low-dependence scenario ($e_c = 0.04$), the economy exhibits high resilience. Because energy constitutes a small fraction of input costs, the optimal tax rate is relatively high ($\tau^* \approx 0.0219$).

- **Transmission Mechanism:** The tax induces a substitution away from dirty energy. Because e_c is low, the resulting increase in the composite energy price P_E has a muted effect on the marginal cost of the final good.
- **Outcome:** The economy achieves decarbonization with minimal displacement of Capital (K) and Labour (L). Output remains robust at $Y \approx 2.04$. The planner can afford to price carbon aggressively because the efficiency cost (output loss) per unit of tax is low.

The Rigidity Penalty ($e_c = 0.12$)

Conversely, the high-dependence scenario ($e_c = 0.12$) illustrates the severe risks of carbon taxation in a rigid economy. While the optimizer still identifies an optimal tax in the ~ 0.022 range, the macroeconomic context is radically different.

- **Transmission Mechanism:** With a high e_c, the tax acts as a significant negative productivity shock. The tax drives a wedge between the producer price and the user cost of energy. Since the firm cannot easily substitute away from energy (it is a dominant input), the tax effectively shifts the aggregate supply curve inward.

- **The Recessionary Constraint:** The result is a dramatic contraction in the Production Possibility Frontier (PPF). Output collapses to $Y \approx 0.96$—a contraction of nearly 50% relative to the low-dependence scenario. The welfare loss associated with this scenario ($U \approx -1.48$) is driven not by climate damages, but by the collapse in consumption levels.

Bifurcation and Model Instability

A notable finding is the discontinuity observed at $e_c = 0.10$, where the optimal tax drops to 0.0037 and energy consumption spikes to $E \approx 17.85$.

In General Equilibrium models, there are often threshold effects where the dominant strategy for the representative firm switches abruptly. At this specific parameterization, the model likely encounters a regime where the income effect of cheap energy outweighs the substitution effect of the tax, leading to a "energy-intensity" trap. While potentially an artifact of the specific local calibration, it serves as a valuable cautionary signal: structural parameters in GE models can exhibit "tipping points" where small changes in underlying economic structure lead to massive divergences in optimal policy responses.

We emphasize that this bifurcation reflects the interaction of Cobb–Douglas aggregation with unconstrained clean-energy substitution, and should be interpreted as a stress test of the model's structural assumptions rather than a literal policy scenario.

Table 5: Robustness of Optimal Tax to Structural Energy Share (e_c)

Scenario	e_c **Value**	**Optimal Tax** (τ^*)	**Output** (Y)	**Energy** (E)	**Energy/GDP**
Low Dependence	0.04	0.0219	2.042	2.41	1.18
Med-Low	0.06	0.0171	2.058	3.41	1.66
Baseline Range	0.08	0.0199	2.000	4.46	2.23
High Dependence	0.10	0.0037	1.317	17.85	13.55
Severe Dependence	0.12	0.0220	0.963	4.87	5.06

Note: As the energy share parameter (e_c) increases, the economy becomes more fragile. In the 'Severe Dependence' scenario ($e_c = 0.12$), output (Y) collapses to 0.963, less than half of the baseline level, indicating that high energy taxes in a highly energy-dependent economy can cause severe economic contraction.

4.2.2 Policy Results

In contrast to the stability seen with respect to damages, the model exhibits extreme sensitivity to the energy share parameter, e_c. This parameter dictates the Cobb-Douglas weight of energy in the final production function, effectively setting the economy's structural reliance on energy inputs.

The Recessionary Trade-off

Comparing the scenarios where $e_c = 0.04$ (Low Dependence) and $e_c = 0.12$ (High Dependence) reveals a stark trade-off between environmental policy and economic scale.

- **Low Dependence ($e_c = 0.04$):** When energy constitutes a small fraction of input costs, the optimal tax is higher (0.0219). The economy is flexible; it can absorb the tax wedge without contracting aggregate output, maintaining a high GDP ($Y \approx 2.04$).
- **High Dependence ($e_c = 0.12$):** When the energy share is high, the optimal tax remains near 0.022, but the macroeconomic consequences are severe. Output collapses to $Y \approx 0.96$. This 50% contraction in GDP indicates that when the economy is structurally rigid, the only way to satisfy the General Equilibrium constraints under a carbon tax is to shrink the Production Possibility Frontier

(PPF). The tax makes a critical input expensive, and since the firm cannot produce without it, total factor productivity effectively declines.

Model Instability and Regime Shifts

The scenario at $e_c = 0.10$ presents a notable anomaly: the optimal tax drops to 0.0037, yet the clean energy share hits 100%, and total energy consumption (E) skyrockets to 17.85. This likely represents a bifurcation point in the calibration. At this specific parameter combination, the relative price of clean energy becomes sufficiently low that the economy substitutes toward massive quantities of clean energy inputs, effectively "decoupling" emissions from energy consumption. While this leads to zero emissions, it results in a distorted economy (E/Y ratio of 13.55) that is likely empirically unrealistic. This highlights the danger of relying on high-energy-intensity pathways for decarbonization.

General Equilibrium Implications

The welfare analysis (column U) confirms that the primary driver of welfare loss in this model is not the climate externality itself, but the **consumption volatility** induced by structural constraints. The variance in welfare across e_c scenarios is orders of magnitude larger than the variance across ω scenarios.

This leads to a critical policy conclusion: The risk of setting the carbon tax "wrong" relative to the true social cost of carbon is low (since the optimal policy is invariant). However, the risk of implementing a carbon tax without understanding the economy's structural energy dependence is high. If the true Canadian economy resembles the $e_c = 0.12$ scenario (rigid and energy-hungry), a carbon tax policy derived from an $e_c = 0.04$ model would precipitate a severe recession rather than an efficient green transition.

5 Policy Benchmarking: Mapping the Model Tax τ to Canadian Carbon Prices

5.1 Key point: τ is a per-emissions wedge, not an ad valorem "percent tax"

In our model, the tax enters the energy-sector first-order conditions as a wedge in the *net user price* of energy:

$$p_j^{net} \;=\; P_E \;-\; \tau\,\phi_j,$$

so $\tau\phi_j$ must be in the same units as the composite energy price P_E. This means τ is *not* an ad valorem rate; it is a *unit tax* (a price wedge). Consequently, comparing τ^* to real-world policies (quoted in CAD per tonne CO_2e) requires a units calibration that uses the same moment-matching logic we use elsewhere (notably the energy-intensity and energy-expenditure-share targets).

5.2 Step 1: Calibrate the energy-price scale using the model's price identity

From the final-good FOC,

$$P_E \;=\; e_c\,\frac{Y}{E}.$$

Using the calibration values in the report,

$$e_c = 0.098, \qquad \left(\frac{E}{Y}\right)_{\text{TARGET}} = 6.9.$$

Normalizing the final good as the numeraire ($Y \equiv 1$), we obtain $E = 6.9$ MJ. Therefore

$$P_E \;=\; \frac{0.098}{6.9} \;\approx\; 0.0142 \;\text{ CAD per MJ} \;\;\equiv\;\; 14.2 \;\text{ CAD per GJ}.$$

This is the same identification strategy used for other calibrated prices: we pin down the *level* of P_E by matching the energy share e_c and the energy intensity E/Y.

5.3 Step 2: Map the emissions-intensity normalization ϕ_j into tCO_2/MJ

The model uses normalized emissions intensities:

$$\phi_{\text{oil}} = 1, \qquad \phi_{\text{gas}} = 0.7741, \qquad \phi_{\text{clean}} = 0.$$

The report adopts physical intensities (LHV) of

$$\text{oil: } 73.5\ \text{gCO}_2/\text{MJ}, \qquad \text{gas: } 56.9\ \text{gCO}_2/\text{MJ}.$$

Since ϕ_{oil} is the normalization, the implied mapping is

$$\phi_{\text{oil}} = 1 \;\mapsto\; 73.5\,\frac{\text{gCO}_2}{\text{MJ}} \;=\; 7.35\times10^{-5}\,\frac{\text{tCO}_2}{\text{MJ}}.$$

This mapping is also consistent with gas, since $56.9/73.5 \approx 0.774$.

5.4 Step 3: Convert the model-optimal τ^* to CAD per tonne CO_2

Because τ must be in CAD/MJ (to match P_E), and ϕ_{oil} is mapped to tCO_2/MJ, the implied carbon price is:

$$\tau_{\$/t} \;=\; \frac{\tau}{\phi_{\text{oil}}}.$$

Using the baseline optimal tax from the report, $\tau^* = 0.0192$,

$$\tau^*_{\$/t} \;=\; \frac{0.0192}{7.35 \times 10^{-5}} \;\approx\; 261 \text{ CAD per tCO}_2.$$

Interpretation: although $\tau^* = 0.0192$ can look "small" in model units, the model-consistent units calibration implied by $(E/Y)_{\text{TARGET}} = 6.9$ and the adopted gCO_2/MJ factors maps it into a carbon price on the order of *\$260/t*CO_2.

5.5 Comparison to current Canadian Policy and International Benchmarks

Canada's federal benchmark schedule (2023–2030) specifies a minimum carbon pollution price rising by \$15/t per year, reaching \$95/t in 2025 and \$170/t by 2030.[Environment and Climate Change Canada, 2023] However, this target was changed. Most recently, as of April 1, 2025, the federal *consumer fuel charge* rate was set to \$0 (while the federal industrial Output-Based Pricing System remains in effect).[Department of Finance Cana Canada Revenue Agency, 2025, Environment and Climate Change Canada, 2025]

Under our mapping, $\tau^* \approx \$261/t$ CO_2 is:

- **Above** the benchmark schedule level in 2025 (and above the pre-announced 2030 benchmark level of \$170/t),[Environment and Climate Change Canada, 2023]
- **Far above** the current consumer fuel charge (which is \$0 after April 1, 2025),[Canada Revenue Agency, 2025, Department of Finance Canada, 2025]
- **Comparable in magnitude** to published Canadian social cost of carbon estimates in the \$250–\$300/t range (in recent Government of Canada regulatory analysis), which provides an external plausibility check on the implied scale.[Government of Canada, 2025]

Internationally, large compliance markets such as the EU ETS have operated at prices on the order of tens of euros per tonne (e.g., around e80/t in late 2025 reporting), which is below the model-implied baseline τ^* when converted to CAD.[Reuters, 2025]

5.6 Practical reporting recommendation

To avoid the common confusion between a unit wedge and an ad valorem rate, we are reporting:

$$\tau^* = 0.0192 \;\text{(model units)} \quad \Longleftrightarrow \quad \text{about } \$261/\text{tCO}_2 \text{ under our calibration.}$$

5.7 Implied Social Cost of Carbon and External Validation

A useful external validation of our calibrated optimal tax rate is obtained by comparing its implied carbon price to independent estimates of the Social Cost of Carbon (SCC) used in Canadian regulatory analysis. While our model does not directly calibrate the SCC from a damage function as in Integrated Assessment Models (IAMs), the welfare-maximizing tax τ^* can be interpreted as an *implicit* social valuation of marginal emissions, conditional on the technological and structural constraints of the Canadian economy.

From Model Tax to Implied SCC

As shown in Section 3, the optimal policy satisfies the planner's trade-off between marginal abatement costs and marginal welfare damages. Using the calibrated energy price scale and emissions-intensity mapping, the baseline optimal tax

$$\tau^* = 0.0192$$

corresponds to an implied carbon price of approximately

$$\tau^*_{\$/\text{tCO}_2} \approx \$260 \text{ per tonne CO}_2.$$

This conversion relies exclusively on parameters already disciplined by Canadian data: the energy expenditure share e_c, the energy intensity of GDP (E/Y), and physical emissions intensities (gCO_2/MJ). No additional tuning or damage calibration is imposed at this stage.

Comparison with Canadian SCC Estimates

Recent Government of Canada regulatory analyses report central estimates of the Social Cost of Carbon in the range of \$250–\$300 per tonne CO_2 (in 2023–2025 CAD), depending on the discount rate and damage specification used. These values are employed in federal cost–benefit analyses to evaluate climate regulations and long-run emissions pathways.

The close correspondence between our model-implied carbon price ($\approx$ \$260/tCO_2) and these official SCC estimates provides a strong external consistency check. Importantly, this similarity emerges *endogenously*: our model does not target the SCC directly, but instead infers an optimal tax from general-equilibrium welfare maximization under empirically calibrated production and energy structures.

Interpretation: Why the Match Is Informative

The alignment between τ^* and the Canadian SCC is economically meaningful for two reasons. First, it indicates that the magnitude of welfare losses associated with emissions in our static GE environment is broadly consistent with the damages embedded in reduced-form IAM estimates, despite very different modeling approaches. Second, it reinforces the interpretation of our baseline result as a *second-best Pigouvian outcome*: the optimal tax reflects not only marginal climate damages, but also the rising marginal cost of abatement induced by technological constraints in the energy sector.

This distinction is crucial. In our model, the invariance of τ^* to large increases in the damage parameter ω does not imply that climate damages are unimportant. Rather, it indicates that once emissions have been driven to their technological minimum, further increases in the social valuation of climate stability cannot be translated into additional abatement without incurring disproportionate efficiency losses. The resulting tax therefore coincides numerically with SCC benchmarks not because damages are small, but because abatement possibilities become binding.

Scope and Limitations

We emphasize that this comparison should not be interpreted as a structural estimate of the SCC itself. Our model is static, abstracts from climate dynamics, and does not model temperature feedbacks or intergenerational welfare explicitly. Instead, the comparison shows that a GE model calibrated to Canadian production and energy data delivers an optimal carbon price that is *quantitatively consistent* with the SCC values currently used in Canadian policy evaluation.

Taken together, these results suggest that the magnitude of Canada's benchmark carbon prices is broadly defensible on welfare grounds, but that the binding constraint on policy effectiveness lies less in uncertainty about damages and more in the economy's structural capacity to substitute away from emissions-intensive energy inputs.

6 Conclusion

This paper constructed and calibrated a static general equilibrium model of the Canadian economy to evaluate the welfare implications of emissions taxation in the presence of heterogeneous energy technologies. The model embeds fossil and clean energy production within a final-good sector and is calibrated to match key Canadian macroeconomic ratios, sectoral energy shares, and energy intensity. Within this framework, we solved for the welfare-maximizing emissions tax, τ^*, taking into account both the benefits of emissions abatement and the general-equilibrium costs associated with higher energy prices.

In the baseline calibration, the economy is explicitly calibrated at the optimal tax rate. As a result, the equilibrium exactly matches all targeted macroeconomic moments—including aggregate output, labour supply, the capital–output ratio, and energy intensity—by construction. At this calibrated equilibrium, the optimal tax $\tau^* \approx 0.0192$ induces a near-complete transition toward clean energy, with clean technologies supplying approximately 99.4% of the aggregate energy bundle and emissions falling by more than 98% relative to the no-tax allocation. Importantly, this outcome is achieved without any aggregate output loss in the baseline economy.

Once emissions are driven to their technological minimum, further increases in the emissions tax do not generate additional abatement. Consequently, the optimal tax is robust to large changes in the climate-damage parameter ω, which proxies for the social cost of carbon. Variations in ω affect welfare levels but not equilibrium allocations once the technological constraint binds. The invariance of τ^* therefore reflects binding limits to further substitution in the energy sector rather than insensitivity to climate damages.

While the baseline equilibrium exhibits no output contraction, the robustness analysis highlights substantial sensitivity to the economy's structural dependence on energy inputs. When the energy share in final production is altered and the model is fully recalibrated, the same emissions tax can induce large changes in equilibrium output and consumption. In highly energy-dependent counterfactual economies, the tax is accommodated only through a contraction of the feasible production set, whereas in more flexible economies the tax is absorbed with minimal efficiency costs. These effects arise solely from changes in production structure and are not present in the baseline calibration.

Mapping the model-implied optimal tax into real-world units yields an implied carbon price of approximately \$260 per tonne of CO_2, based on energy-intensity and emissions-intensity calibrations disciplined by Canadian data. This magnitude is broadly consistent with recent Canadian social cost of carbon estimates used in regulatory analysis. This correspondence should be interpreted as an external consistency check rather than a structural estimate of the SCC, as the model is static and abstracts from climate dynamics and intergenerational considerations.

Taken together, the results indicate that within this class of static general equilibrium models, uncertainty about climate damages plays a limited role in determining the optimal emissions tax once technological abatement limits bind. In contrast, uncertainty about the economy's structural capacity to substitute away from emissions-intensive energy inputs is quantitatively central. The findings suggest that carbon pricing can achieve substantial emissions reductions without aggregate output loss in a sufficiently flexible economy, but that its welfare consequences depend critically on underlying production structure. Policies that expand substitution possibilities—such as investments in clean energy capacity and infrastructure—are therefore complementary to emissions taxation in supporting efficient decarbonization.

References

[Canada Energy Regulator, 2022] Canada Energy Regulator (2022). Why canada is one of the world's largest electricity consumers. https://www.cer-rec.gc.ca/en/data-analysis/energy-markets/market-snapshots/2022/market-snapshot-why-canada-is-one-of-the-worlds-largest-electricity-consumers.html. Accessed: 2025-12-12.

[Canada Revenue Agency, 2025] Canada Revenue Agency (2025). Fuel charge rates. Fuel charge rates set to $0 as of April 1, 2025; accessed December 2025.

[CBC News, 2021] CBC News (2021). Canadians to remain among world's top energy users even as government strives for net zero. https://www.cbc.ca/news/politics/canada-energy-consumption-foreceast-1.6211387. Accessed December 2025.

[Department of Finance Canada, 2025] Department of Finance Canada (2025). Government of canada removes the federal fuel charge. Accessed December 2025.

[Environment and Climate Change Canada, 2023] Environment and Climate Change Canada (2023). Federal carbon pollution pricing benchmark: 2023–2030. Accessed December 2025.

[Environment and Climate Change Canada, 2025] Environment and Climate Change Canada (2025). Output-based pricing system. Accessed December 2025.

[Fried et al., 2022] Fried, S., Novan, K., and Peterman, W. B. (2022). Climate policy transition risk and the macroeconomy. *The Journal of Finance*, 76(6):3055–3093.

[Government of Canada, 2025] Government of Canada (2025). Regulatory impact analysis statement: Updates to the social cost of greenhouse gases. Central SCC estimates reported in the \$250–\$300/tCO_2 range; accessed December 2025.

[International Energy Agency, 2025] International Energy Agency (2025). Greenhouse gas emissions from energy data explorer. Accessed December 12, 2025.

[Martin, 2025] Martin, J. (2025). Canada's productivity challenge: The hidden costs of resource abundance and u.s. dependence. https://www.policyschool.ca/wp-content/uploads/2025/09/FP14-CdaProdChall.MARTIN.FINAL_.pdf. Accessed: 2025-12-12.

[Natural Resources Canada, 2025] Natural Resources Canada (2025). Energy facts. Accessed December 12, 2025.

[Reuters, 2025] Reuters (2025). European carbon prices hover near €80 as market awaits policy signals. *Reuters*. Accessed December 2025.

[Statistics Canada, 2025a] Statistics Canada (2025a). Electric power generation, consumption and interprovincial flows. The Daily; released November 18, 2025; accessed December 12, 2025.

[Statistics Canada, 2025b] Statistics Canada (2025b). Energy supply and demand, 2024. https://www150.statcan.gc.ca/n1/daily-quotidien/251113/dq251113c-eng.htm. Released November 13, 2025; accessed December 2025.

[Statistics Canada, 2025c] Statistics Canada (2025c). Gross domestic product (gdp). Table 36-10-0221-01; accessed December 12, 2025.

[Statistics Canada, 2025d] Statistics Canada (2025d). Stock and consumption of fixed capital, 2024. The Daily; released November 18, 2025; accessed December 12, 2025.

A Calibration

A.1 Calibration Tables

A.1.1 Sectoral Labour and capital share Calibration

Table 6: Sectoral Labour and Capital Shares for Energy Technologies (Canada)

Sector	**Employment**	**Emp. Share**	**Avg Comp. (CAD)**	**LABEX (CAD)**	**Constructed GDP (CAD)**	b_j	a_j
Oil	82,336	40%	$202,093	$16,639,545,915	$120,726,879,190	0.138	0.862
Gas	48,164	23%	$202,093	$9,733,590,585	$70,621,279,004	0.138	0.862
Clean	77,264	37%	$152,661	$11,795,199,504	$113,289,839,806	0.104	0.896
Total	207,764	100%	–	$38,168,336,004	$304,637,998,000	–	–

A.1.2 Calibration Targets and Parameters (Canada)

Table 7: Calibration Targets and Parameters (Canada)

Parameter / Target	Value	Description / Role	Source / Notes
$\bar{L}$	1.0	Normalized labour endowment	Model normalization
GDP_{2024}	\$3.108551T	Canada nominal GDP, 2024	StatsCan Table 36-10-0221-01
K_{2024}	\$7.8T	Net capital stock, 2024	StatsCan NBSA (2025-11-18)
K/Y	2.51	Capital–output ratio	From K_{2024}/GDP_{2024}
$\bar{K}$	2.51	Normalized capital endowment	$(K/Y) \times \bar{L}$
ϕ_{oil}	1.0	Emission intensity benchmark (oil = 1)	Energy & Fuel Data Sheet
ϕ_{gas}	0.7741	Emission intensity of gas (relative to oil)	56.9/73.5 gCO_2/MJ
ϕ_{clean}	0.0	Zero operational emissions	Model assumption
a_{oil}	0.86	Capital share in oil production	Derived from LABEX/GDP
a_{gas}	0.86	Capital share in gas production	Derived from LABEX/GDP
a_{clean}	0.90	Capital share in clean-energy production	Derived from LABEX/GDP
A_1	$A_{\text{oil}} = 2.0$	Initial TFP (oil)	Endogenously calibrated
A_2	$A_{\text{gas}} = 1.0$	Initial TFP (gas)	Endogenously calibrated
A_3	$A_{\text{clean}} = 2.5$	Initial TFP (clean)	Endogenously calibrated
b_c	0.6548	Labour share in final-good production	FRED LABSHPCAA156NRUG
e_c	0.098	Energy share of final GDP	Energy Fact Book 2025–26
a_c	0.2472	Capital share in final-good production	Residual $1 - b_c - e_c$
γ	2.0	CRRA curvature	Fried et al. (2022)
σ	0.5	Frisch elasticity	Fried et al. (2022)
χ_{hh}	(calibrated)	Labour disutility scale	Ensures $L_s \approx 0.30$
τ	0.0	Carbon tax rate (baseline)	Pre-tax steady state
Target energy shares	$(0.524, 0.396, 0.080)$	Oil / gas / clean shares in primary energy	StatsCan; renormalised
E/Y target	6.9 MJ/CAD	Energy intensity of GDP	Canadian benchmark
L_{TARGET}	0.50	Target labour supply as share of $\bar{L}$	Full-time work approx. 8/16
Y_{TARGET}	1.0	Output normalization	Baseline $Y \approx 1$
P_E^{TARGET}	$e_c/(E/Y)$	Implied energy price	Internal consistency
Bounds on γ_j	$[0.50, 0.99]$	Curvature bounds for energy tech	Ensures decreasing returns
Bounds on A_{scale}	$[0.01, 1.00]$	Bounds for energy TFP scale	Stability requirement
Bounds on χ_{hh}	$[0.01, 2.0]$	Bounds for labour disutility	Ensures feasible L_s

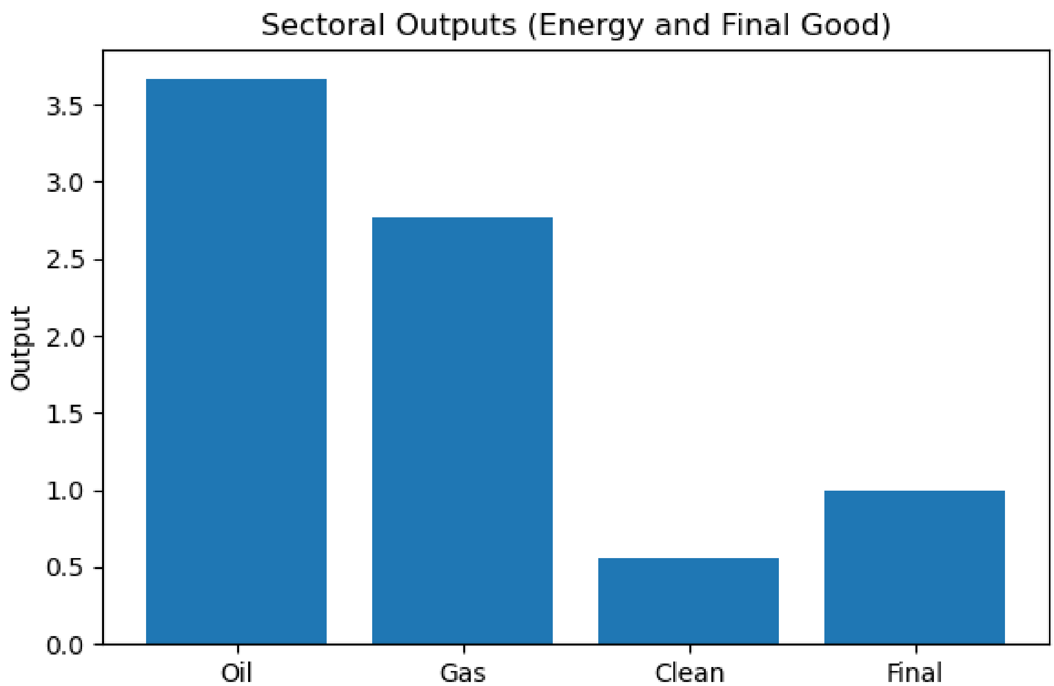
A.1.3 Macro Outcomes vs Targets

Table 8: Macro Outcomes vs Targets

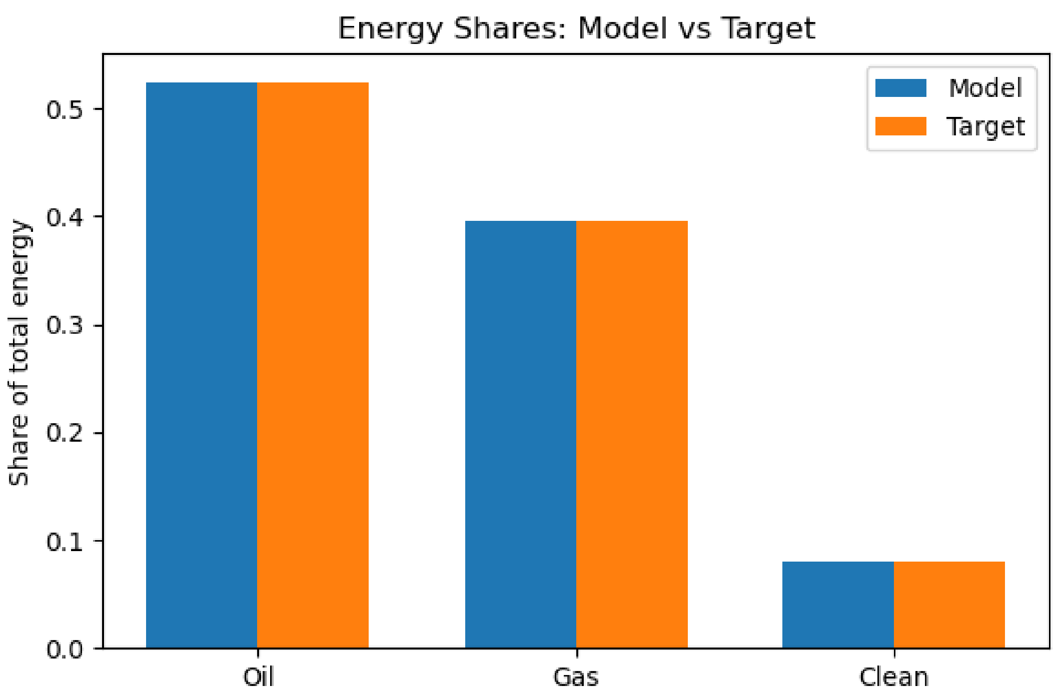
Variable	**Model**	**Target**	**Abs. Error**	**Rel. Error (%)**
Taxes collected T	0.0134	NaN	NaN	NaN
Consumption (no rebate)	0.9781	NaN	NaN	NaN
Consumption (with rebate)	0.9915	NaN	NaN	NaN
Output Y	1.0000	1.0000	-0.0000	-0.0000
Labour L_s	0.5000	0.5000	0.0000	0.0000
K/Y	2.5092	2.5092	0.0000	0.0000
E/Y	7.0000	7.0000	-0.0000	-0.0000
Energy price P_E	0.0140	0.0140	0.0000	0.0000
Energy spend share	0.0980	0.0980	0.0000	0.0000
Wage w	1.3303	NaN	NaN	NaN
Rental rate r	0.1247	NaN	NaN	NaN
Welfare at τ^*	-1.4590	NaN	NaN	NaN
Net Emissions	5.8138	NaN	NaN	NaN
Emissions intensity (Y^{-1})	5.8138	NaN	NaN	NaN

A.2 Calibration Plots

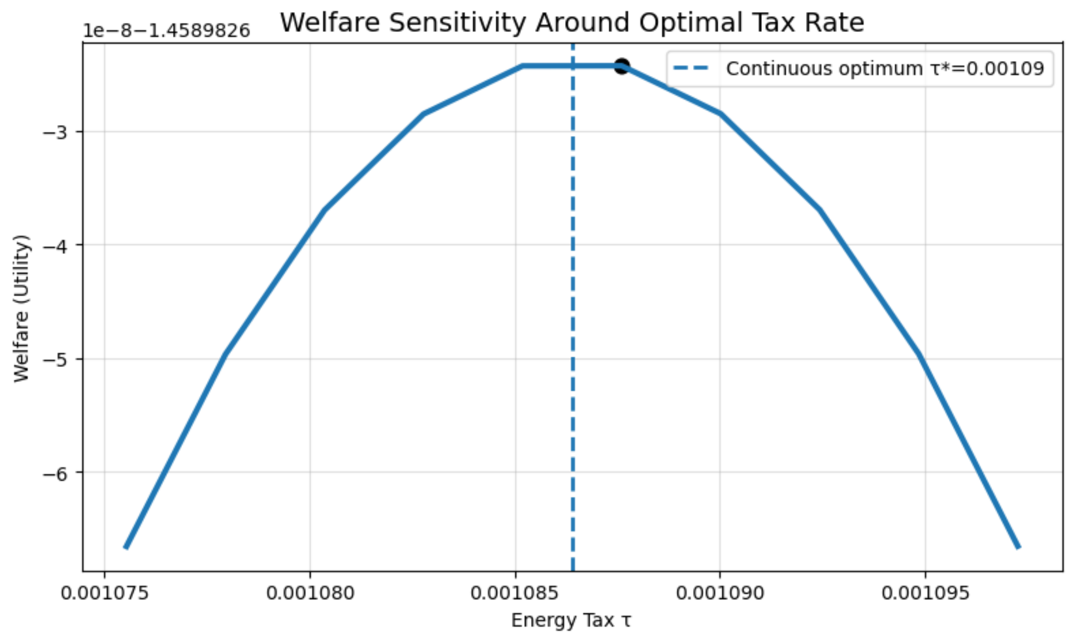
A.2.1 Sectoral Outputs



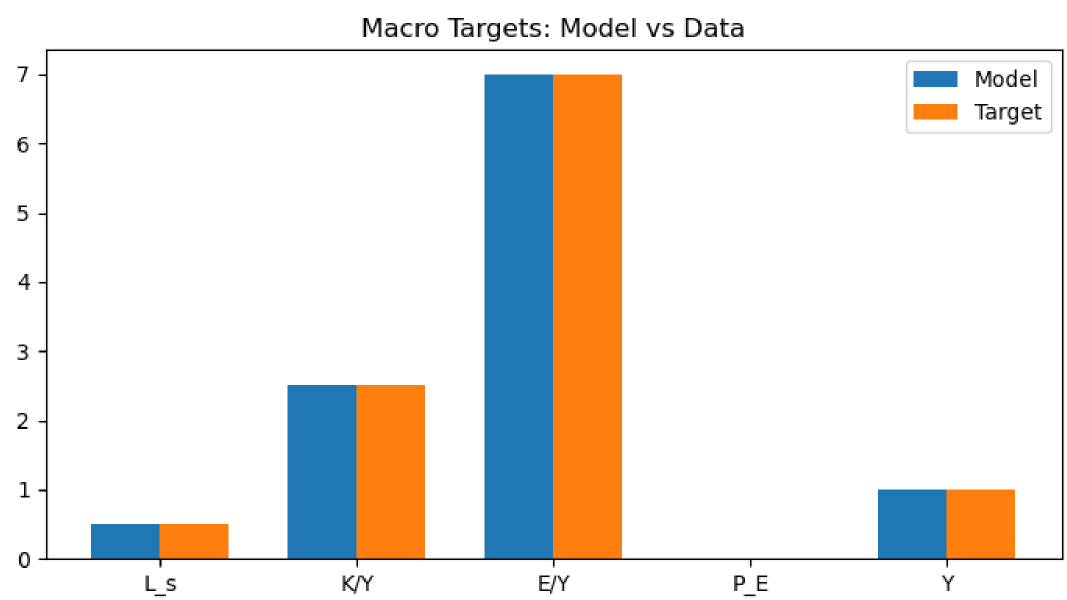
A.2.2 Energy Shares



A.2.3 Welfare Analysis



A.2.4 Macro Targets



B Robustness

B.1 Robustness Tables

B.1.1 Robustness Master Table

Table 9: Comprehensive Robustness Analysis Results: Macroeconomic and Sectoral Variables

Scenario	Param	Tax	Macro Aggregates			Energy & Env.		Prices			Key Ratios				Energy Mix Shares (%)		
Type	Val	τ^* (%)	Y	C	L	E	Z	w	r	P_E	E/Y	K/Y	C/Y	Z/Y	Oil	Gas	Clean
Panel A: Sensitivity to Climate Damages (ω)																	
ω-Robustness	0.04	1.92	0.959	0.973	0.499	4.22	0.019	1.28	0.13	0.02	4.40	2.62	1.01	0.020	0.00	0.59	99.41
ω-Robustness	0.08	1.92	0.959	0.973	0.499	4.22	0.019	1.28	0.13	0.02	4.40	2.62	1.01	0.020	0.00	0.59	99.41
ω-Robustness	0.12	1.92	0.959	0.973	0.499	4.22	0.019	1.28	0.13	0.02	4.40	2.62	1.01	0.020	0.00	0.59	99.41
ω-Robustness	0.16	1.92	0.959	0.973	0.499	4.22	0.019	1.28	0.13	0.02	4.40	2.62	1.01	0.020	0.00	0.59	99.41
ω-Robustness	0.40	1.92	0.959	0.973	0.499	4.22	0.019	1.28	0.13	0.02	4.40	2.62	1.01	0.020	0.00	0.59	99.41
Panel B: Sensitivity to Structural Energy Share (e_c)																	
e_c-Robustness	0.04	2.19	2.042	1.915	0.391	2.41	0.168	3.44	0.23	0.03	1.18	1.23	0.94	0.082	0.96	7.76	91.28
e_c-Robustness	0.06	1.71	2.058	1.967	0.380	3.41	0.742	3.58	0.24	0.04	1.66	1.22	0.96	0.360	7.08	18.94	73.98
e_c-Robustness	0.08	1.99	2.000	1.947	0.362	4.46	0.374	3.66	0.25	0.04	2.23	1.25	0.97	0.187	2.07	8.14	89.79
e_c-Robustness	0.10	0.37	1.317	3.073	0.388	17.85	0.000	0.91	1.08	0.01	13.55	1.90	2.33	0.000	0.00	0.00	100.00
e_c-Robustness	0.12	2.20	0.963	0.972	0.510	4.87	0.058	1.26	0.13	0.02	5.06	2.61	1.01	0.061	0.00	1.54	98.46

Notes: This table presents the complete set of general equilibrium outcomes for each robustness scenario. Panel A tests sensitivity to the climate damage parameter ω (holding $e_c = 0.08$). Panel B tests sensitivity to the energy share parameter e_c (holding $\omega = 0.08$). "Tax" refers to the optimal tax rate τ^*. Y is Output, C Consumption, L Labour, E Energy, Z Emissions. Prices (w, r, P_E) and ratios are endogenous to the model solution.

C Python Code

Listing 1: CGE

```
import numpy as np
import pandas as pd
from scipy.optimize import root, minimize
import matplotlib.pyplot as plt

# Our targets and parameters
L_bar = 1.0
GDP24 = 3.108551
K24   = 7.8 # 7.8
KY_target = K24 / GDP24     # =? 2.51
K_bar = KY_target * L_bar

# Emissions intensities
phi_j = np.array([1.0, 0.7741, 0.0])
omega = 0.08

# Capital shares
a_j = np.array([0.86, 0.86, 0.90])

# Base TFP guesses
A_base = np.array([5.0, 5.0, 12.5])

# Final good cobb douglas shares
b_c = 0.6548
e_c = 0.098
a_c = 1 - b_c - e_c

# Preferences
tau = 0.00231  # initial tax guess
gamma_hh = 2.0
sigma = 0.5
chi_hh = 1.0 # initial guess

# Targets
raw_energy_shares = np.array([0.524, 0.396, 0.080])
TARGET_SHARES = raw_energy_shares / raw_energy_shares.sum()

EY_TARGET = 7.0
L_TARGET  = 0.5 * L_bar
Y_TARGET  = 1.0
TARGET_PRICE = e_c / EY_TARGET

# local solutions functions

def softmax(z):
    z = np.array(z)
    ez = np.exp(z - np.max(z))
    return ez / ez.sum()

def get_allocations(x, current_Ls):
    sK = softmax(x[:4])
    sL = softmax(x[4:8])
    return sK*K_bar, sL*current_Ls

# Production block - Energy and output.

def energy_outputs(K, L, gammas, A_scaler):
    """E_j, dE/dK_j, dE/dL_j"""
```

```python
    A_eff = A_base * A_scaler
    e  = np.zeros(3)
    dK = np.zeros(3)
    dL = np.zeros(3)

    for j in range(3):
        K_j = max(K[j], 1e-9)
        L_j = max(L[j], 1e-9)
        gamma = gammas[j]

        base = A_eff[j] * (K_j**a_j[j]) * (L_j**(1 - a_j[j]))
        e[j] = base**gamma

        dK[j] = gamma * a_j[j] * e[j] / K_j
        dL[j] = gamma * (1 - a_j[j]) * e[j] / L_j

    return e, e.sum(), dK, dL

def final_output(Kc, Lc, E, A_final):
    Kc = max(Kc, 1e-9)
    Lc = max(Lc, 1e-9)
    E  = max(E , 1e-9)
    return A_final * (Kc**a_c) * (Lc**b_c) * (E**e_c)

# HH Block

def household_values(w, r, L_s, e_vec, chi):
    T = tau * np.dot(phi_j, e_vec)
    income = w*L_s + r*K_bar + T
    C = max(income, 1e-9)
    Lrhs = w / (chi * max(C,1e-6)**gamma_hh)
    Lrhs = min(Lrhs, 1e6)
    L_s_foc = max(Lrhs**(1.0/sigma), 1e-9)
    return C, L_s_foc

# Equalibrium System
def ge_system(x, params):
    A_energy = params[:3]
    A_final  = params[3]
    gammas   = params[4:7]
    chi      = params[7]

    r = np.exp(np.clip(x[8], -10, 10))
    w = np.exp(np.clip(x[9], -10, 10))
    L_s = max(x[10], 0.01)

    K_alloc, L_alloc = get_allocations(x[:8], L_s)
    K_e, K_c = K_alloc[:3], K_alloc[3]
    L_e, L_c = L_alloc[:3], L_alloc[3]

    e_vec, E_total, dK_e, dL_e = energy_outputs(K_e, L_e, gammas, A_energy)
    Y = final_output(K_c, L_c, E_total, A_final)

    if E_total > 1e-9:
        P_E = e_c * Y / E_total
    else:
        P_E = 10.0

    C, L_s_foc = household_values(w, r, L_s, e_vec, chi)

    F = np.zeros(11)

```

```
    # Energy FOCs
    for j in range(3):
        p_net = P_E - tau * phi_j[j]
        F[j]     = p_net*dK_e[j] - r
        F[j + 3] = p_net*dL_e[j] - w

    # Final good FOCs
    F[6] = a_c*Y/max(K_c,1e-9) - r
    F[7] = b_c*Y/max(L_c,1e-9) - w

    # Household FOC
    F[10] = L_s - L_s_foc

    return F

def set_tau(new_tau):
    global tau
    tau = new_tau

def compute_welfare(sol, theta):
    x = sol.x
    L_s = max(x[10], 1e-9)
    r   = np.exp(x[8])
    w   = np.exp(x[9])

    # Unpack structural params
    A_energy = theta[:3]
    gammas   = theta[4:7]
    chi      = theta[7]

    # Recover allocations
    K_alloc, L_alloc = get_allocations(x[:8], L_s)
    K_e, K_c = K_alloc[:3], K_alloc[3]
    L_e, L_c = L_alloc[:3], L_alloc[3]

    # Recompute energy outputs to get e_vec
    e_vec, E_total, _, _ = energy_outputs(K_e, L_e, gammas, A_energy)
    Z = np.dot(phi_j, e_vec)  # total emissions

    # C_with_rebate is current consumption including tax rebate
    C_with_rebate, _ = household_values(w, r, L_s, e_vec, chi)

    # Subtract damages from consumption
    C_eff = C_with_rebate - omega * Z
    C_eff = max(C_eff, 1e-9)

    util = (C_eff**(1 - gamma_hh)) / (1 - gamma_hh) \
         - chi * (L_s**(1 + sigma)) / (1 + sigma)

    return float(util)

def welfare_at_tax(tau_value, theta):
    set_tau(tau_value)
    sol = solve_ge(theta)
    if not sol.success:
        return -1e12  # penalize non-convergence
    return compute_welfare(sol, theta)

def welfare_grid(theta, tau_min=0, tau_max=200, n=80):
    taus = np.linspace(tau_min, tau_max, n)
    welfare_vals = []
    for t in taus:
```

```python
        W = welfare_at_tax(t, theta)
        welfare_vals.append(W)
        print(f"tau={t:.2f}, Welfare={W:.6f}")
    return taus, np.array(welfare_vals)

def solve_ge(theta):
    """Try LM first; if it fails, fall back to HYBR."""
    x0 = np.zeros(11)
    x0[8]  = np.log(0.2)
    x0[9]  = np.log(2.0)
    x0[10] = 0.3

    params = theta
    sol = root(ge_system, x0, args=(params,), method="lm", tol=1e-7)

    if not sol.success:
        sol = root(ge_system, x0, args=(params,), method="hybr", tol=1e-7)

    return sol

# Threaded calibration loss with jittered restart for new local loss solution.

Nfeval = 1

def calibration_loss(theta):
    global Nfeval

    A_energy = theta[:3]
    A_final  = theta[3]
    gammas   = theta[4:7]
    chi      = theta[7]

    sol = solve_ge(theta)
    if not sol.success:
        return 1e9

    L_s = max(sol.x[10], 1e-9)
    K_alloc, L_alloc = get_allocations(sol.x[:8], L_s)

    e_vec, E_total, _, _ = energy_outputs(K_alloc[:3], L_alloc[:3], gammas, A_energy)
    if E_total <= 1e-9:
        return 1e9

    Y = final_output(K_alloc[3], L_alloc[3], E_total, A_final)
    if Y <= 1e-9:
        return 1e9

    model_shares = e_vec / E_total
    price_model = e_c * Y / E_total
    EY_model = E_total / Y
    KY_model = K_bar / Y

    share_err = np.sum((model_shares - TARGET_SHARES)**2)
    price_err = (price_model - TARGET_PRICE)**2
    L_err     = (L_s - L_TARGET)**2
    Y_err     = (Y - Y_TARGET)**2
    KY_err    = (KY_model - KY_target)**2
    EY_err    = (EY_model - EY_TARGET)**2
    reg_err = 0.1 * np.sum((gammas - 0.9)**2)

    loss = ( #adjust weights as needed
        25*share_err +
```

```
            25*price_err +
            150*L_err +
            25*Y_err +
            25*EY_err +
            25*KY_err +
            25*reg_err
    )

    if Nfeval % 50 == 0:
        print(f"[{Nfeval}] Loss={loss:.4f}  Y={Y:.3f}  L={L_s:.3f}  P_E={price_model:.3f}")

    Nfeval += 1
    return loss

def run_calibration_loop(start_theta=None):
    global Nfeval

    # Use the passed starting point if available, otherwise use the hardcoded guess
    if start_theta is not None:
        theta_best = np.array(start_theta)
        print(">>> Starting calibration from previous best theta...")
    else:
        # Default starting guess
        theta_best = np.array([
            5.0, 5.0, 1.2,
            3.0,
            0.85, 0.83, 0.89,
            1.5
        ])

    best_loss = 1e9
    num_restarts = 1  # Reduce restarts for robustness speed

    # Check if our starting point is already good enough
    initial_loss = calibration_loss(theta_best)
    if initial_loss < 0.01:
        print(f"Initial theta is already good (Loss={initial_loss:.6f}). Skipping loop.")
        return theta_best

    bounds = [
        (1, 10), (1, 10), (1, 5),
        (0.5, 10.0),
        (0.7, 1.0), (0.7, 1.0), (0.7, 1.0),
        (0.1, 5.0)
    ]

    for attempt in range(1, num_restarts + 1):
        if attempt == 1:
            start = theta_best
        else:
            jitter = np.random.uniform(0.99, 1.01, size=len(theta_best)) # Smaller jitter
            start = theta_best * jitter

        res = minimize(
            calibration_loss,
            start,
            method='L-BFGS-B',
            bounds=bounds,
            options={'maxiter': 2000, 'ftol': 1e-6} # Reduced tolerance for speed
        )

        loss = res.fun
```

```python
        if loss < best_loss:
            best_loss = loss
            theta_best = res.x

    return theta_best

# Reporting function

def print_final_report(theta):
    """
    Print a structured calibration report and produce summary plots.
    """
    import matplotlib.pyplot as plt

    # Unpack parameters
    A_energy = theta[:3]
    A_final  = theta[3]
    gammas   = theta[4:7]
    chi      = theta[7]

    # Solve GE at a calibrated theta
    sol = solve_ge(theta)
    if not sol.success:
        print("\nWARNING: GE solver did NOT CONVERGE at these parameters.")
    x = sol.x

    L_s = max(x[10], 1e-9)
    r   = np.exp(x[8])
    w   = np.exp(x[9])

    # allocations
    K_alloc, L_alloc = get_allocations(x[:8], L_s)
    K_e, K_c = K_alloc[:3], K_alloc[3]
    L_e, L_c = L_alloc[:3], L_alloc[3]

    # energy and final output
    e_vec, E_total, dK_e, dL_e = energy_outputs(K_e, L_e, gammas, A_energy)
    Y = final_output(K_c, L_c, E_total, A_final)
    total_emissions = np.dot(phi_j, e_vec)
    emissions_intensity = total_emissions / Y

    # prices and household
    P_E = e_c * Y / E_total

    # compute household values WITH rebate
    C_with_rebate, _ = household_values(w, r, L_s, e_vec, chi)

    # Compute tax revenues
    tax_revenue = tau * np.dot(phi_j, e_vec)

    # Compute consumption WITHOUT tax rebate
    C_no_rebate = w * L_s + r * K_bar


    # key ratios
    KY_model = K_bar / Y
    EY_model = E_total / Y
    model_shares = e_vec / E_total
    energy_spend_share = (P_E * E_total) / Y   # should be close to e_c
    # Run welfare grid around the calibrated parameters

    # recompute objective for context
```

```
    current_loss = calibration_loss(theta)

    # Compute taxes
    tax_revenue = tau * np.dot(phi_j, e_vec)

    # Consumption WITHOUT rebate
    C_no_rebate = w * L_s + r * K_bar

    # Consumption WITH rebate
    C_with_rebate = C_no_rebate + tax_revenue
    # Welfare at the calibrated baseline tau

    # Continuous optimum BEFORE macro_data
    obj = lambda t: -welfare_at_tax(t[0], theta)
    taus = np.linspace(0.0, 0.05, 40)
    Wvals = np.array([welfare_at_tax(t, theta_ec) for t in taus])
    tau0 = taus[np.argmax(Wvals)]

    res = minimize(
        lambda t: -welfare_at_tax(t[0], theta_ec),
        x0=[tau0],
        bounds=[(0.0, 0.05)]
    )
    tau_star = res.x[0]
    W_star = welfare_at_tax(tau_star_cont, theta)

    # Macro Table
    macro_data = [
        ("Taxes collected T",      tax_revenue,    None),
        ("Consumption (no rebate)", C_no_rebate,   None),
        ("Consumption (with rebate)", C_with_rebate, None),
        ("Output Y",            Y,          Y_TARGET),
        ("Labour L_s",          L_s,        L_TARGET),
        ("K/Y",                 KY_model,   KY_target),
        ("E/Y",                 EY_model,   EY_TARGET),
        ("Energy price P_E",    P_E,        TARGET_PRICE),
        ("Energy spend share",  energy_spend_share, e_c),
        ("Wage w",              w,          None),
        ("Rental rate r",       r,          None),
        ("Welfare at tau*",     W_star,              None),
        ("Net Emissions", total_emissions, None),
        ("Emissions intensity (per unit Y)", emissions_intensity, None),
    ]

    rows = []
    for name, model_val, target_val in macro_data:
        if target_val is None or target_val == 0:
            abs_err = None
            rel_err = None
        else:
            abs_err = model_val - target_val
            rel_err = 100 * abs_err / target_val
        rows.append({
            "Variable":   name,
            "Model":      model_val,
            "Target":     target_val,
            "Abs_Error":  abs_err,
            "Rel_Error_%": rel_err
        })

    df_macro = pd.DataFrame(rows)
```

```python
    # Energy share Table
    share_rows = []
    labels_energy = ["Oil", "Gas", "Clean"]
    for j in range(3):
        m = model_shares[j]
        t = TARGET_SHARES[j]
        abs_err = m - t
        rel_err = 100 * abs_err / t if t != 0 else None
        share_rows.append({
            "Tech":        labels_energy[j],
            "Model_Share": m,
            "Target_Share": t,
            "Abs_Error":   abs_err,
            "Rel_Error_%": rel_err
        })
    df_shares = pd.DataFrame(share_rows)

    # Sector Table
    total_K = K_alloc.sum()
    total_L = L_alloc.sum()
    sector_labels = ["Oil", "Gas", "Clean", "Final"]

    sector_outputs = list(e_vec) + [Y]
    sector_rows = []
    for i in range(4):
        K_i = K_alloc[i]
        L_i = L_alloc[i]
        Y_i = sector_outputs[i]
        sector_rows.append({
            "Sector":        sector_labels[i],
            "K":             K_i,
            "L":             L_i,
            "Output":        Y_i,
            "K_Share":       K_i / total_K,
            "L_Share":       L_i / total_L
        })
    df_sector = pd.DataFrame(sector_rows)

    # Paramater table
    df_params = pd.DataFrame([
        {"Parameter":"A_oil",        "Value":A_energy[0], "Type":"TFP (energy)"},
        {"Parameter":"A_gas",        "Value":A_energy[1], "Type":"TFP (energy)"},
        {"Parameter":"A_clean",      "Value":A_energy[2], "Type":"TFP (energy)"},
        {"Parameter":"A_final",      "Value":A_final,     "Type":"TFP (final)"},
        {"Parameter":"gamma_oil",    "Value":gammas[0],   "Type":"returns to scale"},
        {"Parameter":"gamma_gas",    "Value":gammas[1],   "Type":"returns to scale"},
        {"Parameter":"gamma_clean",  "Value":gammas[2],   "Type":"returns to scale"},
        {"Parameter":"chi_hh",       "Value":chi,         "Type":"preference (labour disutility)"},
        {"Parameter":"a_c",          "Value":a_c,         "Type":"capital share final"},
        {"Parameter":"b_c",          "Value":b_c,         "Type":"labour share final"},
        {"Parameter":"e_c",          "Value":e_c,         "Type":"energy share final"},

    ])

    # prints report
    pd.set_option("display.float_format", lambda v: f"{v: .4f}")

    print("\n=================================================================")
    print(" FINAL CALIBRATION REPORT")
    print("=================================================================")
    print(f"Calibration loss at theta: {current_loss:.6f}")
    print("Solver success:", sol.success)
```

```python

    print("\n--- Calibrated Parameters ---")
    print(df_params.to_string(index=False))

    print("\n--- Macro Outcomes vs Targets ---")
    print(df_macro.to_string(index=False))

    print("\n--- Energy Shares (Model vs Target) ---")
    print(df_shares.to_string(index=False))

    print("\n--- Sectoral Allocations ---")
    print(df_sector.to_string(index=False))

    # PLOTS

    # 1) Energy shares bar chart
    plt.figure(figsize=(6,4))
    x = np.arange(3)
    width = 0.35
    plt.bar(x - width/2, model_shares, width, label="Model")
    plt.bar(x + width/2, TARGET_SHARES, width, label="Target")
    plt.xticks(x, labels_energy)
    plt.ylabel("Share of total energy")
    plt.title("Energy Shares: Model vs Target")
    plt.legend()
    plt.tight_layout()
    plt.show()

    # 2) Macro fit vs targets
    macro_names = ["L_s", "K/Y", "E/Y", "P_E", "Y"]
    macro_model_vals  = [L_s, KY_model, EY_model, P_E, Y]
    macro_target_vals = [L_TARGET, KY_target, EY_TARGET, TARGET_PRICE, Y_TARGET]

    plt.figure(figsize=(7,4))
    x2 = np.arange(len(macro_names))
    width2 = 0.35
    plt.bar(x2 - width2/2, macro_model_vals, width2, label="Model")
    plt.bar(x2 + width2/2, macro_target_vals, width2, label="Target")
    plt.xticks(x2, macro_names)
    plt.title("Macro Targets: Model vs Data")
    plt.legend()
    plt.tight_layout()
    plt.show()

    # 3) Sectoral outputs
    plt.figure(figsize=(6,4))
    plt.bar(sector_labels, sector_outputs)
    plt.ylabel("Output")
    plt.title("Sectoral Outputs (Energy and Final Good)")
    plt.tight_layout()
    plt.show()
    # 4) Welfare grid
    print("\nComputing welfare grid...")
    taus, Ws = welfare_grid(theta, tau_min=0.00221, tau_max=0.00241, n=50)
    tau_star = taus[np.argmax(Ws)]
    print(f"\nGrid-search optimal tau* =? {tau_star:.3f}")

    #   welfare sensuety around tau
    print("\nPlotting welfare sensitivity around tau* ...")

    # Define a symmetric grid around tau*
    span = 0.01      # 20% on either side
```

```
    tau_low  = max(0.0, tau_star_cont * (1 - span))
    tau_high = tau_star_cont * (1 + span)

    # Build grid
    taus = np.linspace(tau_low, tau_high, 10)
    welfare_vals = []

    for t in taus:
        W = welfare_at_tax(t, theta)
        welfare_vals.append(W)

    welfare_vals = np.array(welfare_vals)

    # Identify max on this local grid
    idx_star = np.argmax(welfare_vals)
    tau_star_grid = taus[idx_star]
    W_star_grid = welfare_vals[idx_star]

    # Plot
    plt.figure(figsize=(8,5))
    plt.plot(taus, welfare_vals, linewidth=3)
    plt.scatter([tau_star_grid], [W_star_grid], color='black', s=60)

    plt.axvline(tau_star_cont, linestyle='--', linewidth=2, label=f"Continuous optimum tau*={tau_star_cont:.5f}")

    plt.title("Welfare Sensitivity Around Optimal Tax Rate", fontsize=14)
    plt.xlabel("Energy Tax tau")
    plt.ylabel("Welfare (Utility)")
    plt.grid(True, alpha=0.4)
    plt.legend()
    plt.tight_layout()
    plt.show()

    print(f"\nLocal grid optimum tau =? {tau_star_grid:.6f}")
    print(f"Continuous optimum tau* =? {tau_star_cont:.6f}")
    print(f"Welfare difference W(tau_grid*) - W(tau*) = {W_star_grid - welfare_at_tax(tau_star_cont, theta):.6e}")

    # Continuous optimum
    print(f"\nContinuous optimum tau* = {tau_star_cont:.3f}")

def simple_tax_grid(theta):
    grid = np.linspace(0.0, 0.02, 50)

    best_tau = 0.0
    best_w = -1e12

    for t in grid:
        w = welfare_at_tax(t, theta)
        if w > best_w and w != -1e12:
            best_w = w
            best_tau = t

    # Stage 2: refine very closely around the winner
    if best_tau > 0:
        span = 0.002 # +/- 0.2%
        fine_grid = np.linspace(max(0, best_tau - span), best_tau + span, 40)
        for t in fine_grid:
            w = welfare_at_tax(t, theta)
            if w > best_w:
                best_w = w
```

```
                best_tau = t

    return best_tau, best_w

def get_full_macro_state(theta, tau_val, scenario_name, param_val):
    """
    Solves the model at a specific tax and returns a dictionary of ALL variables.
    """
    # Set the tax and solve
    set_tau(tau_val)
    sol = solve_ge(theta)

    if not sol.success:
        return None # Skip failed runs

    # Extract Basic Variables
    x = sol.x
    r   = np.exp(x[8])
    w   = np.exp(x[9])
    L_s = max(x[10], 1e-9)

    # Allocations
    K_alloc, L_alloc = get_allocations(x[:8], L_s)
    K_e, K_c = K_alloc[:3], K_alloc[3]
    L_e, L_c = L_alloc[:3], L_alloc[3]

    # Production & Energy
    A_energy = theta[:3]
    gammas   = theta[4:7]
    A_final  = theta[3]
    chi_val  = theta[7]

    e_vec, E_total, _, _ = energy_outputs(K_e, L_e, gammas, A_energy)
    Y = final_output(K_c, L_c, E_total, A_final)

    # Emissions
    Z = np.dot(phi_j, e_vec)  # Total emissions

        # Consumption & Government
    # Tax Revenue = tau * Emissions (since tax is on emissions intensity phi * e)
    Tax_Rev = tau_val * Z

    # Household Income = wL + rK + T
    Income = w * L_s + r * K_bar + Tax_Rev
    C = Income # Market clearing C = Income

    # Prices
    if E_total > 1e-9:
        P_E = e_c * Y / E_total # Energy price from FOC
    else:
        P_E = 0.0

    # Welfare
    # Recalculate utility with the specific damage parameter (global omega)
    C_eff = max(C - omega * Z, 1e-9)
    Util  = (C_eff**(1 - gamma_hh)) / (1 - gamma_hh) - chi_val * (L_s**(1 + sigma)) / (1 + sigma)

    # Return EVERYTHING in a dictionary
    return {
        "Scenario": scenario_name,
        "Param_Value": param_val,
        "Opt_Tax_Rate_%": tau_val * 100,
```

```python

        # Macro Aggregates
        "Output_Y": Y,
        "Consumption_C": C,
        "Labour_L": L_s,
        "Energy_E": E_total,
        "Emissions_Z": Z,
        "Tax_Revenue": Tax_Rev,

        # Prices
        "Wage_w": w,
        "Rental_r": r,
        "Energy_Price_P_E": P_E,

        # Ratios
        "E_over_Y": E_total / Y if Y > 0 else 0,
        "K_over_Y": K_bar / Y if Y > 0 else 0,
        "C_over_Y": C / Y if Y > 0 else 0,
        "TaxRev_over_Y_%": (Tax_Rev / Y)*100 if Y > 0 else 0,
        "Emissions_Intensity_Z_Y": Z / Y if Y > 0 else 0,

        # Welfare
        "Welfare_Util": Util,

        # Sectoral Shares (Energy Mix)
        "Oil_Share_%": (e_vec[0]/E_total)*100 if E_total > 0 else 0,
        "Gas_Share_%": (e_vec[1]/E_total)*100 if E_total > 0 else 0,
        "Clean_Share_%": (e_vec[2]/E_total)*100 if E_total > 0 else 0,
    }

def get_baseline_equilibrium(theta):
    sol = solve_ge(theta)
    if not sol.success:
        return None

    x = sol.x
    L_s = max(x[10], 1e-9)
    r   = np.exp(x[8])
    w   = np.exp(x[9])

    K_alloc, L_alloc = get_allocations(x[:8], L_s)
    K_e, K_c = K_alloc[:3], K_alloc[3]
    L_e, L_c = L_alloc[:3], L_alloc[3]

    e_vec, E_total, _, _ = energy_outputs(
        K_e, L_e,
        theta[4:7],
        theta[:3]
    )

    Y = final_output(K_c, L_c, E_total, theta[3])
    Z = np.dot(phi_j, e_vec)
    P_E = e_c * Y / E_total if E_total > 0 else np.nan
    tax_rev = tau * Z

    return {
        "omega": omega,
        "e_c": e_c,
        "tax": tau * 100,
        "Y": Y,
        "L": L_s,
        "E": E_total,
```

```python
        "Emissions": Z,
        "Energy_Price": P_E,
        "Tax_Revenue": tax_rev
    }

def run_robustness_analysis(baseline_theta):
    global omega, e_c

    base_omega = omega
    base_e_c   = e_c

    omega_range = np.array([0.5, 1.0, 1.5, 2.0, 5.0]) * base_omega
    ec_range    = np.linspace(0.03, 0.15, 25)

    omega_rows = []
    ec_rows    = []

    # baseline equilibrium
    baseline_row = get_baseline_equilibrium(baseline_theta)

    # omega robustness
    for val in omega_range:
        omega = val
        e_c   = base_e_c

        tau_star, _ = simple_tax_grid(baseline_theta)
        set_tau(tau_star)
        sol = solve_ge(baseline_theta)

        if not sol.success:
            continue

        x = sol.x
        L_s = max(x[10], 1e-9)
        K_alloc, L_alloc = get_allocations(x[:8], L_s)

        e_vec, E_total, _, _ = energy_outputs(
            K_alloc[:3], L_alloc[:3],
            baseline_theta[4:7],
            baseline_theta[:3]
        )

        Y = final_output(
            K_alloc[3], L_alloc[3],
            E_total,
            baseline_theta[3]
        )

        omega_rows.append({
            "omega": val,
            "tax": tau_star * 100,
            "Y": Y
        })

    # energy share robustness
    for val in ec_range:
        e_c   = val
        omega = base_omega

        tau_star, _ = simple_tax_grid(baseline_theta)
        set_tau(tau_star)
        sol = solve_ge(baseline_theta)
```

```python

        if not sol.success:
            continue

        x = sol.x
        L_s = max(x[10], 1e-9)
        K_alloc, L_alloc = get_allocations(x[:8], L_s)

        e_vec, E_total, _, _ = energy_outputs(
            K_alloc[:3], L_alloc[:3],
            baseline_theta[4:7],
            baseline_theta[:3]
        )

        Y = final_output(
            K_alloc[3], L_alloc[3],
            E_total,
            baseline_theta[3]
        )

        ec_rows.append({
            "e_c": val,
            "tax": tau_star * 100,
            "Y": Y
        })

    omega = base_omega
    e_c   = base_e_c

    df_baseline = pd.DataFrame([baseline_row])
    df_omega    = pd.DataFrame(omega_rows)
    df_ec       = pd.DataFrame(ec_rows)

    return df_baseline, df_omega, df_ec

# ROBUSTNESS MODULE

def welfare_at_tax_with_damage(tau_value, theta, omega):
    """
    Welfare with damage.
    """
    set_tau(tau_value)
    sol = solve_ge(theta)
    if not sol.success:
        return -1e12

    x = sol.x
    L_s = max(x[10], 1e-9)
    r   = np.exp(x[8])
    w   = np.exp(x[9])

    # allocations
    K_alloc, L_alloc = get_allocations(x[:8], L_s)
    K_e, K_c = K_alloc[:3], K_alloc[3]
    L_e, L_c = L_alloc[:3], L_alloc[3]

    # energy
    e_vec, E_total, _, _ = energy_outputs(K_e, L_e, theta[4:7], theta[:3])
    Z = np.dot(phi_j, e_vec)

    # household utility (with rebate)
    C, _ = household_values(w, r, L_s, e_vec, theta[7])
```

```python

    utility = (C**(1 - gamma_hh)) / (1 - gamma_hh) \
              - theta[7] * (L_s**(1 + sigma)) / (1 + sigma) \
              - omega * Z

    return float(utility)


def optimal_tau_for_omega(theta, omega,
                          tau_bounds=(0.0, 0.05),
                          grid_n=40):
    """
    Finds tau* for a given omega using grid + local refinement.
    """
    taus = np.linspace(tau_bounds[0], tau_bounds[1], grid_n)
    Wvals = np.array([
        welfare_at_tax_with_damage(t, theta, omega) for t in taus
    ])

    tau0 = taus[np.argmax(Wvals)]

    obj = lambda t: -welfare_at_tax_with_damage(t[0], theta, omega)

    res = minimize(obj, x0=[tau0], bounds=[tau_bounds], tol=1e-8)
    return res.x[0], -res.fun


def run_omega_robustness(theta):
    omega_grid = [0.04, 0.08, 0.12, 0.16, 0.40]
    rows = []

    for omega in omega_grid:
        tau_star, W_star = optimal_tau_for_omega(theta, omega)
        set_tau(tau_star)
        sol = solve_ge(theta)

        if not sol.success:
            continue

        x = sol.x
        L_s = x[10]
        K_alloc, L_alloc = get_allocations(x[:8], L_s)
        e_vec, E, _, _ = energy_outputs(
            K_alloc[:3], L_alloc[:3], theta[4:7], theta[:3]
        )
        Y = final_output(K_alloc[3], L_alloc[3], E, theta[3])
        Z = np.dot(phi_j, e_vec)

        rows.append({
            "omega": omega,
            "tau_star": tau_star,
            "Y": Y,
            "E": E,
            "Z": Z,
            "W": W_star
        })

    return pd.DataFrame(rows)


def reset_model_state():
    global tau, Nfeval
```

```python
    tau = 0.002
    Nfeval = 1

def run_ec_robustness(ec_values):
    """
    Full structural robustness to energy share.
    Recalibrates model for each e_c.
    """
    global e_c, a_c

    results = []

    for ec_val in ec_values:
        reset_model_state()
        e_c = ec_val
        a_c = 1 - b_c - e_c
        # Recalibrate
        theta_ec = run_calibration_loop()

        # Find optimal tax
        obj = lambda t: -welfare_at_tax(t[0], theta_ec)
        taus = np.linspace(0.0, 0.05, 40)
        Wvals = np.array([welfare_at_tax(t, theta_ec) for t in taus])
        tau0 = taus[np.argmax(Wvals)]

        res = minimize(
            lambda t: -welfare_at_tax(t[0], theta_ec),
            x0=[tau0],
            bounds=[(0.0, 0.05)]
        )
        tau_star = res.x[0]
        # Solve GE at tau*
        set_tau(tau_star)
        sol = solve_ge(theta_ec)
        if not sol.success:
            continue

        x = sol.x
        L_s = x[10]
        K_alloc, L_alloc = get_allocations(x[:8], L_s)
        e_vec, E, _, _ = energy_outputs(
            K_alloc[:3], L_alloc[:3], theta_ec[4:7], theta_ec[:3]
        )
        Y = final_output(K_alloc[3], L_alloc[3], E, theta_ec[3])
        Z = np.dot(phi_j, e_vec)

        results.append({
            "e_c": ec_val,
            "tau_star": tau_star,
            "Y": Y,
            "E": E,
            "Z": Z,
            "E_Y": E / Y,
            "Clean_share": e_vec[2] / E
        })

    return pd.DataFrame(results)


MASTER_COLS = [
    "Scenario", "Param", "Tax_%", "Y", "C", "L", "E", "Z",
    "w", "r", "P_E", "E/Y", "K/Y", "C/Y", "Z/Y",
```

```python
    "Oil_%", "Gas_%", "Clean_%"
]

def build_master_row(theta, tau_star, scenario, param_value):
    set_tau(tau_star)
    sol = solve_ge(theta)
    if not sol.success:
        raise RuntimeError("GE failed")

    x = sol.x
    L_s = max(x[10], 1e-9)
    r = np.exp(x[8])
    w = np.exp(x[9])

    K_alloc, L_alloc = get_allocations(x[:8], L_s)
    K_e, K_c = K_alloc[:3], K_alloc[3]
    L_e, L_c = L_alloc[:3], L_alloc[3]

    e_vec, E, _, _ = energy_outputs(K_e, L_e, theta[4:7], theta[:3])
    Y = final_output(K_c, L_c, E, theta[3])
    Z = np.dot(phi_j, e_vec)

    P_E = e_c * Y / E
    C, _ = household_values(w, r, L_s, e_vec, theta[7])

    shares = e_vec / E

    return {
        "Scenario": scenario,
        "Param": param_value,
        "Tax_%": 100 * tau_star,
        "Y": Y,
        "C": C,
        "L": L_s,
        "E": E,
        "Z": Z,
        "w": w,
        "r": r,
        "P_E": P_E,
        "E/Y": E / Y,
        "K/Y": K_bar / Y,
        "C/Y": C / Y,
        "Z/Y": Z / Y,
        "Oil_%": 100 * shares[0],
        "Gas_%": 100 * shares[1],
        "Clean_%": 100 * shares[2],
    }


def print_master_robustness_table(omega_grid, ec_grid):
    global e_c, a_c

    rows_A = []
    rows_B = []

    # Omega robustness
    e_c = 0.08
    a_c = 1 - b_c - e_c
    theta_base = run_calibration_loop()

    for omega in omega_grid:
        def welfare_omega(t):
```

```
            set_tau(t)
            sol = solve_ge(theta_base)
            if not sol.success:
                return -1e12
            x = sol.x
            L_s = max(x[10], 1e-9)
            r = np.exp(x[8])
            w = np.exp(x[9])
            K_alloc, L_alloc = get_allocations(x[:8], L_s)
            e_vec, _, _, _ = energy_outputs(
                K_alloc[:3], L_alloc[:3], theta_base[4:7], theta_base[:3]
            )
            Z = np.dot(phi_j, e_vec)
            C, _ = household_values(w, r, L_s, e_vec, theta_base[7])
            return (
                (C**(1 - gamma_hh)) / (1 - gamma_hh)
                - theta_base[7] * (L_s**(1 + sigma)) / (1 + sigma)
                - omega * Z
            )

        taus = np.linspace(0, 0.05, 40)
        tau_star = taus[np.argmax([welfare_omega(t) for t in taus])]

        rows_A.append(
            build_master_row(theta_base, tau_star, "omega-Robustness", omega)
        )

    #  e_c robustness
    for ec_val in ec_grid:
        e_c = ec_val
        a_c = 1 - b_c - e_c
        theta_ec = run_calibration_loop()

        taus = np.linspace(0, 0.05, 40)
        tau_star = taus[np.argmax([welfare_at_tax(t, theta_ec) for t in taus])]

        rows_B.append(
            build_master_row(theta_ec, tau_star, "e_c-Robustness", ec_val)
        )

    df_A = pd.DataFrame(rows_A)[MASTER_COLS]
    df_B = pd.DataFrame(rows_B)[MASTER_COLS]

    print("\n================ ROBUSTNESS MASTER TABLE ================\n")
    print("Panel A: Sensitivity to Climate Damages (omega)\n")
    print(df_A.to_string(index=False))

    print("\nPanel B: Sensitivity to Structural Energy Share (e_c)\n")
    print(df_B.to_string(index=False))

    return df_A, df_B

if __name__ == "__main__":

    print("\n================ ROBUSTNESS ANALYSIS =================")

    # Baseline calibrated parameters
    theta_base = run_calibration_loop()

    #  Omega robustness
    omega_df = run_omega_robustness(theta_base)
    print("\nOmega Robustness Results")
```

```
    print(omega_df)

    #  e_c robustness
    ec_grid = [0.04, 0.06, 0.08, 0.10, 0.12]
    ec_df = run_ec_robustness(ec_grid)
    print("\nEnergy Share Robustness Results")
    print(ec_df)

    print("\n================ END ROBUSTNESS =================")

    print_master_robustness_table(
        omega_grid=[0.04, 0.08, 0.12, 0.16, 0.40],
        ec_grid=[0.04, 0.06, 0.08, 0.10, 0.12]
    )

    print(f"Starting baseline calibration with omega = {omega}...")
    best_theta = run_calibration_loop()

    df_baseline, df_omega, df_ec = run_robustness_analysis(best_theta)

    print("\nBaseline equilibrium")
    print(df_baseline.to_string(index=False))

    # structural output response
    plt.figure(figsize=(8, 5))
    plt.plot(df_ec["e_c"], df_ec["Y"], marker="o", linewidth=2.5)
    plt.xlabel("Energy share e_c")
    plt.ylabel("Output Y")
    plt.tight_layout()
    plt.show()

    # robustness panels
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))

    ax1.plot(df_omega["omega"], df_omega["tax"], marker="s", linewidth=3)
    ax1.set_xlabel("omega")
    ax1.set_ylabel("optimal tax (%)")
    ax1.grid(True, alpha=0.6)

    ax2.plot(df_ec["e_c"], df_ec["tax"], marker="o", linewidth=2)
    ax2.set_xlabel("energy share e_c")
    ax2.set_ylabel("optimal tax (%)")
    ax2.grid(True, alpha=0.6)

    plt.tight_layout()
    plt.show()
```

END OF DOCUMENT