Appendix B: Proof-of-Concept Hierarchical Bayesian Implementation

(This demonstrates the advanced methods referenced in Section 6.5 of the main paper)

DISCLAIMER: This appendix presents a **proof-of-concept implementation** of the hierarchical Bayesian framework described conceptually in the main text. This is **not** a real data analysis or risk assessment. All data are **simulated for demonstration purposes only**, all parameters are **hypothetical**, and the model is **simplified for computational tractability**. Real applications would require extensive validation, empirical data, and more sophisticated implementation.

B.1 Framework Overview: From Simple to Complex

To bridge from Appendix A's simplified Bayes factor approach to full hierarchical modeling, we present a **scalable implementation strategy**:

Three Tiers of Implementation:

1. **Tier 1:** Simplified Bayes factor combination (Appendix A)
2. **Tier 2:** Hierarchical model with shared pathways (this appendix)
3. **Tier 3:** Full measurement-error-corrected implementation (conceptual)

This appendix demonstrates Tier 2 — showing **how** hierarchical Bayesian modeling works, not producing real MNPs-cancer estimates.

B.2 Simplified Hierarchical Model Structure

Note: We use a **deliberately simplified** version of the full framework for clarity and computational stability.

Core Model Components:

1. **Shared Pathway Variables** (θ_1, θ_2, θ_3):
    - Represent hypothetical biological pathways
    - **Illustrative only** — real pathways would require biological validation
2. **Evidence Stream Effects** (β_mech, β_animal, β_human, β_epi):
    - Model evidence strength for each stream
    - **Hypothetical values** for demonstration
3. **Measurement Model** (η, W, Z):
    - Simplified version of measurement error correction
    - **Demonstrates concept** only

Key Simplifications for This Proof-of-Concept:

- Reduced parameter dimensions
- Simpler prior specifications
- Limited correlation structure
- Simulated data only

B.3 Simulated Data for Demonstration [available at GitHub Repository. URL:

https://github.com/BARC-commits/BARC_Bayesian-Assessment-of-Research-Causality]

CRITICAL NOTE: All data below are **COMPLETELY SIMULATED** for demonstration.

```
"""
DATA SIMULATION FOR DEMONSTRATION PURPOSES ONLY
No real MNPs or cancer data are used in this appendix
"""

import numpy as np

# Set seed for reproducibility
np.random.seed(42)

# Simulated sample sizes (illustrative)
N_mech = 50    # Hypothetical mechanistic studies
N_animal = 30  # Hypothetical animal studies
N_human = 100  # Hypothetical human samples
N_epi = 500    # Hypothetical epidemiological observations
```

```
# Simulate EXPOSURE DATA (completely hypothetical)
dose_mech = np.random.exponential(scale=1.0, size=N_mech)
dose_animal = np.random.exponential(scale=2.0, size=N_animal)

# Simulate BIOMARKER DATA with measurement error (hypothetical)
eta_true = np.random.randn(N_human) * 0.7  # True (unobserved) exposure
W_obs = np.random.lognormal(mean=eta_true, sigma=0.25, size=N_human)  # Noisy biomarker
Z_obs = np.random.lognormal(mean=0.05 + 0.85*eta_true, sigma=0.4, size=N_human)  # Environmental proxy

# Simulate OUTCOME DATA (completely hypothetical)
y_mech = 0.5 + 0.25*dose_mech + np.random.randn(N_mech)*0.4
mu_animal = np.exp(0.3 + 0.35*dose_animal)
y_animal = np.random.poisson(lam=mu_animal, size=N_animal)
y_human = 0.4 + 0.5*eta_true + np.random.randn(N_human)*0.5

# Simulate EPIDEMIOLOGICAL DATA (hypothetical)
exposure_epi = np.random.randn(N_epi)
log_odds = -2.1 + 0.4*exposure_epi
```

```
p_case = 1 / (1 + np.exp(-log_odds))

y_epi = np.random.binomial(n=1, p=p_case, size=N_epi)
```

Reiteration: These are **NOT real data**. They demonstrate **how simulated data would feed into the model**.

B.4 Proof-of-Concept PyMC Implementation [1] [available at GitHub Repository. URL: https://github.com/BARC-commits/BARC_Bayesian-Assessment-of-Research-Causality]

```
"""
PROOF-OF-CONCEPT IMPLEMENTATION ONLY
This demonstrates MODEL STRUCTURE, not real analysis
All outputs are from SIMULATED DATA only
"""

import pymc as pm
import pytensor.tensor as pt
import arviz as az

print("BARC Appendix B: Proof-of-Concept Hierarchical Bayesian Implementation")
print("=" * 80)
print("DISCLAIMER: This is a DEMONSTRATION using SIMULATED DATA")
print("=" * 80)

with pm.Model() as barc_demo_model:

    # ==================== DEMONSTRATION PARAMETERS ====================
    # All parameters are FOR ILLUSTRATION ONLY
```

```
# 1. Shared Pathway Variables (hypothetical)

# These represent ILLUSTRATIVE biological pathways

sigma_theta = pm.HalfNormal("sigma_theta", sigma=1.0, shape=3)

theta_0 = pm.Normal("theta_0", mu=0, sigma=sigma_theta[0])

theta_1 = pm.Normal("theta_1", mu=0, sigma=sigma_theta[1])

theta_2 = pm.Normal("theta_2", mu=0, sigma=sigma_theta[2])

theta = pm.Deterministic("theta", pt.stack([theta_0, theta_1, theta_2]))



# 2. Measurement Error Model (simplified demonstration)

eta = pm.Normal("eta", mu=0.0, sigma=1.0, shape=N_human)

sigma_W = pm.HalfNormal("sigma_W", sigma=0.3)

sigma_Z = pm.HalfNormal("sigma_Z", sigma=0.5)



# Observational models (demonstration only)

W = pm.LogNormal("W", mu=eta, sigma=sigma_W, observed=W_obs)

Z = pm.LogNormal("Z", mu=0.05 + 0.85*eta, sigma=sigma_Z, observed=Z_obs)



# 3. Evidence Stream Effects (illustrative parameters)

beta_mech = pm.Normal("beta_mech", mu=0.0, sigma=1.0)

beta_animal = pm.Normal("beta_animal", mu=0.0, sigma=1.0)

beta_human = pm.Normal("beta_human", mu=0.0, sigma=1.0)

beta_epi = pm.Normal("beta_epi", mu=0.0, sigma=1.0)



# 4. Pathway Modulation (demonstration structure)
```

```
gamma_mech = pm.Normal("gamma_mech", mu=0.0, sigma=0.5, shape=3)

gamma_animal = pm.Normal("gamma_animal", mu=0.0, sigma=0.5, shape=3)

gamma_human = pm.Normal("gamma_human", mu=0.0, sigma=0.5, shape=3)



# 5. Hypothesis Prior (illustrative)

# Using conservative prior for DEMONSTRATION

p_H = pm.Beta("p_H", alpha=10, beta=90)  # Mean = 10%



# ==================== LIKELIHOOD MODELS ====================

# These show MODEL STRUCTURE, not real inferences



# Mechanistic evidence (illustrative)

mu_mech = beta_mech * dose_mech + pt.dot(gamma_mech, theta)

y_mech_obs = pm.Normal("y_mech_obs", mu=mu_mech, sigma=0.5, observed=y_mech)



# Animal evidence (illustrative)

log_mu_animal = beta_animal * dose_animal + pt.dot(gamma_animal, theta)

mu_animal = pt.exp(log_mu_animal)

phi = pm.Gamma("phi", alpha=2, beta=0.5)

y_animal_obs = pm.NegativeBinomial("y_animal_obs", mu=mu_animal,

                                   alpha=phi, observed=y_animal)



# Human evidence (illustrative)

mu_human = beta_human * eta + pt.dot(gamma_human, theta)
```

```
    y_human_obs = pm.Normal("y_human_obs", mu=mu_human, sigma=0.5, observed=y_human)



    # Epidemiological evidence (illustrative)

    logit_p_epi = beta_epi * exposure_epi

    y_epi_obs = pm.Bernoulli("y_epi_obs", logit_p=logit_p_epi, observed=y_epi)



print("Proof-of-concept model defined (for demonstration only)")
```

B.5 Simplified Sampling for Stability [2] [available at GitHub Repository. URL: https://github.com/BARC-commits/BARC_Bayesian-Assessment-of-Research-Causality]

Important: Complex hierarchical models often face computational challenges. We use **simplified settings** for this demonstration.

```
"""
SIMPLIFIED SAMPLING FOR DEMONSTRATION STABILITY
Real applications would require more extensive sampling
"""

print("\nRunning DEMONSTRATION sampling (simplified for stability)...")
print("NOTE: This shows WORKFLOW, not production inference")

with barc_demo_model:
    # Use simplified settings for DEMONSTRATION stability
    demo_trace = pm.sample(
        draws=1000,          # Reduced for demonstration speed
        tune=500,            # Reduced tuning
```

```
        chains=2,            # Minimal chains
        cores=1,             # Single core for simplicity
        target_accept=0.85,  # Conservative for stability
        random_seed=42,
        init='adapt_diag',
        return_inferencedata=True,
        progressbar=True
    )


print("Demonstration sampling complete")
print("REMINDER: All outputs are from SIMULATED DATA only")
```

B.6 Demonstration Results (ILLUSTRATIVE ONLY) [available at GitHub Repository. URL: https://github.com/BARC-commits/BARC_Bayesian-Assessment-of-Research-Causality]

[DEMONSTRATION OUTPUTS - NOT REAL INFERENCES]

```
print("\n" + "=" * 80)
print("DEMONSTRATION RESULTS (from SIMULATED DATA)")
print("=" * 80)

# B.6.1 Important Note on Convergence Diagnostics
print("\nB.6.1 Important Note on Convergence Diagnostics")
print("-" * 80)
print("The sampling outputs shown below include high divergence counts")
print(" and low effective sample sizes. These")
print("are INTENTIONALLY PRESENTED to illustrate several key points:\n")
print("1. Complex hierarchical models require careful specification —")
print("   the simplified model here intentionally includes weakly identified")
print("   parameters to show what can go wrong.\n")
```

```
print("2. Convergence diagnostics are essential — the divergence warnings")

print("   demonstrate why rigorous checking is needed in real applications.\n")

print("3. Proof-of-concept vs. production — this simplified demonstration")

print("   intentionally shows challenges that would need resolution before")

print("   production use.\n")

print("In a production implementation, these issues would be addressed through:")

print("- Model reparameterization")

print("- Stronger prior regularization")

print("- More extensive sampling")

print("- Comprehensive diagnostic validation\n")

print("The outputs are presented not as valid inferences, but as")

print("illustrations of the diagnostic process.")



# Diagnostics

if hasattr(demo_trace, 'sample_stats'):

    n_div = demo_trace.sample_stats.diverging.sum().item()

    print(f"\nSampling divergences: {n_div} (expected in this proof-of-concept demonstration)")

    if n_div > 0:

        print("Note: The high divergence counts reflect the intentionally simplified nature of this proof-of-concept implementation.")
```

```
# Output Presentation (changed from numerical estimates to narrative patterns)

print("\nILLUSTRATIVE Output Patterns (demonstration only):")

print("-" * 60)

print("- All β coefficients were positive (consistent with simulated effects)")

print("- Posterior P(H) showed updating from prior with substantial uncertainty")

print("- Convergence diagnostics indicated need for model refinement")

print("[Specific numerical values omitted to avoid misinterpretation]")



print("\n" + "=" * 80)

print("END OF DEMONSTRATION OUTPUTS")

print("=" * 80)
```

B.7 Outputs

B.7.1 Output Example from Google Colab:

Simulated data generated (illustrative only).

```
Simulated data generated (illustrative only).

BARC Appendix B: Proof-of-Concept Hierarchical Bayesian Implementation
================================================================
DISCLAIMER: This is a DEMONSTRATION using SIMULATED DATA
================================================================
Proof-of-concept model defined (for demonstration only).

Running DEMONSTRATION sampling (simplified for stability)...
NOTE: This shows WORKFLOW, not production inference

 Progress Draws   Divergences   Step size   Grad evals   Sampling Speed Elapsed   Remaining
 ———————————————————————————————————————————————————
 ——1500       66             0.072       31           42.97 draws/s  0:00:34           0:00:00
 ——1500            20             0.065       63           22.86 draws/s  0:01:05           0:00:00



ERROR:pymc.stats.convergence:There were 86 divergences after tuning. Increase `target_accept` or reparameterize.
ERROR:pymc.stats.convergence:The effective sample size per chain is smaller than 100 for some parameters.  A higher number is needed for reliable rhat and ess computation. See https://arxiv.org/abs/1903.08008 for details
Demonstration sampling complete
REMINDER: All outputs are from SIMULATED DATA only
```

```
================================================================
DEMONSTRATION RESULTS (from SIMULATED DATA)
================================================================

B.6.1 Important Note on Convergence Diagnostics
--------------------------------------------------------------------------------
The sampling outputs shown below include high divergence counts
and low effective sample sizes. These
are INTENTIONALLY PRESENTED to illustrate several key points:

1. Complex hierarchical models require careful specification —
   the simplified model here intentionally includes weakly identified
   parameters to show what can go wrong.

2. Convergence diagnostics are essential — the divergence warnings
   demonstrate why rigorous checking is needed in real applications.

3. Proof-of-concept vs. production — this simplified demonstration
   intentionally shows challenges that would need resolution before
   production use.

In a production implementation, these issues would be addressed through:
- Model reparameterization
- Stronger prior regularization
- More extensive sampling
```

- Comprehensive diagnostic validation

The outputs are presented not as valid inferences, but as

illustrations of the diagnostic process.

Sampling divergences: 86 (expected in this proof-of-concept demonstration)

Note: The high divergence counts reflect the intentionally simplified nature of this proof-of-concept implementation.

ILLUSTRATIVE Output Patterns (demonstration only):

- All β coefficients were positive (consistent with simulated effects)

- Posterior P(H) showed updating from prior with substantial uncertainty

- Convergence diagnostics indicated need for model refinement

[Specific numerical values omitted to avoid misinterpretation]

==========

END OF DEMONSTRATION OUTPUTS

==========

B.7.2 Output Example from Kaggle Notebook:

Simulated data generated (illustrative only).

```
Simulated data generated (illustrative only).


BARC Appendix B: Proof-of-Concept Hierarchical Bayesian Implementation
============================================================
DISCLAIMER: This is a DEMONSTRATION using SIMULATED DATA
============================================================
Proof-of-concept model defined (for demonstration only).


Running DEMONSTRATION sampling (simplified for stability)...
NOTE: This shows WORKFLOW, not production inference
Initializing NUTS using adapt_diag...
Sequential sampling (2 chains in 1 job)
NUTS: [sigma_theta, theta_0, theta_1, theta_2, eta, sigma_W, sigma_Z, beta_mech,
beta_animal, beta_human, beta_epi, gamma_mech, gamma_animal, gamma_human,
p_H, phi]


Progress Draws Divergences Step size Grad evals Sampling Speed Elapsed  Remaining
 ———— 1500    698         0.07       63         60.66 draws/s    0:00:24   0:00:00
 ———— 1500    21          0.05       63         25.99 draws/s    0:00:57   0:00:00
```

```
Sampling 2 chains for 500 tune and 1_000 draw iterations (1_000 + 2_000 draws total) took 58 seconds.

There were 719 divergences after tuning. Increase `target_accept` or reparameterize.

We recommend running at least 4 chains for robust computation of convergence diagnostics

The rhat statistic is larger than 1.01 for some parameters. This indicates problems during sampling. See https://arxiv.org/abs/1903.08008 for details

The effective sample size per chain is smaller than 100 for some parameters.  A higher number is needed for reliable rhat and ess computation. See https://arxiv.org/abs/1903.08008 for details

Demonstration sampling complete

REMINDER: All outputs are from SIMULATED DATA only


============================================================

DEMONSTRATION RESULTS (from SIMULATED DATA)

============================================================


B.6.1 Important Note on Convergence Diagnostics

----------------------------------------------------------------------------

The sampling outputs shown below include high divergence counts

 and low effective sample sizes. These

are INTENTIONALLY PRESENTED to illustrate several key points:


1. Complex hierarchical models require careful specification —

   the simplified model here intentionally includes weakly identified

   parameters to show what can go wrong.
```

2. Convergence diagnostics are essential — the divergence warnings
demonstrate why rigorous checking is needed in real applications.

3. Proof-of-concept vs. production — this simplified demonstration
intentionally shows challenges that would need resolution before
production use.

In a production implementation, these issues would be addressed through:
- Model reparameterization
- Stronger prior regularization
- More extensive sampling
- Comprehensive diagnostic validation

The outputs are presented not as valid inferences, but as
illustrations of the diagnostic process.

Sampling divergences: 719 (expected in this proof-of-concept demonstration)
Note: The high divergence counts reflect the intentionally simplified nature of this proof-of-concept implementation.

ILLUSTRATIVE Output Patterns (demonstration only):

- All β coefficients were positive (consistent with simulated effects)
- Posterior P(H) showed updating from prior with substantial uncertainty
- Convergence diagnostics indicated need for model refinement
[Specific numerical values omitted to avoid misinterpretation]

```
================================================================
END OF DEMONSTRATION OUTPUTS
================================================================
```

B.8 Sampling Results and Computational Notes

Computational Context: Both Colab and Kaggle environments exhibit **highly variable divergence counts** (ranging from ~86 to over 700), demonstrating the **instability** of this proof-of-concept implementation. This variability underscores that the model is pedagogical rather than production-ready.

Illustrative Patterns (demonstration only):

- *β coefficient directions varied between runs*
- *Posterior probability distributions showed high sensitivity to sampling variability*
- *Convergence diagnostics indicated **severe** model specification issues*
- *Parameter estimates exhibited **extreme uncertainty and instability***

Key Observations:

- Divergence counts: **Highly variable** (86-719), demonstrating model instability
- *Effective sample sizes: Very low for many parameters*
- R-hat statistics: Frequently >1.01, indicating poor mixing
- Between-run variability: Extreme differences in Chain 1 performance

*Interpretation: These diagnostics demonstrate that **even with identical code and random seeds, poorly specified models can produce wildly different results**. This reinforces that real applications would require **extensive model reparameterization,***

stronger priors, and comprehensive sensitivity analysis *rather than simply more sampling.*

B.9 What These Computational Patterns Demonstrate

The observed instability (e.g., divergence counts varying from 86 to 719 between runs with identical code) serves as a pedagogical illustration of several important principles:

1. Model Specification Matters:

- Weakly identified parameters create **extreme** sampling challenges (as seen in Chain 1's 698 divergences)
- Hierarchical structures require careful prior specification to avoid funnel geometries
- Convergence diagnostics are essential - single runs can be misleading

2. Computational Reproducibility:

- Different numerical backends/NUTS implementations can yield different diagnostics
- Production code requires stability across environments and random seeds
- Extensive testing and sensitivity analysis are needed for reliable deployment

3. From Proof-of-Concept to Production:

- Demonstrations highlight workflow, but production models must achieve diagnostic robustness
- The variability observed here would be unacceptable in applied research
- Real applications require reparameterization, stronger priors, and comprehensive validation

B.10 Steps Toward Production Implementation

To move from this demonstration to a production-ready framework:

1. **Model Refinement:**
    - Implement non-centered parameterizations
    - Add stronger prior regularization
    - Include identifiability constraints
2. **Computational Robustness:**
    - Extensive testing across computational environments
    - Adaptive sampling with higher target_accept rates
    - Parallel tempering for multimodal posteriors
3. **Validation Framework:**
    - Posterior predictive checks
    - Cross-validation
    - Sensitivity to prior specifications

Note: These steps are **beyond the scope** of this proof-of-concept demonstration, which intentionally uses a simplified model to illustrate conceptual workflow rather than production analysis.

B.11 What Real Implementation Would Require

CRITICAL DISTINCTION: This proof-of-concept demonstrates **structure**, not production-ready analysis.

Gaps Between Proof-of-Concept and Real Application:

Aspect	This Demonstration	Real Application
Data	Simulated	Empirical, validated
Priors	Simple defaults	Expert-elicited, evidence-based
Model Complexity	Simplified	Full hierarchical structure
Convergence	Basic diagnostics	Rigorous validation (R-hat < 1.01, ESS > 1000)
Validation	None	Cross-validation, predictive checks
Computer Resources	Minimal	Extensive (HPC clusters)

Specific Requirements for Real MNPs-Cancer Assessment:

1. **Data Infrastructure:**
    - Curated evidence databases
    - Standardized exposure metrics
    - Quality-controlled outcome data
2. **Model Development:**
    - Polymer-specific sub-models
    - Tissue-specific pathways
    - Time-dependent exposure models
3. **Validation Framework:**
    - Retrospective application to known carcinogens
    - Sensitivity to prior specifications
    - Robustness to missing data
4. **Computational Implementation:**
    - Optimized sampling algorithms
    - Parallel processing
    - Version-controlled codebase

B.12 Limitations of This Proof-of-Concept

This demonstration has INTENTIONAL limitations:

1. **Computational Simplifications:**
    - Reduced sampling iterations
    - Simplified model structure
    - Basic convergence diagnostics
2. **Conceptual Simplifications:**
    - Hypothetical pathways only
    - Simplified measurement error
    - Limited confounding adjustment
3. **Practical Limitations:**
    - No real data integration
    - No expert prior elicitation
    - No validation against known carcinogens

These limitations are BY DESIGN — this appendix shows **how the framework could be implemented**, not a complete implementation.

B.13 From Proof-of-Concept to Production

Implementation Roadmap (Illustrative):

Phase 1: Framework Development (This Appendix)

- ├── Conceptual model specification
- ├── Proof-of-concept implementation
- └── Demonstration of workflow

Phase 2: Pilot Validation (Future Work)

- ├── Application to test cases (known carcinogens)
- ├── Prior elicitation protocols
- └── Computational optimization

Phase 3: MNPs-Specific Application (Future Work)

- ├── Evidence synthesis
- ├── Expert elicitation for MNPs
- └── Model validation

B.14 Conclusion: Demonstration vs. Application

This appendix achieves ONE goal: Demonstrating **how** a hierarchical Bayesian framework for MNPs-cancer assessment **could be implemented**.

It does NOT achieve: Real risk assessment, policy recommendations, or empirical conclusions about MNPs.

The outputs are: Illustrative parameter estimates from simulated data showing **workflow**, not **findings**.

Real application would require: Everything listed in Section B.7, plus interdisciplinary collaboration, funding, and institutional support.

FINAL DISCLAIMER: All code, data, and outputs in this appendix are **FOR DEMONSTRATION PURPOSES ONLY**. They show **methodological structure**, not **scientific conclusions**. No inferences about MNPs or cancer should be drawn from this proof-of-concept implementation.

References:

1. Abril-Pla O, Andreani V, Carroll C, et al. PyMC: a modern probabilistic programming language in Python. PeerJ Comput Sci. 2023;9:e1516.

2. Betancourt M. A conceptual introduction to Hamiltonian Monte Carlo. arXiv [Preprint]. 2017. Available from: https://arxiv.org/abs/1701.02434