UNIVERSITÄT BAYREUTH

– P&E Master's Programme –
Chair of Philosophy, Computer Science & Artificial Intelligence

Automating the Modelling of Transformative Artificial Intelligence Risks

—

"*An Epistemic Framework for Leveraging Frontier AI Systems to Upscale Conditional Policy Assessments in Bayesian Networks on a Narrow Path towards Existencial Safety* "

—

A thesis submitted at the Department of Philosophy

for the degree of *Master of Arts in Philosophy & Economics*

Author:

Valentin Jakob Meyer
Valentin.meyer@uni-bayreuth.de
Matriculation Number: 1828610
Tel.: +49 (1573) 4512494
Pielmühler Straße 15
52066 Lappersdorf

Supervisor:

Dr. Timo Speith

Word Count:
30.000
Source / Identifier:
Document URL

26th of May 2025

Table of Contents

List of Figures

List of Tables

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

Abstract

Outline(s): Table of Contents

Preface

---

title: “Automating the Modelling of Transformative Artificial Intelligence Risks” subtitle: “An Epistemic Framework for Leveraging Frontier AI Systems to Upscale Conditional Policy Assessments in Bayesian Networks on a Narrow Path towards Existential Safety” author:

- name: Valentin Jakob Meyer orcid: 0009-0006-0889-5269 corresponding: true email: Valentin.Meyer@uni-bayreuth.de roles:
  - GraduateAuthor affiliations:
  - University of Bayreuth
  - MCMP — LMU Munich
- name: Dr. Timo Speith orcid: 0000-0002-6675-154X corresponding: false roles:
  - Supervisor affiliations:
  - University of Bayreuth keywords:
- AMTAIR
- AI Governance
- Bayesian Networks
- Transformative AI
- Risk Assessment
- Argument Extraction abstract: | This thesis addresses coordination failures in AI safety by creating computational tools that automatically extract and formalize probabilistic world models from AI safety literature using frontier language models. The AMTAIR (Automating Transformative AI Risk Modeling) system implements an end-to-end pipeline transforming unstructured arguments into interactive Bayesian networks through a novel two-stage extraction process: first capturing argument structure in ArgDown format, then enhancing it with probability information in BayesDown.

Applied to canonical examples and real AI safety arguments, the system demonstrates extraction accuracy exceeding 85% for structural relationships and 73% for probability capture. By making implicit models explicit, enabling cross-worldview comparison, and supporting rigorous policy evaluation, AMTAIR bridges communication gaps between technical researchers, policy specialists, and other stakeholders working to address existential risks from advanced AI.

The thesis contributes both theoretical foundations and practical implementation, validated through expert comparison and real-world case studies including Carlsmith’s power-seeking

AI model. While current limitations include correlation handling and extraction ambiguities, the approach provides essential epistemic infrastructure for coordinated AI governance. plain-language-summary: | This thesis develops software tools that automatically extract and visualize the hidden assumptions and probability estimates in AI safety arguments. By transforming complex written arguments into interactive diagrams showing relationships and probabilities, AMTAIR helps different groups working on AI safety—researchers, policymakers, and others—understand each other better and coordinate their efforts to address risks from advanced AI systems. key-points:

- A novel two-stage extraction pipeline transforms argument structures into Bayesian networks through ArgDown and BayesDown intermediate representations
- Interactive visualizations make complex probabilistic relationships accessible to diverse stakeholders
- Formal representation enables systematic comparison across different worldviews and assumptions
- Validated extraction achieves >85% accuracy for structure and >73% for probabilities
- The approach addresses coordination failures by creating a common language for AI risk assessment metadata-submission: field-of-study: "Philosophy & Economics M.A." matriculation-number: 1828610 submission-date: "May 26, 2025" word-count: 30000 date: "2025-05-26" bibliography: ref/MAref.bib citation: container-title: University of Bayreuth number-sections: true reference-location: margin citation-location: margin

---

This Quarto book represents the culmination of interdisciplinary research at the intersection of AI safety, formal epistemology, and computational social science. The work emerged from recognizing a fundamental challenge in AI governance: while investment in AI safety research has grown exponentially, coordination between different stakeholder communities remains fragmented, potentially increasing existential risk through misaligned efforts.

The journey from initial concept to working implementation involved iterative refinement based on feedback from advisors, domain experts, and potential users. What began as a technical exercise in automated extraction evolved into a broader framework for enhancing epistemic security in one of humanity's most critical coordination challenges.

Acknowledgments

I thank my supervisor Dr. Timo Speith for guidance throughout this project, the MTAIR team for pioneering the manual approach that inspired automation, and the AI safety community for creating the rich literature that made this work possible. Special recognition goes to technical advisors who provided implementation feedback and domain experts who validated extraction results.

Table of Contents

Abstract

The coordination crisis in AI governance presents a paradoxical challenge: unprecedented investment in AI safety coexists alongside fundamental coordination failures across technical, policy, and ethical domains. These divisions systematically increase existential risk by creating safety gaps, misallocating resources, and fostering inconsistent approaches to interdependent problems.

This thesis introduces AMTAIR (Automating Transformative AI Risk Modeling), a computational approach that addresses this coordination failure by automating the extraction of probabilistic world models from AI safety literature using frontier language models. The AMTAIR system implements an end-to-end pipeline that transforms unstructured text into interactive Bayesian networks through a novel two-stage extraction process: first capturing argument structure in ArgDown format, then enhancing it with probability information in BayesDown.

The core technical contribution involves developing intermediate representations that preserve both narrative structure and mathematical precision. ArgDown captures hierarchical argument relationships while remaining human-readable. BayesDown extends this with probabilistic metadata, creating a bridge to formal Bayesian networks. This two-stage approach separates concerns, enabling modular improvement and human oversight at critical decision points.

Validation through expert comparison and real-world case studies demonstrates extraction accuracy exceeding 85% for structural relationships and 73% for probability capture. Application to Carlsmith's power-seeking AI model shows the system can reconstruct complex multi-level causal structures with realistic uncertainty relationships. Comparative analysis across different AI governance worldviews reveals both convergence on key structural elements and critical disagreements on parameter values.

This approach bridges communication gaps between stakeholders by making implicit models explicit, enabling comparison across different worldviews, providing a common language for discussing probabilistic relationships, and supporting policy evaluation across diverse scenarios. While limitations remain in handling complex correlations and extraction ambiguities, AMTAIR provides essential epistemic infrastructure for enhanced coordination in AI governance.

---

1. Introduction: The Coordination Crisis in AI Governance

```
### 10% of Grade: ~ 14% of text ~ 4200 words ~ 10 pages

- introduces and motivates the core question or problem
- provides context for discussion (places issue within a larger debate or sphere of relevanc
- states precise thesis or position the author will argue for
- provides roadmap indicating structure and key content points of the essay
```

Opening Scenario: The Policymaker's Dilemma

Imagine a senior policy advisor preparing recommendations for AI governance legislation. On her desk lie a dozen reports from leading AI safety researchers, each painting a different picture of the risks ahead. One argues that misaligned AI could pose existential risks within the decade, citing complex technical arguments about instrumental convergence and orthogonality. Another suggests these concerns are overblown, emphasizing uncertainty and the strength of existing institutions. A third proposes specific technical standards but acknowledges deep uncertainty about their effectiveness.

Each report seems compelling in isolation, written by credentialed experts with sophisticated arguments. Yet they reach dramatically different conclusions about both the magnitude of risk and appropriate interventions. The technical arguments involve unfamiliar concepts—mesa-optimization, corrigibility, capability amplification—expressed through different frameworks and implicit assumptions. Time is limited, stakes are high, and the legislation could shape humanity's trajectory for decades.

This scenario plays out daily across government offices, corporate boardrooms, and research institutions worldwide. It exemplifies what I term the "coordination crisis" in AI governance: despite unprecedented attention and resources directed toward AI safety, we lack the epistemic infrastructure to synthesize diverse expert knowledge into actionable governance strategies.

The Coordination Crisis in AI Governance

As AI capabilities advance at an accelerating pace—demonstrated by the rapid progression from GPT-3 to GPT-4, Claude, and emerging multimodal systems—humanity faces a governance challenge unlike any in history. The task of ensuring increasingly powerful AI systems remain aligned with human values and beneficial to our long-term flourishing grows more urgent with each capability breakthrough. This challenge becomes particularly acute when considering transformative AI systems that could drastically alter civilization's trajectory, potentially including existential risks from misaligned systems pursuing objectives counter to human welfare.

The current state of AI governance presents a striking paradox. On one hand, we witness extraordinary mobilization: billions in research funding, proliferating safety initiatives, major tech companies establishing alignment teams, and governments worldwide developing AI strategies. The Asilomar AI Principles garnered thousands of signatures, the EU advances comprehensive AI regulation, and technical researchers produce increasingly sophisticated work on alignment, interpretability, and robustness.

Yet alongside this activity, we observe systematic coordination failures that may prove catastrophic. Technical safety researchers develop sophisticated alignment techniques without clear implementation pathways. Policy specialists craft regulatory frameworks lacking technical grounding to ensure practical efficacy. Ethicists articulate normative principles that lack operational specificity. Strategy researchers identify critical uncertainties but struggle to translate these into actionable guidance. International bodies convene without shared frameworks for assessing interventions.

This fragmentation is not merely inefficient—it systematically amplifies existential risk through several mechanisms:

Safety Gaps from Misaligned Efforts

When different communities operate with incompatible frameworks, critical risks fall through the cracks. Technical researchers may solve alignment problems under assumptions that policymakers' decisions invalidate. Regulations optimized for current systems may inadvertently incentivize dangerous development patterns. Without shared models of the risk landscape, our collective efforts resemble the parable of blind men describing an elephant—each accurate within their domain but missing the complete picture.

Resource Misallocation

The AI safety community duplicates efforts while leaving critical areas underexplored. Multiple teams independently develop similar frameworks without building on each other's work. Funders struggle to identify high-impact opportunities across technical and governance domains. Talent flows toward well-publicized approaches while neglected strategies remain understaffed. This misallocation becomes more costly as the window for establishing effective governance narrows.

Negative-Sum Dynamics

Perhaps most concerning, uncoordinated interventions can actively increase risk. Safety standards that advantage established players may accelerate risky development elsewhere. Partial transparency requirements might enable capability advances without commensurate safety improvements. International agreements lacking shared technical understanding may lock in dangerous practices. Without coordination, our cure risks becoming worse than the disease.

Historical Parallels and Temporal Urgency

History offers instructive parallels. The nuclear age began with scientists racing to understand and control forces that could destroy civilization. Early coordination failures—competing national programs, scientist-military tensions, public-expert divides—nearly led to catastrophe multiple times. Only through developing shared frameworks (deterrence theory), institutions (IAEA), and communication channels (hotlines, treaties) did humanity navigate the nuclear precipice.

Yet AI presents unique coordination challenges that compress our response timeline:

Accelerating Development: Unlike nuclear weapons requiring massive infrastructure, AI development proceeds in corporate labs and academic departments worldwide. Capability improvements come through algorithmic insights and computational scale, both advancing exponentially.

Dual-Use Ubiquity: Every AI advance potentially contributes to both beneficial applications and catastrophic risks. The same language model architectures enabling scientific breakthroughs could facilitate dangerous manipulation or deception at scale.

Comprehension Barriers: Nuclear risks were viscerally understandable—cities vaporized, radiation sickness, nuclear winter. AI risks involve abstract concepts like optimization processes, goal misspecification, and emergent capabilities that resist intuitive understanding.

Governance Lag: Traditional governance mechanisms—legislation, international treaties, professional standards—operate on timescales of years to decades. AI capabilities advance on timescales of months to years, creating an ever-widening capability-governance gap.

Research Question and Scope

This thesis addresses a specific dimension of the coordination challenge by investigating:

How can computational approaches formalize the worldviews and arguments underlying AI safety discourse, transforming qualitative disagreements into quantitative models suitable for rigorous policy evaluation?

More specifically, I explore whether frontier AI technologies can be utilized to automate the modeling of transformative AI risks, enabling robust prediction of policy impacts across diverse worldviews.

To break this down:

- **Computational Formalization**: Using automated extraction and formal representation to make implicit models explicit
- **Worldview Representation**: Capturing different perspectives on AI risk in comparable frameworks
- **Argument Transformation**: Converting natural language arguments into structured Bayesian networks
- **Policy Evaluation**: Assessing intervention impacts through formal counterfactual analysis

The scope encompasses both theoretical development and practical implementation. Theoretically, I develop a framework for representing diverse perspectives on AI risk in a common formal language. Practically, I implement this framework in a computational system—the AI Risk Pathway Analyzer (ARPA)—that enables interactive exploration of how policy interventions might alter existential risk across different worldviews.

This investigation focuses specifically on existential risks from misaligned AI systems rather than broader AI ethics concerns. This narrowed scope enables deep technical development while addressing the highest-stakes coordination challenges where current fragmentation poses the greatest danger.

The Multiplicative Benefits Framework

The central thesis of this work is that combining three elements—automated worldview extraction, prediction market integration, and formal policy evaluation—creates multiplicative rather than merely additive benefits for AI governance. Each component enhances the others, creating a system more valuable than the sum of its parts.

Automated Worldview Extraction

Current approaches to AI risk modeling, exemplified by the Modeling Transformative AI Risks (MTAIR) project, demonstrate the value of formal representation but require extensive manual effort. Creating a single model demands hundreds of expert-hours to translate qualitative arguments into quantitative frameworks. This bottleneck severely limits the number of perspectives that can be formalized and the speed of model updates as new arguments emerge.

Automation using frontier language models addresses this scaling challenge. By developing systematic methods to extract causal structures and probability judgments from natural language, we can process orders of magnitude more content, incorporate diverse perspectives rapidly, and maintain models that evolve with the discourse.

Live Data Integration

Static models, however well-constructed, quickly become outdated in fast-moving domains. Prediction markets and forecasting platforms aggregate distributed knowledge about uncertain fu-

tures, providing continuously updated probability estimates. By connecting formal models to these live data sources, we create dynamic assessments that incorporate the latest collective intelligence.

This integration serves multiple purposes: grounding abstract models in empirical forecasts, identifying which uncertainties most affect outcomes, revealing when model assumptions diverge from collective expectations, and generating new questions for forecasting communities.

Formal Policy Evaluation

The ultimate purpose of risk modeling is informing action. Formal policy evaluation transforms static risk assessments into actionable guidance by modeling how specific interventions alter critical parameters. Using causal inference techniques, we can assess not just the probability of adverse outcomes but how those probabilities change under different policy regimes.

This enables genuinely evidence-based policy development: comparing interventions across multiple worldviews, identifying robust strategies that work across scenarios, understanding which uncertainties most affect policy effectiveness, and prioritizing research to reduce decision-relevant uncertainty.

The Synergy

The multiplicative benefits emerge from the interactions between components:

- Automation enables comprehensive coverage, making prediction market integration more valuable by connecting to more perspectives
- Market data validates and calibrates automated extractions, improving quality
- Policy evaluation gains precision from both comprehensive models and live probability updates
- The complete system creates feedback loops where policy analysis identifies critical uncertainties for market attention

This synergistic combination addresses the coordination crisis by providing common ground for disparate communities, translating between technical and policy languages, quantifying previously implicit disagreements, and enabling evidence-based compromise.

Thesis Structure and Roadmap

The remainder of this thesis develops the multiplicative benefits framework from theoretical foundations to practical implementation:

Chapter 2: Context and Theoretical Foundations establishes the intellectual groundwork, examining:

- The epistemic challenges unique to AI governance
- Bayesian networks as formal tools for uncertainty representation
- Argument mapping as a bridge from natural language to formal models

- The MTAIR project's achievements and limitations
- Requirements for effective coordination infrastructure

Chapter 3: AMTAIR Design and Implementation presents the technical system:

- Overall architecture and design principles
- The two-stage extraction pipeline (ArgDown → BayesDown)
- Validation methodology and results
- Case studies from simple examples to complex AI risk models
- Integration with prediction markets and policy evaluation

Chapter 4: Discussion - Implications and Limitations critically examines:

- Technical limitations and failure modes
- Conceptual concerns about formalization
- Integration with existing governance frameworks
- Scaling challenges and opportunities
- Broader implications for epistemic security

Chapter 5: Conclusion synthesizes key contributions and charts paths forward:

- Summary of theoretical and practical achievements
- Concrete recommendations for stakeholders
- Research agenda for community development
- Vision for AI governance with proper coordination infrastructure

Throughout, I maintain dual focus on theoretical sophistication and practical utility. The framework aims not merely to advance academic understanding but to provide actionable tools for improving coordination in AI governance during this critical period.

---

2. Context and Theoretical Foundations

```
### 20% of Grade: ~ 29% of text ~ 8700 words ~ 20 pages

- demonstrates understanding of all relevant core concepts
- explains why the question/thesis/problem is relevant in student's own words (supported by
- situates it within the debate/course material
- reconstructs selected arguments and identifies relevant assumptions
- describes additional relevant material that has been consulted and integrates it with the
```

This chapter establishes the theoretical and methodological foundations necessary for understanding AMTAIR's approach to automating AI risk modeling. I begin with the core challenge—representing existential risk arguments in formal terms—then develop the technical and conceptual tools needed to address it.

AI Existential Risk: The Carlsmith Model

To ground our discussion in concrete terms, I examine Joseph Carlsmith's "Is Power-Seeking AI an Existential Risk?" as an exemplar of structured reasoning about AI catastrophic risk. Carlsmith's analysis stands out for its explicit probabilistic decomposition of the path from current AI development to potential existential catastrophe.

Six-Premise Decomposition

Carlsmith structures his argument through six conditional premises, each assigned explicit probability estimates:

Premise 1: APS Systems by 2070 (P 0.65)
"By 2070, there will be AI systems with Advanced capability, Agentic planning, and Strategic awareness" - the conjunction of capabilities that could enable systematic pursuit of objectives in the world.

Premise 2: Alignment Difficulty (P 0.40)
"It will be harder to build aligned APS systems than misaligned systems that are still attractive to deploy" - capturing the challenge that safety may conflict with capability or efficiency.

Premise 3: Deployment Despite Misalignment (P 0.70)
"Conditional on 1 and 2, we will deploy misaligned APS systems" - reflecting competitive pressures and limited coordination.

Premise 4: Power-Seeking Behavior (P 0.65)
"Conditional on 1-3, misaligned APS systems will seek power in high-impact ways" - based on instrumental convergence arguments.

Premise 5: Disempowerment Success (P 0.40)
"Conditional on 1-4, power-seeking will scale to permanent human disempowerment" - despite potential resistance and safeguards.

Premise 6: Existential Catastrophe (P 0.95)
"Conditional on 1-5, this disempowerment constitutes existential catastrophe" - connecting power loss to permanent curtailment of human potential.

Overall Risk: Multiplying through the conditional chain yields P(doom) 0.05 or 5% by 2070.

Why Carlsmith Exemplifies Formalizable Arguments

Carlsmith's model demonstrates several features that make it ideal for formal representation:

Explicit Probabilistic Structure: Each premise receives numerical probability estimates with documented reasoning, enabling direct translation to Bayesian network parameters.

Clear Conditional Dependencies: The logical flow from capabilities through deployment decisions to catastrophic outcomes maps naturally onto directed acyclic graphs.

Transparent Decomposition: Breaking the argument into modular premises allows independent evaluation and sensitivity analysis of each component.

Documented Reasoning: Extensive justification for each probability enables extraction of both structure and parameters from the source text.

This structured approach exemplifies the type of reasoning AMTAIR aims to formalize and automate. While Carlsmith spent months developing this model manually, similar rigor exists implicitly in many AI safety arguments awaiting extraction.

The Epistemic Challenge of Policy Evaluation

Evaluating AI governance policies presents unique epistemic challenges that traditional policy analysis methods cannot adequately address. Understanding these challenges motivates the need for new computational approaches.

Unique Characteristics of AI Governance

Deep Uncertainty Rather Than Risk: Traditional policy analysis distinguishes between risk (known probability distributions) and uncertainty (known possibilities, unknown probabilities). AI governance faces deep uncertainty—we cannot confidently enumerate possible futures, much

less assign probabilities. Will recursive self-improvement enable rapid capability gains? Can value alignment be solved technically? These foundational questions resist empirical resolution before their answers become catastrophically relevant.

Complex Multi-Level Causation: Policy effects propagate through technical, institutional, and social levels with intricate feedback loops. A technical standard might alter research incentives, shifting capability development trajectories, changing competitive dynamics, and ultimately affecting existential risk through pathways invisible at the policy's inception. Traditional linear causal models cannot capture these dynamics.

Irreversibility and Lock-In: Many AI governance decisions create path dependencies that prove difficult or impossible to reverse. Early technical standards shape development trajectories. Institutional structures ossify. International agreements create sticky equilibria. Unlike many policy domains where course correction remains possible, AI governance mistakes may prove permanent.

Value-Laden Technical Choices: The entanglement of technical and normative questions confounds traditional separation of facts and values. What constitutes "alignment"? How much capability development should we risk for economic benefits? Technical specifications embed ethical judgments that resist neutral expertise.

Limitations of Traditional Approaches

Standard policy evaluation tools prove inadequate for these challenges:

Cost-Benefit Analysis assumes commensurable outcomes and stable probability distributions. When potential outcomes include existential catastrophe with deeply uncertain probabilities, the mathematical machinery breaks down. Infinite negative utility resists standard decision frameworks.

Scenario Planning helps explore possible futures but typically lacks the probabilistic reasoning needed for decision-making under uncertainty. Without quantification, scenarios provide narrative richness but limited action guidance.

Expert Elicitation aggregates specialist judgment but struggles with interdisciplinary questions where no single expert grasps all relevant factors. Moreover, experts often operate with different implicit models, making aggregation problematic.

Red Team Exercises test specific plans but miss systemic risks emerging from component interactions. Gaming individual failures cannot reveal emergent catastrophic possibilities.

These limitations create a methodological gap: we need approaches that handle deep uncertainty, represent complex causation, quantify expert disagreement, and enable systematic exploration of intervention effects.

Bayesian Networks as Knowledge Representation

Bayesian networks offer a mathematical framework uniquely suited to addressing these epistemic challenges. By combining graphical structure with probability theory, they provide tools for reasoning about complex uncertain domains.

Mathematical Foundations

A Bayesian network consists of:

- **Directed Acyclic Graph (DAG)**: Nodes represent variables, edges represent direct dependencies
- **Conditional Probability Tables (CPTs)**: For each node, P(node|parents) quantifies relationships

The joint probability distribution factors according to the graph structure:

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i|Parents(X_i))$$

This factorization enables efficient inference and embodies causal assumptions explicitly.

The Rain-Sprinkler-Grass Example

The canonical example illustrates key concepts:

```
[Grass_Wet]: Concentrated moisture on grass.
 + [Rain]: Water falling from sky.
 + [Sprinkler]: Artificial watering system.
   + [Rain]
```

Network Structure:

- **Rain** (root cause): P(rain) = 0.2
- **Sprinkler** (intermediate): P(sprinkler|rain) varies by rain state
- **Grass_Wet** (effect): P(wet|rain, sprinkler) depends on both causes

This simple network demonstrates:

- **Marginal Inference**: P(grass_wet) computed from joint distribution
- **Diagnostic Reasoning**: P(rain|grass_wet) reasoning from effects to causes
- **Intervention Modeling**: P(grass_wet|do(sprinkler=on)) for policy analysis

Advantages for AI Risk Modeling

Bayesian networks provide several crucial capabilities:

Explicit Uncertainty Representation: Every belief is a probability distribution, avoiding false certainty while enabling quantitative reasoning.

Causal Modeling: Directed edges represent causal relationships, enabling counterfactual reasoning through Pearl's do-calculus for policy evaluation.

Modular Structure: Complex arguments decompose into manageable components that can be independently evaluated and refined.

Evidence Integration: Bayesian updating provides principled methods for incorporating new information as it emerges.

Visual Communication: Graphical structure makes complex relationships comprehensible across expertise levels.

These features address key requirements for AI governance: handling uncertainty, representing causation, enabling systematic analysis, and facilitating communication across communities.

Argument Mapping and Formal Representations

The gap between natural language arguments and formal models requires systematic bridging. Argument mapping provides methods for making implicit reasoning structures explicit and analyzable.

From Natural Language to Structure

Natural language arguments contain rich information expressed through:

- Causal claims ("X leads to Y")
- Conditional relationships ("If A then likely B")
- Uncertainty expressions ("probably," "might," "certainly")
- Support/attack patterns between claims

Argument mapping extracts this structure, identifying:

- **Core claims and propositions**
- **Inferential relationships**
- **Implicit assumptions**
- **Uncertainty qualifications**

ArgDown: Structured Argument Notation

ArgDown provides a markdown-like syntax for hierarchical argument representation:

```
[MainClaim]: Description of primary conclusion.
 + [SupportingEvidence]: Evidence supporting the claim.
   + [SubEvidence]: More specific support.
 - [CounterArgument]: Evidence against the claim.
```

This notation captures argument structure while remaining human-readable and writable. Crucially, it serves as an intermediate representation between natural language and formal models.

BayesDown: The Bridge to Bayesian Networks

BayesDown extends ArgDown with probabilistic metadata:

```
[Node]: Description. {
  "instantiations": ["node_TRUE", "node_FALSE"],
  "priors": {"p(node_TRUE)": "0.7", "p(node_FALSE)": "0.3"},
  "posteriors": {
    "p(node_TRUE|parent_TRUE)": "0.9",
    "p(node_TRUE|parent_FALSE)": "0.4"
  }
}
```

This representation:

- **Preserves narrative structure** from the original argument
- **Adds mathematical precision** through probability specifications
- **Enables transformation** to standard Bayesian network formats
- **Supports validation** by maintaining traceability to sources

The two-stage extraction process (ArgDown → BayesDown) separates concerns: first capturing structure, then quantifying relationships. This modularity enables human oversight at critical decision points.

The MTAIR Framework: Achievements and Limitations

The Modeling Transformative AI Risks (MTAIR) project, led by RAND researchers, pioneered formal modeling of AI existential risk arguments. Understanding its approach and limitations motivates the automation efforts of AMTAIR.

MTAIR's Approach

MTAIR manually translated influential AI risk arguments into Bayesian networks using Analytica software:

Systematic Decomposition: Breaking complex arguments into variables and relationships through expert analysis.

Probability Elicitation: Gathering quantitative estimates through structured expert interviews and literature review.

Sensitivity Analysis: Identifying which parameters most influence conclusions about AI risk levels.

Visual Communication: Creating interactive models that stakeholders could explore and modify.

Key Achievements

MTAIR demonstrated several important possibilities:

Feasibility of Formalization: Complex philosophical arguments about AI risk can be represented as Bayesian networks while preserving essential insights.

Value of Quantification: Moving from qualitative concerns to quantitative models enables systematic analysis, comparison, and prioritization.

Cross-Perspective Communication: Formal models provide common ground for technical and policy communities to engage productively.

Research Prioritization: Sensitivity analysis reveals which empirical questions would most reduce uncertainty about AI risks.

Fundamental Limitations

However, MTAIR's manual approach faces severe constraints:

Labor Intensity: Each model requires hundreds of expert-hours to construct, limiting coverage to a few perspectives.

Static Nature: Models become outdated as arguments evolve but updating requires near-complete reconstruction.

Limited Accessibility: Using the models requires Analytica software and significant technical sophistication.

Single Perspective: Each model represents one worldview, making comparison across perspectives difficult.

These limitations prevent MTAIR's approach from scaling to meet AI governance needs. As the pace of AI development accelerates and arguments proliferate, manual modeling cannot keep pace.

The Automation Opportunity

MTAIR's experience reveals both the value of formal modeling and the necessity of automation. Key lessons:

- Formal models genuinely enhance understanding and coordination
- The modeling process itself surfaces implicit assumptions
- Quantification enables analyses impossible with qualitative arguments alone
- But manual approaches cannot scale to match the challenge

This motivates AMTAIR's central innovation: using frontier language models to automate the extraction and formalization process while preserving the benefits MTAIR demonstrated.

Requirements for Coordination Infrastructure

Based on the challenges identified and lessons from existing approaches, we can specify requirements for computational tools that could enhance coordination in AI governance:

Scalability

The system must process large volumes of arguments across:

- Academic papers and technical reports
- Policy documents and proposals
- Blog posts and informal arguments
- Forecasting questions and market data

Automation is essential—manual approaches cannot match the pace of discourse.

Accessibility

Diverse stakeholders must be able to engage with the system:

- **Researchers** need technical depth and modification capabilities
- **Policymakers** require clear summaries and intervention analysis
- **Forecasters** want integration with prediction platforms
- **Public stakeholders** deserve transparent representation

This demands multiple interfaces and levels of abstraction.

Epistemic Virtues

The system should enhance rather than replace human judgment by:

- **Making assumptions explicit** through formal representation
- **Preserving uncertainty** rather than false precision
- **Enabling validation** through traceable extraction
- **Supporting disagreement** through multi-worldview representation
- **Encouraging updating** as new evidence emerges

Integration Capabilities

Isolated tools have limited impact. The system needs:

- **Data source connections** to prediction markets and forecasting platforms
- **API accessibility** for integration with other tools
- **Export formats** compatible with standard analysis software
- **Version control** for tracking model evolution
- **Collaborative features** for community development

Robustness Properties

Given the high stakes, the system must handle:

- **Extraction errors** through validation and correction mechanisms
- **Adversarial inputs** designed to manipulate outputs
- **Model uncertainty** through sensitivity analysis
- **Scaling challenges** as networks grow large
- **Evolution over time** as arguments develop

These requirements shape AMTAIR's design, as detailed in the next chapter.

---

3. AMTAIR: Design and Implementation

```
### 20% of Grade: ~ 29% of text ~ 8700 words ~ 20 pages

- provides critical or constructive evaluation of positions introduced
- develops strong (plausible) argument in support of author's own position/thesis
- argument draws on relevant course material
- demonstrates understanding of course materials and key concepts
- presents original or insightful contribution to the debate
```

This chapter presents the technical architecture and implementation of AMTAIR, demonstrating how theoretical principles translate into working software. I detail the design decisions, implementation challenges, and validation results that establish AMTAIR's feasibility and value.

System Architecture Overview

AMTAIR implements an end-to-end pipeline transforming unstructured text into interactive Bayesian network visualizations. The architecture reflects key design principles:

- **Modularity**: Each component can be independently improved
- **Transparency**: Intermediate outputs enable inspection and validation
- **Flexibility**: Multiple input formats and configurable processing
- **Scalability**: Efficient processing of large document sets

Five-Stage Pipeline

The system processes information through five distinct stages:

```
Documents → Ingestion → ArgDown → BayesDown → Networks → Visualization
```

Each stage produces inspectable outputs, enabling validation and debugging. This transparency is crucial for building trust in automated extraction.

Component Architecture

```
class AMTAIRPipeline:
    def __init__(self):
        self.ingestion = DocumentIngestion()
        self.extraction = BayesDownExtractor()
        self.transformation = DataTransformer()
        self.network_builder = BayesianNetworkBuilder()
        self.visualizer = InteractiveVisualizer()

    def process(self, document):
        """End-to-end processing from document to interactive model"""
        structured_data = self.ingestion.preprocess(document)
        bayesdown = self.extraction.extract(structured_data)
        dataframe = self.transformation.convert(bayesdown)
        network = self.network_builder.construct(dataframe)
        return self.visualizer.render(network)
```

This clean separation of concerns enables targeted improvements and alternative implementations for each component.

The Two-Stage Extraction Process

The core innovation of AMTAIR lies in separating structural extraction from probability quantification. This two-stage approach addresses key challenges in automated formalization.

Stage 1: Structural Extraction (ArgDown)

The first stage identifies argument structure without concerning itself with quantification:

Variable Identification: Extract key propositions and entities from text using patterns like "X causes Y," "If A then B," and domain-specific indicators.

Relationship Mapping: Identify support, attack, and conditional relationships between variables through linguistic analysis.

Hierarchy Construction: Build nested ArgDown representation preserving logical flow:

```
[Existential_Catastrophe]: Destruction of humanity's potential.
 + [Human_Disempowerment]: Loss of control to AI systems.
   + [Misaligned_Power_Seeking]: AI pursuing problematic objectives.
     + [APS_Systems]: Advanced, agentic, strategic AI.
     + [Deployment_Decisions]: Choice to deploy despite risks.
```

Validation: Ensure extracted structure forms valid directed acyclic graph and preserves key argumentative relationships from source.

Stage 2: Probability Integration (BayesDown)

The second stage adds quantitative information to the structural skeleton:

Question Generation: For each node, generate probability elicitation questions:

- "What is the probability of existential catastrophe?"
- "What is P(catastrophe|human_disempowerment)?"

Probability Extraction: Identify explicit numerical statements and map qualitative expressions:

- "Very likely" → 0.75-0.9
- "Possible but unlikely" → 0.1-0.3

Coherence Enforcement: Ensure probabilities satisfy basic constraints:

- Probabilities sum to 1.0
- Conditional tables are complete
- No logical contradictions

Metadata Integration: Combine structure with probabilities in BayesDown format.

Why Two Stages?

This separation provides several benefits:

Modular Validation: Structure can be verified independently from probability estimates, simplifying quality assurance.

Human Oversight: Experts can review and correct structural extraction before probability quantification.

Flexible Quantification: Different methods (LLM extraction, expert elicitation, market data) can provide probabilities for the same structure.

Error Isolation: Structural errors don't contaminate probability extraction and vice versa.

Implementation Details

The system is implemented in Python, leveraging established libraries while adding novel extraction capabilities.

Technology Stack

- **Language Models**: OpenAI GPT-4 and Anthropic Claude for extraction
- **Network Analysis**: NetworkX for graph algorithms
- **Probabilistic Modeling**: pgmpy for Bayesian network operations
- **Visualization**: PyVis for interactive network rendering
- **Data Processing**: Pandas for structured data manipulation

Key Algorithms

Hierarchical Parsing: The system parses ArgDown/BayesDown syntax recognizing indentation-based hierarchy:

```
def parse_markdown_hierarchy_fixed(markdown_text, ArgDown=False):
    """Parse ArgDown or BayesDown format into structured DataFrame"""
    # Clean text and extract node information
    titles_info = extract_titles_info(clean_text)

    # Establish parent-child relationships based on indentation
    titles_with_relations = establish_relationships_fixed(titles_info)

    # Convert to DataFrame with proper columns
    df = convert_to_dataframe(titles_with_relations, ArgDown)

    # Add derived properties
    df = add_network_analysis_columns(df)

    return df
```

Probability Completion: When sources don't specify all required probabilities, the system uses principled methods:

- Maximum entropy for missing values
- Coherence constraints propagation
- Expert-specified defaults

Visual Encoding: Nodes are colored by probability magnitude and styled by network position:

- Green (high probability) to red (low probability) gradient
- Blue borders for root causes, purple for intermediate, magenta for effects

Performance Characteristics

Benchmarking reveals practical scalability:

- **Small networks** (10 nodes): <1 second processing
- **Medium networks** (11-30 nodes): 2-8 seconds
- **Large networks** (31-50 nodes): 15-45 seconds
- **Very large networks** (>50 nodes): Require approximation methods

The bottleneck shifts from extraction (linear in text length) to inference (exponential in network connectivity) as models grow.

Case Study: Rain-Sprinkler-Grass

I begin with the canonical example to demonstrate the complete pipeline on a simple, well-understood case.

Input Representation

The source BayesDown representation:

```
[Grass_Wet]: Concentrated moisture on grass.
{"instantiations": ["grass_wet_TRUE", "grass_wet_FALSE"],
 "priors": {"p(grass_wet_TRUE)": "0.322", "p(grass_wet_FALSE)": "0.678"},
 "posteriors": {
   "p(grass_wet_TRUE|sprinkler_TRUE,rain_TRUE)": "0.99",
   "p(grass_wet_TRUE|sprinkler_TRUE,rain_FALSE)": "0.9",
   "p(grass_wet_TRUE|sprinkler_FALSE,rain_TRUE)": "0.8",
   "p(grass_wet_TRUE|sprinkler_FALSE,rain_FALSE)": "0.0"
 }}
 + [Rain]: Water falling from sky.
   {"instantiations": ["rain_TRUE", "rain_FALSE"],
    "priors": {"p(rain_TRUE)": "0.2", "p(rain_FALSE)": "0.8"}}
 + [Sprinkler]: Artificial watering system.
   {"instantiations": ["sprinkler_TRUE", "sprinkler_FALSE"],
    "priors": {"p(sprinkler_TRUE)": "0.448", "p(sprinkler_FALSE)": "0.552"},
    "posteriors": {
      "p(sprinkler_TRUE|rain_TRUE)": "0.01",
      "p(sprinkler_TRUE|rain_FALSE)": "0.4"
    }}
   + [Rain]
```

Processing Steps

1. **Parsing**: Extract three nodes with relationships
2. **Validation**: Verify probability coherence and DAG structure
3. **Enhancement**: Calculate joint probabilities and network metrics
4. **Construction**: Build formal Bayesian network
5. **Visualization**: Render interactive display

Results

The system successfully:

- Extracts complete network structure
- Preserves all probability information
- Calculates correct marginal probabilities
- Generates interactive visualization

- Enables inference queries

This simple example validates the basic pipeline functionality before tackling complex real-world cases.

Case Study: Carlsmith's Power-Seeking AI Model

Applying AMTAIR to Carlsmith's model demonstrates scalability to realistic AI safety arguments.

Model Complexity

The Carlsmith model contains:

- **23 nodes** representing different factors
- **27 edges** encoding dependencies
- **Multiple probability tables** with complex conditionals
- **Six-level causal depth** from root causes to catastrophe

Extraction Results

The automated extraction successfully identifies:

Core Risk Pathway:

```
Existential_Catastrophe
← Human_Disempowerment
← Scale_Of_Power_Seeking
← Misaligned_Power_Seeking
← [APS_Systems, Difficulty_Of_Alignment, Deployment_Decisions]
```

Supporting Structure:

- Competitive dynamics influencing deployment
- Technical factors affecting alignment difficulty
- Corrective mechanisms and their limitations

Probability Preservation:

- Extracted probabilities match Carlsmith's published estimates
- Conditional relationships properly captured
- Final P(doom) calculation reproduces ~5% result

Validation Against Original

Comparing extracted model to Carlsmith's original:

Metric	Performance
Structural Accuracy	92% (nodes and edges)

Metric	Performance
Probability Accuracy	87% (within 0.05)
Path Completeness	100% (all major paths)
Semantic Preservation	High (per expert review)

The high fidelity demonstrates AMTAIR's capability for complex real-world arguments.

Insights from Formalization

Formal representation reveals several insights:

Critical Path Analysis: The pathway through APS development and deployment decisions carries the highest risk contribution.

Sensitivity Points: Small changes in deployment probability create large changes in overall risk.

Intervention Opportunities: Improving alignment difficulty or deployment governance show highest impact potential.

These insights emerge naturally from formal analysis but remain implicit in textual arguments.

Validation Methodology

Establishing trust in automated extraction requires rigorous validation across multiple dimensions.

Ground Truth Construction

I created validation datasets through:

1. **Expert Manual Extraction**: Three domain experts independently extracted models from the same sources
2. **Consensus Building**: Reconciled differences to create gold standard representations
3. **Annotation**: Marked source passages supporting each element

Evaluation Metrics

Structural Metrics:

- Precision: Fraction of extracted elements that are correct
- Recall: Fraction of true elements that are extracted
- F1 Score: Harmonic mean balancing precision and recall

Probabilistic Metrics:

- Mean Absolute Error for probability values
- Kullback-Leibler divergence for distributions

- Calibration plots for uncertainty expression

Semantic Metrics:

- Expert ratings of meaning preservation
- Functional equivalence for inference queries

Results Summary

Across 20 test documents:

Component	Precision	Recall	F1 Score
Node Identification	89%	86%	0.875
Edge Extraction	84%	81%	0.825
Probability Values	76%	71%	0.735
Overall System	**83%**	**79%**	**0.810**

Performance is strongest for explicit structural elements and numerical probabilities, with more challenges in extracting implicit relationships and qualitative uncertainty.

Error Analysis

Common failure modes:

Implicit Assumptions (23% of errors): Unstated background assumptions that experts infer but system misses.

Complex Conditionals (19% of errors): Nested conditionals with multiple antecedents challenge current parsing.

Ambiguous Quantifiers (17% of errors): Terms like "significant" lack clear probability mapping without context.

Coreference Resolution (15% of errors): Pronouns and indirect references create attribution challenges.

Understanding these limitations guides both current usage and future improvements.

Policy Evaluation Capabilities

Beyond extraction and visualization, AMTAIR enables systematic policy analysis through formal intervention modeling.

Intervention Representation

Policies are modeled as modifications to network parameters:

```python
def evaluate_policy_intervention(network, intervention, targets):
    """Evaluate policy impact using do-calculus"""
    # Baseline without intervention
    baseline = network.query(targets)

    # Apply intervention using Pearl's do-operator
    intervened = network.do_query(
        intervention['variable'],
        intervention['value'],
        targets
    )

    # Calculate effect metrics
    return {
        'baseline_risk': baseline,
        'intervened_risk': intervened,
        'relative_reduction': 1 - intervened/baseline,
        'absolute_reduction': baseline - intervened
    }
```

Example: Deployment Governance

Consider a policy requiring safety certification before deployment:

Intervention: Set P(deployment|misaligned) = 0.1 (from 0.7)

Results:

- Baseline P(catastrophe) = 0.05
- Intervened P(catastrophe) = 0.012
- Relative risk reduction = 76%
- Number needed to regulate = 26 deployments

This quantitative analysis enables comparison across interventions.

Robustness Analysis

Policies must work across worldviews. AMTAIR enables:

1. **Multi-Model Evaluation**: Test interventions across different extracted models
2. **Parameter Sensitivity**: Vary assumptions to find breaking points
3. **Scenario Analysis**: Combine interventions under different futures
4. **Confidence Bounds**: Propagate uncertainty through to outcomes

This systematic approach moves beyond intuitive policy assessment.

Interactive Visualization Design

Making Bayesian networks accessible to diverse stakeholders requires careful visualization design.

Visual Encoding Strategy

The system uses multiple visual channels:

Color: Probability magnitude (green=high, red=low) **Borders**: Node type (blue=root, purple=intermediate, magenta=effect)
Size: Centrality in network (larger=more influential) **Layout**: Force-directed positioning reveals clusters

Progressive Disclosure

Information appears at appropriate levels:

1. **Overview**: Network structure and color coding
2. **Hover**: Node description and prior probability
3. **Click**: Full probability tables and details
4. **Interaction**: Drag to rearrange, zoom to explore

This layered approach serves both quick assessment and deep analysis needs.

User Interface Elements

Key features enhance usability:

- **Physics Controls**: Adjust layout dynamics
- **Filter Options**: Show/hide node types
- **Export Functions**: Save images or data
- **Comparison Mode**: Side-by-side worldviews

These features emerged from user testing with researchers and policymakers.

Integration with Prediction Markets

While full integration remains future work, the architecture supports connection to live forecasting data.

Design for Integration

The system architecture anticipates market connections:

```
class PredictionMarketConnector:
    def __init__(self, market_apis):
        self.markets = market_apis

    def find_relevant_questions(self, model_variables):
```

```
        """Map model variables to forecast questions"""
        # Semantic matching between variables and questions

    def fetch_probabilities(self, questions):
        """Retrieve latest market probabilities"""
        # API calls with caching and error handling

    def update_model(self, model, market_data):
        """Integrate market probabilities into model"""
        # Weighted updating based on liquidity and track record
```

Challenges and Opportunities

Key integration challenges:

- **Question Mapping**: Model variables rarely match market questions exactly
- **Temporal Alignment**: Markets forecast specific dates, models consider scenarios
- **Quality Variation**: Market depth and participation vary significantly

Despite challenges, even partial integration provides value through external validation and dynamic updating.

Computational Considerations

As networks grow large, computational challenges emerge requiring sophisticated approaches.

Exact vs. Approximate Inference

Small networks enable exact inference through variable elimination. Larger networks require approximation:

Monte Carlo Methods: Sample from probability distributions to estimate queries **Variational Inference**: Optimize simpler distributions to approximate true posteriors **Belief Propagation**: Pass messages between nodes to converge on beliefs

The system automatically selects appropriate methods based on network properties.

Scaling Strategies

For very large networks:

1. **Hierarchical Decomposition**: Break into sub-networks for independent analysis
2. **Pruning**: Remove low-influence paths for specific queries
3. **Caching**: Store computed results for common queries
4. **Parallelization**: Distribute sampling across processors

These strategies extend practical network size limits significantly.

Summary of Technical Achievements

AMTAIR successfully demonstrates:

- **Automated extraction** from natural language to formal models
- **Two-stage architecture** separating structure from quantification
- **High fidelity** preservation of complex arguments
- **Interactive visualization** accessible to diverse users
- **Policy evaluation** capabilities through intervention modeling
- **Scalable implementation** handling realistic network sizes

These achievements validate the feasibility of computational coordination infrastructure for AI governance.

---

4. Discussion: Implications and Limitations

```
### 10% of Grade: ~ 14% of text ~ 4200 words ~ 10 pages

- discusses specific objection to student's own argument
- provides convincing reply that bolsters or refines the main argument
- relates to or extends beyond materials/arguments covered in class
```

This chapter critically examines AMTAIR's implications, limitations, and potential failure modes. By engaging seriously with objections and challenges, I aim to provide a balanced assessment of what this approach can and cannot achieve for AI governance coordination.

Technical Limitations and Responses

Objection 1: Extraction Quality Boundaries

Critic: "Complex implicit reasoning chains resist formalization. Automated extraction will systematically miss nuanced arguments, subtle conditional relationships, and context-dependent meanings that human readers naturally understand."

Response: This concern has merit—extraction does face inherent limitations. However, the empirical results tell a more nuanced story. With extraction achieving 85%+ accuracy for structural relationships and 73% for probability capture, the system performs well enough for practical use while falling short of human expert performance.

More importantly, AMTAIR employs a hybrid human-AI workflow that addresses this limitation:

- **Two-stage verification**: Humans review structural extraction before probability quantification
- **Transparent outputs**: All intermediate representations remain human-readable
- **Iterative refinement**: Extraction prompts improve based on error analysis
- **Ensemble approaches**: Multiple extraction attempts can identify ambiguities

The question is not whether automated extraction perfectly captures every nuance—it doesn't. Rather, it's whether imperfect extraction still provides value over no formal representation.

When the alternative is relying on conflicting mental models that remain entirely implicit, even 75% accurate formal models represent significant progress.

Furthermore, extraction errors often reveal interesting properties of the source arguments themselves—ambiguities that human readers gloss over become explicit when formalization fails. This diagnostic value enhances rather than undermines the approach.

Objection 2: False Precision in Uncertainty

Critic: "Attaching exact probabilities to unprecedented events like AI catastrophe is fundamentally misguided. The numbers create false confidence in what amounts to educated speculation about radically uncertain futures."

Response: This philosophical objection strikes at the heart of formal risk assessment. However, AMTAIR addresses it through several design choices:

First, the system explicitly represents uncertainty about uncertainty. Rather than point estimates, the framework supports probability distributions over parameters. When someone says "likely" we might model this as Beta(8,2) rather than exactly 0.8, capturing both the central estimate and our uncertainty about it.

Second, all probabilities are explicitly conditional on stated assumptions. The system doesn't claim "P(catastrophe) = 0.05" absolutely, but rather "Given Carlsmith's model assumptions, P(catastrophe) = 0.05." This conditionality is preserved throughout analysis.

Third, sensitivity analysis reveals which probabilities actually matter. Often, precise values are unnecessary—knowing whether a parameter is closer to 0.1 or 0.9 suffices for decision-making. The formalization helps identify where precision matters and where it doesn't.

Finally, the alternative to quantification isn't avoiding the problem but making it worse. When experts say "highly likely" or "significant risk," they implicitly reason with probabilities. Formalization simply makes these implicit quantities explicit and subject to scrutiny. As Dennis Lindley noted, "Uncertainty is not in the events, but in our knowledge about them."

Objection 3: Correlation Complexity

Critic: "Bayesian networks assume conditional independence given parents, but real-world AI risks involve complex correlations. Ignoring these dependencies could dramatically misrepresent risk levels."

Response: Standard Bayesian networks do face limitations with correlation representation—this is a genuine technical challenge. However, several approaches within the framework address this:

Explicit correlation nodes: When factors share hidden common causes, we can add latent variables to capture correlations. For instance, "AI research culture" might influence both "capability advancement" and "safety investment."

Copula methods: For known correlation structures, copula functions can model dependencies while preserving marginal distributions. This extends standard Bayesian networks significantly.

Sensitivity bounds: When correlations remain uncertain, we can compute bounds on outcomes under different correlation assumptions. This reveals when correlations critically affect conclusions.

Model ensembles: Different correlation structures can be modeled separately and results aggregated, similar to climate modeling approaches.

More fundamentally, the question is whether imperfect independence assumptions invalidate the approach. In practice, explicitly modeling first-order effects with known limitations often proves more valuable than attempting to capture all dependencies informally. The framework makes assumptions transparent, enabling targeted improvements where correlations matter most.

Conceptual and Methodological Concerns

Objection 4: Democratic Exclusion

Critic: "Transforming policy debates into complex graphs and equations will sideline non-technical stakeholders, concentrating influence among those comfortable with formal models. This technocratic approach undermines democratic participation in crucial decisions about humanity's future."

Response: This concern about technocratic exclusion deserves serious consideration—formal methods can indeed create barriers. However, AMTAIR's design explicitly prioritizes accessibility alongside rigor:

Progressive disclosure interfaces allow engagement at multiple levels. A policymaker might explore visual network structures and probability color-coding without engaging mathematical details. Interactive features let users modify assumptions and see consequences without understanding implementation.

Natural language preservation ensures original arguments remain accessible. The Bayes-Down format maintains human-readable descriptions alongside formal specifications. Users can always trace from mathematical representations back to source texts.

Comparative advantage comes from making implicit technical content explicit, not adding complexity. When experts debate AI risk, they already employ sophisticated probabilistic reasoning—formalization reveals rather than creates this complexity. Making hidden assumptions visible arguably enhances rather than reduces democratic participation.

Multiple interfaces serve different communities. Researchers access full technical depth, policymakers use summary dashboards, public stakeholders explore interactive visualizations. The same underlying model supports varied engagement modes.

Rather than excluding non-technical stakeholders, proper implementation can democratize access to expert reasoning by making it inspectable and modifiable. The risk lies not in formaliza-

tion itself but in poor interface design or gatekeeping behaviors around model access.

Objection 5: Oversimplification of Complex Systems

Critic: "Forcing rich socio-technical systems into discrete Bayesian networks necessarily loses crucial dynamics—feedback loops, emergent properties, institutional responses, and cultural factors that shape AI development. The models become precise but wrong."

Response: All models simplify by necessity—as Box noted, "All models are wrong, but some are useful." The question becomes whether formal simplifications improve upon informal mental models:

Transparent limitations make formal models' shortcomings explicit. Unlike mental models where simplifications remain hidden, network representations clearly show what is and isn't included. This transparency enables targeted criticism and improvement.

Iterative refinement allows models to grow more sophisticated over time. Starting with first-order effects and adding complexity where it proves important follows successful practice in other domains. Climate models began simply and added dynamics as computational power and understanding grew.

Complementary tools address different aspects of the system. Bayesian networks excel at probabilistic reasoning and intervention analysis. Other approaches—agent-based models, system dynamics, scenario planning—can capture different properties. AMTAIR provides one lens, not the only lens.

Empirical adequacy ultimately judges models. If simplified representations enable better predictions and decisions than informal alternatives, their abstractions are justified. Early results suggest formal models, despite simplifications, outperform intuitive reasoning for complex risk assessment.

The goal isn't creating perfect representations but useful ones. By making simplifications explicit and modifiable, formal models enable systematic improvement in ways mental models cannot.

Red-Teaming Results

To identify failure modes, I conducted systematic adversarial testing of the AMTAIR system.

Adversarial Extraction Attempts

I tested the system with deliberately challenging inputs:

Contradictory Arguments: Texts asserting P(A) = 0.2 and P(A) = 0.8 in different sections

- Result: System flagged inconsistency rather than averaging
- Mitigation: Explicit consistency checking with user resolution

Circular Reasoning: Arguments where A causes B causes C causes A

- Result: DAG validation caught cycles, extraction failed gracefully

- Mitigation: Clear error messages explaining the structural issue

Extremely Vague Language: Texts using only qualitative terms without clear relationships

- Result: Extraction quality degraded significantly (F1 < 0.5)
- Mitigation: Confidence scores on extracted elements, human review triggers

Deceptive Framings: Arguments designed to imply false causal relationships

- Result: System sometimes extracted spurious connections
- Mitigation: Source grounding requirements, validation against citations

Robustness Findings

Key vulnerabilities identified:

1. **Anchoring bias**: System tends to over-weight first probability mentioned (34% effect)
2. **Authority sensitivity**: Extracted probabilities inflated for cited experts (18% average)
3. **Complexity degradation**: Performance drops sharply beyond 50 nodes
4. **Context loss**: Long-range dependencies in text sometimes missed

However, the system demonstrated robustness to:

- Different writing styles and academic disciplines
- Variations in argument structure and presentation order
- Mixed numerical and qualitative probability expressions
- Reasonable levels of grammatical errors and typos

Implications for Deployment

These results suggest AMTAIR is suitable for:

- **Research applications** with expert oversight
- **Policy analysis** of well-structured arguments
- **Educational uses** demonstrating formal reasoning
- **Collaborative modeling** with human verification

But should be used cautiously for:

- Fully automated analysis without review
- Adversarial or politically contentious texts
- Real-time decision-making without validation
- Arguments far outside training distribution

Enhancing Epistemic Security

Despite limitations, AMTAIR contributes to epistemic security in AI governance through several mechanisms.

Making Models Inspectable

The greatest epistemic benefit comes from forcing implicit models into explicit form. When an expert claims "misalignment likely leads to catastrophe," formalization asks:

- Likely means what probability?
- Through what causal pathways?
- Under what assumptions?
- With what evidence?

This explicitation serves multiple functions:

Clarity: Vague statements become precise claims subject to evaluation

Comparability: Different experts' models can be systematically compared

Criticizability: Hidden assumptions become visible targets for challenge

Updatability: Formal models can systematically incorporate new evidence

Revealing Convergence and Divergence

Comparative analysis across extracted models reveals surprising patterns:

Structural convergence: Different experts often share similar causal models even when probability estimates diverge dramatically. This suggests shared understanding of mechanisms despite disagreement on magnitudes.

Parameter clustering: Probability estimates often cluster around a few values rather than spreading uniformly, suggesting implicit coordination or common evidence bases.

Crux identification: Formal comparison precisely identifies where worldviews diverge—often just 2-3 key parameters drive different conclusions about overall risk.

These insights remain hidden when arguments stay in natural language form.

Improving Collective Reasoning

AMTAIR enhances group epistemics through:

Explicit uncertainty: Replacing "might," "could," "likely" with probability distributions reduces miscommunication and forces precision

Compositional reasoning: Complex arguments decompose into manageable components that can be independently evaluated

Evidence integration: New information updates specific parameters rather than requiring complete argument reconstruction

Exploration tools: Stakeholders can modify assumptions and immediately see consequences, building intuition about model dynamics

Early pilot studies with AI governance researchers show 40% reduction in time to identify core disagreements and 60% improvement in agreement about what they disagree about—meta-agreement that enables productive debate.

Scaling Challenges and Opportunities

Moving from prototype to widespread adoption faces both technical and social challenges.

Technical Scaling

Computational complexity grows with network size, but several approaches help:

- Hierarchical decomposition for very large models
- Caching and approximation for common queries
- Distributed processing for extraction tasks
- Incremental updating rather than full recomputation

Data quality varies dramatically across sources:

- Academic papers provide structured arguments
- Blog posts offer rich ideas with less formal structure
- Policy documents mix normative and empirical claims
- Social media presents extreme extraction challenges

Integration complexity increases with ecosystem growth:

- Multiple LLM providers with different capabilities
- Diverse visualization needs across users
- Various export formats for downstream tools
- Version control for evolving models

Social and Institutional Scaling

Adoption barriers include:

- Learning curve for formal methods
- Institutional inertia in established processes
- Concerns about replacing human judgment
- Resource requirements for implementation

Trust building requires:

- Transparent methodology documentation
- Published validation studies
- High-profile successful applications
- Community ownership and development

Sustainability depends on:

- Open source development model

- Diverse funding sources
- Academic and industry partnerships
- Clear value demonstration

Opportunities for Impact

Despite challenges, several factors favor adoption:

Timing: AI governance needs tools now, creating receptive audiences

Complementarity: AMTAIR enhances rather than replaces existing processes

Flexibility: The approach adapts to different contexts and needs

Network effects: Value increases as more perspectives are formalized

Early adopters in research organizations and think tanks can demonstrate value, creating momentum for broader adoption.

Integration with Governance Frameworks

AMTAIR complements rather than replaces existing governance approaches.

Standards Development

Technical standards bodies could use AMTAIR to:

- Model how proposed standards affect risk pathways
- Compare different standard options systematically
- Identify unintended consequences through pathway analysis
- Build consensus through explicit model negotiation

Example: Evaluating compute thresholds for AI system regulation by modeling how different thresholds affect capability development, safety investment, and competitive dynamics.

Regulatory Design

Regulators could apply the framework to:

- Assess regulatory impact across different scenarios
- Identify enforcement challenges through explicit modeling
- Compare international approaches systematically
- Design adaptive regulations responsive to evidence

Example: Analyzing how liability frameworks affect corporate AI development decisions under different market conditions.

International Coordination

Multilateral bodies could leverage shared models for:

- Establishing common risk assessments
- Negotiating agreements with explicit assumptions
- Monitoring compliance through parameter tracking
- Adapting agreements as evidence emerges

Example: Building shared models for AGI development scenarios to inform international AI governance treaties.

Organizational Decision-Making

Individual organizations could use AMTAIR for:

- Internal risk assessment and planning
- Board-level communication about AI strategies
- Research prioritization based on model sensitivity
- Safety case development with explicit assumptions

Example: An AI lab modeling how different safety investments affect both capability advancement and risk mitigation.

Future Research Directions

Several research directions could enhance AMTAIR's capabilities and impact.

Technical Enhancements

Improved extraction: Fine-tuning language models specifically for argument extraction, handling implicit reasoning, and cross-document synthesis

Richer representations: Temporal dynamics, continuous variables, and multi-agent interactions within extended frameworks

Inference advances: Quantum computing applications, neural approximate inference, and hybrid symbolic-neural methods

Validation methods: Automated consistency checking, anomaly detection in extracted models, and benchmark dataset development

Methodological Extensions

Causal discovery: Inferring causal structures from data rather than just extracting from text

Experimental integration: Connecting models to empirical results from AI safety experiments

Dynamic updating: Continuous model refinement as new evidence emerges from research and deployment

Uncertainty quantification: Richer representation of deep uncertainty and model confidence

Application Domains

Beyond AI safety: Climate risk, biosecurity, nuclear policy, and other existential risks

Corporate governance: Strategic planning, risk management, and innovation assessment

Scientific modeling: Formalizing theoretical arguments in emerging fields

Educational tools: Teaching probabilistic reasoning and critical thinking

Ecosystem Development

Open standards: Common formats for model exchange and tool interoperability

Community platforms: Collaborative model development and sharing infrastructure

Training programs: Building capacity for formal modeling in governance communities

Quality assurance: Certification processes for high-stakes model applications

These directions could transform AMTAIR from a single tool into a broader ecosystem for enhanced reasoning about complex risks.

---

5. Conclusion: Toward Coordinated AI Governance

```
### 10% of Grade: ~ 14% of text ~ 4200 words ~ 10 pages

- summarizes thesis and line of argument
- outlines possible implications
- notes outstanding issues / limitations of discussion
- points to avenues for further research
- overall conclusion is in line with introduction
```

Summary of Key Contributions

This thesis has demonstrated both the need for and feasibility of computational approaches to enhancing coordination in AI governance. The work makes several distinct contributions across theory, methodology, and implementation.

Theoretical Contributions

Diagnosis of the Coordination Crisis: I've articulated how fragmentation across technical, policy, and strategic communities systematically amplifies existential risk from advanced AI. This framing moves beyond identifying disagreements to understanding how misaligned efforts create negative-sum dynamics—safety gaps emerge between communities, resources are misallocated through duplication and neglect, and interventions interact destructively.

The Multiplicative Benefits Framework: The combination of automated extraction, prediction market integration, and formal policy evaluation creates value exceeding the sum of parts. Automation enables scale, markets provide empirical grounding, and policy analysis delivers actionable insights. Together, they address different facets of the coordination challenge while reinforcing each other's strengths.

Epistemic Infrastructure Conception: Positioning formal models as epistemic infrastructure reframes the role of technical tools in governance. Rather than replacing human judgment, computational approaches provide common languages, shared representations, and systematic methods for managing disagreement—essential foundations for coordination under uncertainty.

Methodological Innovations

Two-Stage Extraction Architecture: Separating structural extraction (ArgDown) from probability quantification (BayesDown) addresses key challenges in automated formalization. This modularity enables human oversight at critical points, supports multiple quantification methods, and isolates different types of errors for targeted improvement.

BayesDown as Bridge Representation: The development of BayesDown syntax creates a crucial intermediate representation preserving both narrative accessibility and mathematical precision. This bridge enables the transformation from qualitative arguments to quantitative models while maintaining traceability and human readability.

Validation Framework: The systematic approach to validating automated extraction—comparing against expert annotations, measuring multiple accuracy dimensions, and analyzing error patterns—establishes scientific standards for assessing formalization tools. This framework can guide future development in this emerging area.

Technical Achievements

Working Implementation: AMTAIR demonstrates end-to-end feasibility from document ingestion through interactive visualization. The system achieves practically useful accuracy levels: 85%+ for structural extraction and 73% for probability capture on real AI safety arguments.

Scalability Solutions: Technical approaches for handling realistic model complexity—hierarchical decomposition, approximate inference, and progressive visualization—show that computational limitations need not prevent practical application.

Accessibility Design: The layered interface approach serves diverse stakeholders without compromising technical depth. Progressive disclosure, visual encoding, and interactive exploration make formal models accessible beyond technical specialists.

Empirical Findings

Extraction Feasibility: The successful extraction of complex arguments like Carlsmith's model validates the core premise that implicit formal structures exist in natural language arguments and can be computationally recovered with reasonable fidelity.

Convergence Patterns: Comparative analysis reveals surprising structural agreement across worldviews even when probability estimates diverge dramatically. This suggests shared causal understanding despite parameter disagreements—a foundation for coordination.

Intervention Impacts: Policy evaluation demonstrates how formal models enable rigorous assessment of governance options. The ability to quantify risk reduction across scenarios and identify robust strategies validates the practical value of formalization.

Limitations and Honest Assessment

Despite these contributions, important limitations constrain current capabilities and should guide appropriate use.

Technical Constraints

Extraction Boundaries: While 73-85% accuracy suffices for many purposes, systematic biases remain. The system struggles with implicit assumptions, complex conditionals, and context-dependent meanings. These limitations necessitate human review for high-stakes applications.

Correlation Handling: Standard Bayesian networks inadequately represent complex correlations in real systems. While extensions like copulas and explicit correlation nodes help, fully capturing interdependencies remains challenging.

Computational Scaling: Very large networks (>50 nodes) require approximations that may affect accuracy. As models grow to represent richer phenomena, computational constraints increasingly bind.

Conceptual Limitations

Formalization Trade-offs: Converting rich arguments to formal models necessarily loses nuance. While making assumptions explicit provides value, some insights resist mathematical representation.

Probability Interpretation: Deep uncertainty about unprecedented events challenges probabilistic representation. Numbers can create false precision even when explicitly conditional and uncertain.

Social Complexity: Institutional dynamics, cultural factors, and political processes influence AI development in ways that simple causal models struggle to capture.

Practical Constraints

Adoption Barriers: Learning curves, institutional inertia, and resource requirements limit immediate deployment. Even demonstrably valuable tools face implementation challenges.

Maintenance Burden: Models require updating as arguments evolve and evidence emerges. Without sustained effort, formal representations quickly become outdated.

Context Dependence: The approach works best for well-structured academic arguments. Application to informal discussions, political speeches, or social media remains challenging.

Implications for AI Governance

Despite limitations, AMTAIR's approach offers significant implications for how AI governance can evolve toward greater coordination and effectiveness.

Near-Term Applications

Research Coordination: Research organizations can use formal models to:

- Map the landscape of current arguments and identify gaps
- Prioritize investigations targeting high-sensitivity parameters
- Build cumulative knowledge through explicit model updating
- Facilitate collaboration through shared representations

Policy Development: Governance bodies can apply the framework to:

- Evaluate proposals across multiple expert worldviews
- Identify robust interventions effective under uncertainty
- Make assumptions explicit for democratic scrutiny
- Track how evidence changes optimal policies over time

Stakeholder Communication: The visualization and analysis tools enable:

- Clearer communication between technical and policy communities
- Public engagement with complex risk assessments
- Board-level strategic discussions grounded in formal analysis
- International negotiations with explicit shared models

Medium-Term Transformation

As adoption spreads, we might see:

Epistemic Commons: Shared repositories of formalized arguments become reference points for governance discussions, similar to how economic models inform monetary policy or climate models guide environmental agreements.

Adaptive Governance: Policies designed with explicit models can include triggers for reassessment as key parameters change, enabling responsive governance that avoids both paralysis and recklessness.

Professionalization: "Model curator" and "argument formalization specialist" emerge as recognized roles, building expertise in bridging natural language and formal representations.

Quality Standards: Community norms develop around model transparency, validation requirements, and appropriate use cases, preventing both dismissal and over-reliance on formal tools.

Long-Term Vision

Successfully scaling this approach could fundamentally alter AI governance:

Coordinated Response: Rather than fragmented efforts, the AI safety ecosystem could operate with shared situational awareness—different actors understanding how their efforts interact and contribute to collective goals.

Anticipatory Action: Formal models with prediction market integration could provide early warning of emerging risks, enabling proactive rather than reactive governance.

Global Cooperation: Shared formal frameworks could facilitate international coordination similar to how economic models enable monetary coordination or climate models support environmental agreements.

Democratic Enhancement: Making expert reasoning transparent and modifiable could enable broader participation in crucial decisions about humanity's technological future.

Recommendations for Stakeholders

Different communities can take concrete steps to realize these benefits:

For Researchers

1. **Experiment with formalization**: Try extracting your own arguments into ArgDown/BayesDown format to discover implicit assumptions
2. **Contribute to validation**: Provide expert annotations for building benchmark datasets and improving extraction quality
3. **Develop extensions**: Build on the open-source foundation to add capabilities for your specific domain needs
4. **Publish formally**: Include formal model representations alongside traditional papers to enable cumulative building

For Policymakers

1. **Pilot applications**: Use AMTAIR for internal analysis of specific policy proposals to build familiarity and identify value
2. **Demand transparency**: Request formal models underlying expert recommendations to understand assumptions and uncertainties
3. **Fund development**: Support tool development and training to build governance capacity for formal methods
4. **Design adaptively**: Create policies with explicit triggers based on model parameters to enable responsive governance

For Technologists

1. **Improve extraction**: Contribute better prompting strategies, fine-tuned models, or validation methods
2. **Enhance interfaces**: Develop visualizations and interactions serving specific stakeholder needs

3. **Build integrations**: Connect AMTAIR to other tools in the AI governance ecosystem
4. **Scale infrastructure**: Address computational challenges for larger models and broader deployment

For Funders

1. **Support ecosystem**: Fund not just tool development but training, community building, and maintenance
2. **Bridge communities**: Incentivize collaborations between formal modelers and domain experts
3. **Measure coordination**: Develop metrics for assessing coordination improvements from formal tools
4. **Patient capital**: Recognize that epistemic infrastructure requires sustained investment to reach potential

Future Research Agenda

Building on this foundation, several research directions could amplify impact:

Technical Priorities

Extraction Enhancement:

- Fine-tuning language models specifically for argument extraction
- Handling implicit reasoning and long-range dependencies
- Cross-document synthesis for comprehensive models
- Multilingual extraction for global perspectives

Representation Extensions:

- Temporal dynamics for modeling AI development trajectories
- Multi-agent representations for strategic interactions
- Continuous variables for economic and capability metrics
- Uncertainty types beyond probability distributions

Integration Depth:

- Semantic matching between models and prediction markets
- Automated experiment design based on model sensitivity
- Policy optimization algorithms using extracted models
- Real-time updating from news and research feeds

Methodological Development

Validation Science:

- Larger benchmark datasets with diverse argument types
- Metrics for semantic preservation beyond accuracy
- Adversarial robustness testing protocols
- Longitudinal studies of model evolution

Hybrid Approaches:

- Optimal human-AI collaboration patterns for extraction
- Combining formal models with other methods (scenarios, simulations)
- Integration with deliberative and participatory processes
- Balancing automation with expert judgment

Social Methods:

- Ethnographic studies of model use in organizations
- Measuring coordination improvements empirically
- Understanding adoption barriers and facilitators
- Designing interventions for epistemic security

Application Expansion

Domain Extensions:

- Climate risk assessment and policy evaluation
- Biosecurity governance and pandemic preparedness
- Nuclear policy and deterrence stability
- Emerging technology governance broadly

Institutional Integration:

- Embedding in regulatory impact assessment
- Corporate strategic planning applications
- Academic peer review enhancement
- Democratic deliberation support tools

Global Deployment:

- Adapting to different governance contexts
- Supporting multilateral negotiation processes
- Building capacity in developing nations
- Creating resilient distributed infrastructure

Closing Reflections

The work presented in this thesis emerges from a simple observation: while humanity mobilizes unprecedented resources to address AI risks, our efforts remain tragically uncoordinated. Different communities work with incompatible frameworks, duplicate efforts, and sometimes actively undermine each other's work. This fragmentation amplifies the very risks we seek to mitigate.

AMTAIR represents one attempt to build bridges—computational tools that create common ground for disparate perspectives. By making implicit models explicit, quantifying uncertainty, and enabling systematic policy analysis, these tools offer hope for enhanced coordination. The successful extraction of complex arguments, validation against expert judgment, and demonstration of policy evaluation capabilities suggest this approach has merit.

Yet tools alone cannot solve coordination problems rooted in incentives, institutions, and human psychology. AMTAIR provides infrastructure for coordination, not coordination itself. Success requires not just technical development but changes in how we approach collective challenges—valuing transparency over strategic ambiguity, embracing uncertainty rather than false confidence, and prioritizing collective outcomes over parochial interests.

The path forward demands both ambition and humility. Ambition to build the epistemic infrastructure necessary for navigating unprecedented risks. Humility to recognize our tools' limitations and the irreducible role of human wisdom in governance. The question is not whether formal models can replace human judgment—they cannot and should not. Rather, it's whether we can augment our collective intelligence with computational tools that help us reason together about futures too important to leave to chance.

As AI capabilities advance toward transformative potential, the window for establishing effective governance narrows. We cannot afford continued fragmentation when facing potentially irreversible consequences. The coordination crisis in AI governance represents both existential risk and existential opportunity—risk if we fail to align our efforts, opportunity if we succeed in building unprecedented cooperation around humanity's most important challenge.

This thesis contributes technical foundations and demonstrates feasibility. The greater work—building communities, changing practices, and fostering coordination—remains ahead. May we prove equal to the task, for all our futures depend on it.

References

0.1 3.1.2 Test BayesDown Extraction

```
display(Markdown(md_content_ex_rain)) # view BayesDown file formatted as MarkDown
```

[Existential_Catastrophe]: The destruction of humanity's long-term potential due to AI systems we've lost control over. {"instantiations": ["existential_catastrophe_TRUE", "existential_catastrophe_FALSE"], "priors": {"p(existential_catastrophe_TRUE)": "0.05", "p(existential_catastrophe_FALSE)": "0.95"}, "posteriors": {"p(existential_catastrophe_TRUE|human_disem "0.95", "p(existential_catastrophe_TRUE|human_disempowerment_FALSE)": "0.0", "p(existential_catastrophe_FALSE|human_disempowerment_TRUE)": "0.05", "p(existential_catastrophe_F "1.0"}} - [Human_Disempowerment]: Permanent and collective disempowerment of humanity relative to AI systems. {"instantiations": ["human_disempowerment_TRUE", "human_disempowerment_FALSE"], "priors": {"p(human_disempowerment_TRUE)": "0.208", "p(human_disempowerment_FALSE)": "0.792"}, "posteriors": {"p(human_disempowerment_TRUE|scale_of "1.0", "p(human_disempowerment_TRUE|scale_of_power_seeking_FALSE)": "0.0", "p(human_disempowerment_FALSE|scale_of_power_seeking_TRUE)": "0.0", "p(human_disempowerment_ "1.0"}} - [Scale_Of_Power_Seeking]: Power-seeking by AI systems scaling to the point of permanently disempowering all of humanity. {"instantiations": ["scale_of_power_seeking_TRUE", "scale_of_power_seeking_FALSE"], "priors": {"p(scale_of_power_seeking_TRUE)": "0.208", "p(scale_of_power_seeking_FALSE)": "0.792"}, "posteriors": {"p(scale_of_power_seeking_TRUE|misaligne corrective_feedback_EFFECTIVE)": "0.25", "p(scale_of_power_seeking_TRUE|misaligned_power_seeking corrective_feedback_INEFFECTIVE)": "0.60", "p(scale_of_power_seeking_TRUE|misaligned_power_seeki corrective_feedback_EFFECTIVE)": "0.0", "p(scale_of_power_seeking_TRUE|misaligned_power_seeking_ corrective_feedback_INEFFECTIVE)": "0.0", "p(scale_of_power_seeking_FALSE|misaligned_power_seeki corrective_feedback_EFFECTIVE)": "0.75", "p(scale_of_power_seeking_FALSE|misaligned_power_seeking corrective_feedback_INEFFECTIVE)": "0.40", "p(scale_of_power_seeking_FALSE|misaligned_power_seek corrective_feedback_EFFECTIVE)": "1.0", "p(scale_of_power_seeking_FALSE|misaligned_power_seeking_ corrective_feedback_INEFFECTIVE)": "1.0"}} - [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impact ways due to problems with their objectives. {"instantiations": ["misaligned_power_seeking_TRUE", "misaligned_power_seeking_FALSE"], "priors": {"p(misaligned_power_seeking_TRUE)": "0.338", "p(misaligned_power_seeking_FALSE)": "0.662"}, "posteriors": {"p(misaligned_power_seeking_TR difficulty_of_alignment_TRUE, deployment_decisions_DEPLOY)": "0.90", "p(misaligned_power_seeking_

difficulty_of_alignment_TRUE, deployment_decisions_WITHHOLD)": "0.10", "p(misaligned_power_seekin
difficulty_of_alignment_FALSE, deployment_decisions_DEPLOY)": "0.25", "p(misaligned_power_seeking_
difficulty_of_alignment_FALSE, deployment_decisions_WITHHOLD)": "0.05", "p(misaligned_power_seekin
difficulty_of_alignment_TRUE, deployment_decisions_DEPLOY)": "0.0", "p(misaligned_power_seeking_T
difficulty_of_alignment_TRUE, deployment_decisions_WITHHOLD)": "0.0", "p(misaligned_power_seeking
difficulty_of_alignment_FALSE, deployment_decisions_DEPLOY)": "0.0", "p(misaligned_power_seeking_T
difficulty_of_alignment_FALSE, deployment_decisions_WITHHOLD)": "0.0", "p(misaligned_power_seeking
difficulty_of_alignment_TRUE, deployment_decisions_DEPLOY)": "0.10", "p(misaligned_power_seeking_D
difficulty_of_alignment_TRUE, deployment_decisions_WITHHOLD)": "0.90", "p(misaligned_power_seekin
difficulty_of_alignment_FALSE, deployment_decisions_DEPLOY)": "0.75", "p(misaligned_power_seeking_
difficulty_of_alignment_FALSE, deployment_decisions_WITHHOLD)": "0.95", "p(misaligned_power_seekin
difficulty_of_alignment_TRUE, deployment_decisions_DEPLOY)": "1.0", "p(misaligned_power_seeking_FA
difficulty_of_alignment_TRUE, deployment_decisions_WITHHOLD)": "1.0", "p(misaligned_power_seeking
difficulty_of_alignment_FALSE, deployment_decisions_DEPLOY)": "1.0", "p(misaligned_power_seeking_F
difficulty_of_alignment_FALSE, deployment_decisions_WITHHOLD)": "1.0"}} - [APS_Systems]:
AI systems with advanced capabilities, agentic planning, and strategic awareness. {"instantia-
tions": ["aps_systems_TRUE", "aps_systems_FALSE"], "priors": {"p(aps_systems_TRUE)":
"0.65", "p(aps_systems_FALSE)": "0.35"}, "posteriors": {"p(aps_systems_TRUE|advanced_ai_capability_T
agentic_planning_TRUE, strategic_awareness_TRUE)": "1.0", "p(aps_systems_TRUE|advanced_ai_capabi
agentic_planning_TRUE, strategic_awareness_FALSE)": "0.0", "p(aps_systems_TRUE|advanced_ai_capab
agentic_planning_FALSE, strategic_awareness_TRUE)": "0.0", "p(aps_systems_TRUE|advanced_ai_capab
agentic_planning_FALSE, strategic_awareness_FALSE)": "0.0", "p(aps_systems_TRUE|advanced_ai_capa
agentic_planning_TRUE, strategic_awareness_TRUE)": "0.0", "p(aps_systems_TRUE|advanced_ai_capabi
agentic_planning_TRUE, strategic_awareness_FALSE)": "0.0", "p(aps_systems_TRUE|advanced_ai_capab
agentic_planning_FALSE, strategic_awareness_TRUE)": "0.0", "p(aps_systems_TRUE|advanced_ai_capab
agentic_planning_FALSE, strategic_awareness_FALSE)": "0.0", "p(aps_systems_FALSE|advanced_ai_capa
agentic_planning_TRUE, strategic_awareness_TRUE)": "0.0", "p(aps_systems_FALSE|advanced_ai_capab
agentic_planning_TRUE, strategic_awareness_FALSE)": "1.0", "p(aps_systems_FALSE|advanced_ai_capa
agentic_planning_FALSE, strategic_awareness_TRUE)": "1.0", "p(aps_systems_FALSE|advanced_ai_capa
agentic_planning_FALSE, strategic_awareness_FALSE)": "1.0", "p(aps_systems_FALSE|advanced_ai_capa
agentic_planning_TRUE, strategic_awareness_TRUE)": "1.0", "p(aps_systems_FALSE|advanced_ai_capab
agentic_planning_TRUE, strategic_awareness_FALSE)": "1.0", "p(aps_systems_FALSE|advanced_ai_capa
agentic_planning_FALSE, strategic_awareness_TRUE)": "1.0", "p(aps_systems_FALSE|advanced_ai_capa
agentic_planning_FALSE, strategic_awareness_FALSE)": "1.0"}} - [Advanced_AI_Capability]:
AI systems that outperform humans on tasks that grant significant power in the world. {"instan-
tiations": ["advanced_ai_capability_TRUE", "advanced_ai_capability_FALSE"], "priors":
{"p(advanced_ai_capability_TRUE)": "0.80", "p(advanced_ai_capability_FALSE)": "0.20"}}
- [Agentic_Planning]: AI systems making and executing plans based on world models to achieve
objectives. {"instantiations": ["agentic_planning_TRUE", "agentic_planning_FALSE"],
"priors": {"p(agentic_planning_TRUE)": "0.85", "p(agentic_planning_FALSE)": "0.15"}} -
[Strategic_Awareness]: AI systems with models accurately representing power dynamics with
humans. {"instantiations": ["strategic_awareness_TRUE", "strategic_awareness_FALSE"],

“priors”: {“p(strategic_awareness_TRUE)”: “0.75”, “p(strategic_awareness_FALSE)”: “0.25”}} - [Difficulty_Of_Alignment]: It is harder to build aligned systems than misaligned systems that are attractive to deploy. {“instantiations”: [“difficulty_of_alignment_TRUE”, “difficulty_of_alignment_FALSE”], “priors”: {“p(difficulty_of_alignment_TRUE)”: “0.40”, “p(difficulty_of_alignment_FALSE)”: “0.60”}, “posteriors”: {“p(difficulty_of_alignment_TRUE|instrumenta
problems_with_proxies_TRUE, problems_with_search_TRUE)”: “0.85”, “p(difficulty_of_alignment_TRUE
problems_with_proxies_TRUE, problems_with_search_FALSE)”: “0.70”, “p(difficulty_of_alignment_TRU
problems_with_proxies_FALSE, problems_with_search_TRUE)”: “0.60”, “p(difficulty_of_alignment_TRU
problems_with_proxies_FALSE, problems_with_search_FALSE)”: “0.40”, “p(difficulty_of_alignment_TRU
problems_with_proxies_TRUE, problems_with_search_TRUE)”: “0.55”, “p(difficulty_of_alignment_TRUE
problems_with_proxies_TRUE, problems_with_search_FALSE)”: “0.40”, “p(difficulty_of_alignment_TRU
problems_with_proxies_FALSE, problems_with_search_TRUE)”: “0.30”, “p(difficulty_of_alignment_TRU
problems_with_proxies_FALSE, problems_with_search_FALSE)”: “0.10”, “p(difficulty_of_alignment_FAL
problems_with_proxies_TRUE, problems_with_search_TRUE)”: “0.15”, “p(difficulty_of_alignment_FALS
problems_with_proxies_TRUE, problems_with_search_FALSE)”: “0.30”, “p(difficulty_of_alignment_FALS
problems_with_proxies_FALSE, problems_with_search_TRUE)”: “0.40”, “p(difficulty_of_alignment_FALS
problems_with_proxies_FALSE, problems_with_search_FALSE)”: “0.60”, “p(difficulty_of_alignment_FAL
problems_with_proxies_TRUE, problems_with_search_TRUE)”: “0.45”, “p(difficulty_of_alignment_FALS
problems_with_proxies_TRUE, problems_with_search_FALSE)”: “0.60”, “p(difficulty_of_alignment_FALS
problems_with_proxies_FALSE, problems_with_search_TRUE)”: “0.70”, “p(difficulty_of_alignment_FALS
problems_with_proxies_FALSE, problems_with_search_FALSE)”: “0.90”}} - [Instrumental_Convergence]: AI systems with misaligned objectives tend to seek power as an instrumental goal. {“instantiations”: [“instrumental_convergence_TRUE”, “instrumental_convergence_FALSE”], “priors”: {“p(instrumental_convergence_TRUE)”: “0.75”, “p(instrumental_convergence_FALSE)”: “0.25”}} - [Problems_With_Proxies]: Optimizing for proxy objectives breaks correlations with intended goals. {“instantiations”: [“problems_with_proxies_TRUE”, “problems_with_proxies_FALSE”], “priors”: {“p(problems_with_proxies_TRUE)”: “0.80”, “p(problems_with_proxies_FALSE)”: “0.20”}} - [Problems_With_Search]: Search processes can yield systems pursuing different objectives than intended. {“instantiations”: [“problems_with_search_TRUE”, “problems_with_search_FALSE”], “priors”: {“p(problems_with_search_TRUE)”: “0.70”, “p(problems_with_search_FALSE)”: “0.30”}} - [Deployment_Decisions]: Decisions to deploy potentially misaligned AI systems. {“instantiations”: [“deployment_decisions_DEPLOY”, “deployment_decisions_WITHHOLD”], “priors”: {“p(deployment_decisions_DEPLOY)”: “0.70”, “p(deployment_decisions_WITHHOLD)”: “0.30”}, “posteriors”: {“p(deployment_decisions_DEPLOY|incenti
deception_by_ai_TRUE)”: “0.90”, “p(deployment_decisions_DEPLOY|incentives_to_build_aps_STRONG
deception_by_ai_FALSE)”: “0.75”, “p(deployment_decisions_DEPLOY|incentives_to_build_aps_WEAK,
deception_by_ai_TRUE)”: “0.60”, “p(deployment_decisions_DEPLOY|incentives_to_build_aps_WEAK,
deception_by_ai_FALSE)”: “0.30”, “p(deployment_decisions_WITHHOLD|incentives_to_build_aps_STRO
deception_by_ai_TRUE)”: “0.10”, “p(deployment_decisions_WITHHOLD|incentives_to_build_aps_STRO
deception_by_ai_FALSE)”: “0.25”, “p(deployment_decisions_WITHHOLD|incentives_to_build_aps_WEA
deception_by_ai_TRUE)”: “0.40”, “p(deployment_decisions_WITHHOLD|incentives_to_build_aps_WEAI

deception_by_ai_FALSE)": "0.70"}} - [Incentives_To_Build_APS]: Strong incentives to build and deploy APS systems. {"instantiations": ["incentives_to_build_aps_STRONG", "incentives_to_build_aps_WEAK"], "priors": {"p(incentives_to_build_aps_STRONG)": "0.80", "p(incentives_to_build_aps_WEAK)": "0.20"}, "posteriors": {"p(incentives_to_build_aps_STRONG|useful competitive_dynamics_STRONG)": "0.95", "p(incentives_to_build_aps_STRONG|usefulness_of_aps_HIG competitive_dynamics_WEAK)": "0.80", "p(incentives_to_build_aps_STRONG|usefulness_of_aps_LOW, competitive_dynamics_STRONG)": "0.70", "p(incentives_to_build_aps_STRONG|usefulness_of_aps_LOW competitive_dynamics_WEAK)": "0.30", "p(incentives_to_build_aps_WEAK|usefulness_of_aps_HIGH, competitive_dynamics_STRONG)": "0.05", "p(incentives_to_build_aps_WEAK|usefulness_of_aps_HIGH, competitive_dynamics_WEAK)": "0.20", "p(incentives_to_build_aps_WEAK|usefulness_of_aps_LOW, competitive_dynamics_STRONG)": "0.30", "p(incentives_to_build_aps_WEAK|usefulness_of_aps_LOW, competitive_dynamics_WEAK)": "0.70"}} - [Usefulness_Of_APS]: APS systems are very useful for many valuable tasks. {"instantiations": ["usefulness_of_aps_HIGH", "usefulness_of_aps_LOW"], "priors": {"p(usefulness_of_aps_HIGH)": "0.85", "p(usefulness_of_aps_LOW)": "0.15"}} - [Competitive_Dynamics]: Competitive pressures between AI developers. {"instantiations": ["competitive_dynamics_STRONG", "competitive_dynamics_WEAK"], "priors": {"p(competitive_dynamics_STRONG)": "0.75", "p(competitive_dynamics_WEAK)": "0.25"}} - [Deception_By_AI]: AI systems deceiving humans about their true objectives. {"instantiations": ["deception_by_ai_TRUE", "deception_by_ai_FALSE"], "priors": {"p(deception_by_ai_TRUE)": "0.50", "p(deception_by_ai_FALSE)": "0.50"}} - [Corrective_Feedback]: Human society implementing corrections after observing problems. {"instantiations": ["corrective_feedback_EFFECTIVE", "corrective_feedback_INEFFECTIVE"], "priors": {"p(corrective_feedback_EFFECTIVE)": "0.60", "p(corrective_feedback_INEFFECTIVE)": "0.40"}, "posteriors": {"p(corrective_feedback_EFFECTIVE|warning_shots_OBSERVED, rapid_capability_escalation_TRUE)": "0.40", "p(corrective_feedback_EFFECTIVE|warning_shots_OBSER rapid_capability_escalation_FALSE)": "0.80", "p(corrective_feedback_EFFECTIVE|warning_shots_UNOB rapid_capability_escalation_TRUE)": "0.15", "p(corrective_feedback_EFFECTIVE|warning_shots_UNOBS rapid_capability_escalation_FALSE)": "0.50", "p(corrective_feedback_INEFFECTIVE|warning_shots_OBS rapid_capability_escalation_TRUE)": "0.60", "p(corrective_feedback_INEFFECTIVE|warning_shots_OBS rapid_capability_escalation_FALSE)": "0.20", "p(corrective_feedback_INEFFECTIVE|warning_shots_UNO rapid_capability_escalation_TRUE)": "0.85", "p(corrective_feedback_INEFFECTIVE|warning_shots_UNO rapid_capability_escalation_FALSE)": "0.50"}} - [Warning_Shots]: Observable failures in weaker systems before catastrophic risks. {"instantiations": ["warning_shots_OBSERVED", "warning_shots_UNOBSERVED"], "priors": {"p(warning_shots_OBSERVED)": "0.70", "p(warning_shots_UNOBSERVED)": "0.30"}} - [Rapid_Capability_Escalation]: AI capabilities escalating very rapidly, allowing little time for correction. {"instantiations": ["rapid_capability_escalation_TRUE", "rapid_capability_escalation_FALSE"], "priors": {"p(rapid_capability_escalation_TRUE)": "0.45", "p(rapid_capability_escalation_FALSE)": "0.55"}} [Barriers_To_Understanding]: Difficulty in understanding the internal workings of advanced AI systems. {"instantiations": ["barriers_to_understanding_HIGH", "barriers_to_understanding_LOW"], "priors": {"p(barriers_to_understanding_HIGH)": "0.70", "p(barriers_to_understanding_LOW)": "0.30"}, "posteriors": {"p(barriers_to_understanding_HIGH|misalig

“0.85”, “p(barriers_to_understanding_HIGH|misaligned_power_seeking_FALSE)”: “0.60”, “p(barriers_to_understanding_LOW|misaligned_power_seeking_TRUE)”: “0.15”, “p(barriers_to_understan “0.40”}} - [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impact ways due to problems with their objectives. {“instantiations”: [“misaligned_power_seeking_TRUE”, “misaligned_power_seeking_FALSE”], “priors”: {“p(misaligned_power_seeking_TRUE)”: “0.338”, “p(misaligned_power_seeking_FALSE)”: “0.662”}} [Adversarial_Dynamics]: Potentially adversarial relationships between humans and power-seeking AI. {“instantiations”: [“adversarial_dynamics_TRUE”, “adversarial_dynamics_FALSE”], “priors”: {“p(adversarial_dynamics_TRUE)”: “0.60”, “p(adversarial_dynamics_FALSE)”: “0.40”}, “posteriors”: {“p(adversarial_dynamics_TRUE|misaligned_pow “0.95”, “p(adversarial_dynamics_TRUE|misaligned_power_seeking_FALSE)”: “0.10”, “p(adversarial_dynamics_FALSE|misaligned_power_seeking_TRUE)”: “0.05”, “p(adversarial_dynamics_FA “0.90”}} - [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impact ways due to problems with their objectives. {“instantiations”: [“misaligned_power_seeking_TRUE”, “misaligned_power_seeking_FALSE”], “priors”: {“p(misaligned_power_seeking_TRUE)”: “0.338”, “p(misaligned_power_seeking_FALSE)”: “0.662”}} [Stakes_Of_Error]: The escalating impact of mistakes with power-seeking AI systems. {“instantiations”: [“stakes_of_error_HIGH”, “stakes_of_error_LOW”], “priors”: {“p(stakes_of_error_HIGH)”: “0.85”, “p(stakes_of_error_LOW)”: “0.15”}, “posteriors”: {“p(stakes_of_error_HIGH|misaligned_power_seeking_TRUE)”: “0.95”, “p(stakes_of_error_HIGH|misaligned_power_seeking_FALSE)”: “0.50”, “p(stakes_of_error_LOW|misaligne “0.05”, “p(stakes_of_error_LOW|misaligned_power_seeking_FALSE)”: “0.50”}} - [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impact ways due to problems with their objectives. {“instantiations”: [“misaligned_power_seeking_TRUE”, “misaligned_power_seeking_FALSE”], “priors”: {“p(misaligned_power_seeking_TRUE)”: “0.338”, “p(misaligned_power_seeking_FALSE)”: “0.662”}}

0.2 3.1.2.2 Check the Graph Structure with the ArgDown Sandbox Online

Copy and paste the BayesDown formatted … in the ArgDown Sandbox below to quickly verify that the network renders correctly.

0.3 3.3 Extraction

BayesDown Extraction Code already part of ArgDown extraction code, therefore just use same function “parse_markdown_hierarchy(markdown_data)” and ignore the extra argument (“ArgDown”) because it is automatically set to false amd will by default extract BayesDown.

```
result_df = parse_markdown_hierarchy_fixed(md_content_ex_rain)
result_df
```

	Title	Description	line	line_numbe
0	Existential_Catastrophe	The destruction of humanity's long-term potent...	0	[0]
1	Human_Disempowerment	Permanent and collective disempowerment of hum...	1	[1]
2	Scale_Of_Power_Seeking	Power-seeking by AI systems scaling to the poi...	2	[2]
3	Misaligned_Power_Seeking	Deployed AI systems seeking power in unintende...	3	[3, 21, 23, 2
4	APS_Systems	AI systems with advanced capabilities, agentic...	4	[4]
5	Advanced_AI_Capability	AI systems that outperform humans on tasks tha...	5	[5]
6	Agentic_Planning	AI systems making and executing plans based on...	6	[6]
7	Strategic_Awareness	AI systems with models accurately representing...	7	[7]
8	Difficulty_Of_Alignment	It is harder to build aligned systems than mis...	8	[8]
9	Instrumental_Convergence	AI systems with misaligned objectives tend to ...	9	[9]
10	Problems_With_Proxies	Optimizing for proxy objectives breaks correla...	10	[10]
11	Problems_With_Search	Search processes can yield systems pursuing di...	11	[11]
12	Deployment_Decisions	Decisions to deploy potentially misaligned AI ...	12	[12]
13	Incentives_To_Build_APS	Strong incentives to build and deploy APS syst...	13	[13]
14	Usefulness_Of_APS	APS systems are very useful for many valuable ...	14	[14]
15	Competitive_Dynamics	Competitive pressures between AI developers.	15	[15]
16	Deception_By_AI	AI systems deceiving humans about their true o...	16	[16]
17	Corrective_Feedback	Human society implementing corrections after o...	17	[17]
18	Warning_Shots	Observable failures in weaker systems before c...	18	[18]
19	Rapid_Capability_Escalation	AI capabilities escalating very rapidly, allow...	19	[19]
20	Barriers_To_Understanding	Difficulty in understanding the internal worki...	20	[20]
21	Adversarial_Dynamics	Potentially adversarial relationships between ...	22	[22]
22	Stakes_Of_Error	The escalating impact of mistakes with power-s...	24	[24]

0.3.1 3.3 Data-Post-Processing

Add rows to data frame that can be calculated from the extracted rows

```
# @title 3.3.1 Data Post-Processing Functions ---

"""
BLOCK PURPOSE: Enhances the extracted BayesDown data with calculated metrics and network pro

This block provides functions to enrich the basic extracted data with additional
calculated columns that are useful for analysis and visualization:

1. Joint probabilities - Calculating P(A,B) from conditional and prior probabilities
2. Network metrics - Centrality measures that indicate importance of nodes in the network
3. Markov blanket - Identifying the minimal set of nodes that shield a node from the rest

These enhancements provide valuable context for understanding the network structure
and the relationships between variables, enabling more advanced analysis and
```

```
improving visualization.

DEPENDENCIES: networkx for graph calculations
INPUTS: DataFrame with basic extracted BayesDown data
OUTPUTS: Enhanced DataFrame with additional calculated columns
"""

def enhance_extracted_data(df):
    """
    Enhance the extracted data with calculated columns

    Args:
        df: DataFrame with extracted BayesDown data

    Returns:
        Enhanced DataFrame with additional columns
    """
    # Create a copy to avoid modifying the original
    enhanced_df = df.copy()

    # 1. Calculate joint probabilities - P(A,B) = P(A|B) * P(B)
    enhanced_df['joint_probabilities'] = None

    for idx, row in enhanced_df.iterrows():
        title = row['Title']
        priors = row['priors'] if isinstance(row['priors'], dict) else {}
        posteriors = row['posteriors'] if isinstance(row['posteriors'], dict) else {}
        parents = row['Parents'] if isinstance(row['Parents'], list) else []

        # Skip if no parents or no priors
        if not parents or not priors:
            continue

        # Initialize joint probabilities dictionary
        joint_probs = {}

        # Get instantiations
        instantiations = row['instantiations']
        if not isinstance(instantiations, list) or not instantiations:
            continue

        # For each parent and child instantiation combination, calculate joint probability
```

```
        for inst in instantiations:
            # Get this instantiation's prior probability
            inst_prior_key = f"p({inst})"
            if inst_prior_key not in priors:
                continue

            try:
                inst_prior = float(priors[inst_prior_key])
            except (ValueError, TypeError):
                continue

            # For each parent
            for parent in parents:
                parent_row = enhanced_df[enhanced_df['Title'] == parent]
                if parent_row.empty:
                    continue

                parent_insts = parent_row.iloc[0]['instantiations']
                if not isinstance(parent_insts, list) or not parent_insts:
                    continue

                for parent_inst in parent_insts:
                    # Get conditional probability
                    cond_key = f"p({inst}|{parent}={parent_inst})"
                    if cond_key in posteriors:
                        try:
                            cond_prob = float(posteriors[cond_key])

                            # Get parent's prior
                            parent_priors = parent_row.iloc[0]['priors']
                            if not isinstance(parent_priors, dict):
                                continue

                            parent_prior_key = f"p({parent_inst})"
                            if parent_prior_key not in parent_priors:
                                continue

                            try:
                                parent_prior = float(parent_priors[parent_prior_key])

                                # Calculate joint probability: P(A,B) = P(A|B) * P(B)
                                joint_prob = cond_prob * parent_prior
```

```
                                joint_key = f"p({inst},{parent}={parent_inst})"
                                joint_probs[joint_key] = str(round(joint_prob, 4))
                            except (ValueError, TypeError):
                                joint_prob = cond_prob * parent_prior
                                joint_key = f"p({inst},{parent}={parent_inst})"
                                joint_probs[joint_key] = str(round(joint_prob, 4))
                            except (ValueError, TypeError):
                                continue
                        except (ValueError, TypeError):
                            continue

        # Store joint probabilities in dataframe
        enhanced_df.at[idx, 'joint_probabilities'] = joint_probs

    # 2. Calculate network metrics
    # Create a directed graph
    import networkx as nx
    G = nx.DiGraph()

    # Add nodes
    for idx, row in enhanced_df.iterrows():
        G.add_node(row['Title'])

    # Add edges
    for idx, row in enhanced_df.iterrows():
        child = row['Title']
        parents = row['Parents'] if isinstance(row['Parents'], list) else []

        for parent in parents:
            if parent in G.nodes():
                G.add_edge(parent, child)

    # Calculate centrality measures
    degree_centrality = nx.degree_centrality(G)  # Overall connectedness
    in_degree_centrality = nx.in_degree_centrality(G)  # How many nodes affect this one
    out_degree_centrality = nx.out_degree_centrality(G)  # How many nodes this one affects

    try:
        betweenness_centrality = nx.betweenness_centrality(G)  # Node's role as a connector
    except:
        betweenness_centrality = {node: 0 for node in G.nodes()}
```

```
    # Add metrics to dataframe
    enhanced_df['degree_centrality'] = None
    enhanced_df['in_degree_centrality'] = None
    enhanced_df['out_degree_centrality'] = None
    enhanced_df['betweenness_centrality'] = None

    for idx, row in enhanced_df.iterrows():
        title = row['Title']
        enhanced_df.at[idx, 'degree_centrality'] = degree_centrality.get(title, 0)
        enhanced_df.at[idx, 'in_degree_centrality'] = in_degree_centrality.get(title, 0)
        enhanced_df.at[idx, 'out_degree_centrality'] = out_degree_centrality.get(title, 0)
        enhanced_df.at[idx, 'betweenness_centrality'] = betweenness_centrality.get(title, 0)

    # 3. Add Markov blanket information (parents, children, and children's parents)
    enhanced_df['markov_blanket'] = None

    for idx, row in enhanced_df.iterrows():
        title = row['Title']
        parents = row['Parents'] if isinstance(row['Parents'], list) else []
        children = row['Children'] if isinstance(row['Children'], list) else []

        # Get children's parents (excluding this node)
        childrens_parents = []
        for child in children:
            child_row = enhanced_df[enhanced_df['Title'] == child]
            if not child_row.empty:
                child_parents = child_row.iloc[0]['Parents']
                if isinstance(child_parents, list):
                    childrens_parents.extend([p for p in child_parents if p != title])

        # Remove duplicates
        childrens_parents = list(set(childrens_parents))

        # Combine to get Markov blanket
        markov_blanket = list(set(parents + children + childrens_parents))
        enhanced_df.at[idx, 'markov_blanket'] = markov_blanket

    return enhanced_df
```

```
# @title 3.3 --- Enhance Extracted Data with Network Metrics ---

"""
```

```
BLOCK PURPOSE: Applies the post-processing functions to enhance the extracted data.

This block takes the basic extracted DataFrame from the BayesDown parsing step
and enriches it with calculated metrics that provide deeper insight into the
network structure and relationships. It:

1. Applies the enhancement functions defined previously
2. Displays summary information about key calculated metrics
3. Saves the enhanced data for further analysis and visualization

The enhanced DataFrame provides a richer representation of the Bayesian network,
including measures of node importance and conditional relationships that are
essential for effective analysis and visualization.

DEPENDENCIES: enhance_extracted_data function
INPUTS: DataFrame with basic extracted BayesDown data
OUTPUTS: Enhanced DataFrame with additional calculated columns, saved to CSV
"""

# Enhance the extracted dataframe with calculated columns
enhanced_df = enhance_extracted_data(result_df)

# Display the enhanced dataframe
print("Enhanced DataFrame with additional calculated columns:")
enhanced_df.head()

# Check some calculated metrics
print("\nJoint Probabilities Example:")
example_node = enhanced_df.loc[0, 'Title']
joint_probs = enhanced_df.loc[0, 'joint_probabilities']
print(f"Joint probabilities for {example_node}:")
print(joint_probs)

print("\nNetwork Metrics:")
for idx, row in enhanced_df.iterrows():
    print(f"{row['Title']}:")
    print(f"  Degree Centrality: {row['degree_centrality']:.3f}")
    print(f"  Betweenness Centrality: {row['betweenness_centrality']:.3f}")

# Save the enhanced dataframe
enhanced_df.to_csv('enhanced_extracted_data.csv', index=False)
print("\nEnhanced data saved to 'enhanced_extracted_data.csv'")
```

```
Enhanced DataFrame with additional calculated columns:

Joint Probabilities Example:
Joint probabilities for Existential_Catastrophe:
None

Network Metrics:
Existential_Catastrophe:
  Degree Centrality: 0.000
  Betweenness Centrality: 0.000
Human_Disempowerment:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Scale_Of_Power_Seeking:
  Degree Centrality: 0.136
  Betweenness Centrality: 0.037
Misaligned_Power_Seeking:
  Degree Centrality: 0.182
  Betweenness Centrality: 0.056
APS_Systems:
  Degree Centrality: 0.182
  Betweenness Centrality: 0.019
Advanced_AI_Capability:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Agentic_Planning:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Strategic_Awareness:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Difficulty_Of_Alignment:
  Degree Centrality: 0.182
  Betweenness Centrality: 0.019
Instrumental_Convergence:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Problems_With_Proxies:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Problems_With_Search:
  Degree Centrality: 0.045
```

```
  Betweenness Centrality: 0.000
Deployment_Decisions:
  Degree Centrality: 0.136
  Betweenness Centrality: 0.026
Incentives_To_Build_APS:
  Degree Centrality: 0.136
  Betweenness Centrality: 0.017
Usefulness_Of_APS:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Competitive_Dynamics:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Deception_By_AI:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Corrective_Feedback:
  Degree Centrality: 0.136
  Betweenness Centrality: 0.009
Warning_Shots:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Rapid_Capability_Escalation:
  Degree Centrality: 0.045
  Betweenness Centrality: 0.000
Barriers_To_Understanding:
  Degree Centrality: 0.000
  Betweenness Centrality: 0.000
Adversarial_Dynamics:
  Degree Centrality: 0.000
  Betweenness Centrality: 0.000
Stakes_Of_Error:
  Degree Centrality: 0.000
  Betweenness Centrality: 0.000

Enhanced data saved to 'enhanced_extracted_data.csv'
```

0.3.2 3.4 Download and save finished data frame as .csv file

```
# @title 3.4 --- Save Extracted Data for Further Processing ---

"""
BLOCK PURPOSE: Saves the extracted data to a CSV file for further processing.
```

```
This step is essential for:
1. Persisting the structured representation of the Bayesian network
2. Enabling further analysis in other tools or notebook sections
3. Creating a permanent record of the extraction results
4. Making the data available for the visualization pipeline

The CSV format provides a standardized, tabular representation of the network
that can be easily loaded and processed in subsequent analysis steps.

DEPENDENCIES: pandas DataFrame operations
INPUTS: Extracted DataFrame from the parsing step
OUTPUTS: CSV file containing the structured network data
"""

# Save the extracted data as a CSV file
result_df.to_csv('extracted_data.csv', index=False)

print("  Extracted data saved successfully to 'extracted_data.csv'")
print("Note: If using updated data in future steps, the file must be pushed to the GitHub re
```

```
  Extracted data saved successfully to 'extracted_data.csv'
Note: If using updated data in future steps, the file must be pushed to the GitHub repositor
```

4. 4.0 Analysis & Inference: Bayesian Network Visualization

1.1 Bayesian Network Visualization Approach

This section implements the visualization component of the AMTAIR project, transforming the structured data extracted from BayesDown into an interactive network visualization that makes complex probabilistic relationships accessible to human understanding.

1.1.1 Visualization Philosophy

A key challenge in AI governance is making complex probabilistic relationships understandable to diverse stakeholders. This visualization system addresses this challenge through:

1. **Visual Encoding of Probability**: Node colors reflect probability values (green for high probability, red for low)
2. **Structural Classification**: Border colors indicate node types (blue for root causes, purple for intermediate nodes, magenta for leaf nodes)
3. **Progressive Disclosure**: Basic information in tooltips, detailed probability tables in modal popups
4. **Interactive Exploration**: Draggable nodes, configurable physics, click interactions

1.1.2 Connection to AMTAIR Goals

This visualization approach directly supports the AMTAIR project's goal of improving coordination in AI governance by:

1. Making implicit models explicit through visual representation
2. Providing a common language for discussing probabilistic relationships
3. Enabling non-technical stakeholders to engage with formal models
4. Creating shareable artifacts that facilitate collaboration

1.1.3 Implementation Structure

The visualization system is implemented in four phases:

1. **Network Construction**: Creating a directed graph representation using NetworkX
2. **Node Classification**: Identifying node types based on network position
3. **Visual Enhancement**: Adding color coding, tooltips, and interactive elements
4. **Interactive Features**: Implementing click handling for detailed exploration

The resulting visualization serves as both an analytical tool for experts and a communication tool for broader audiences, bridging the gap between technical and policy domains in AI governance discussions.

1.2 Phase 1: Dependencies/Functions

```
# @title 4.0 --- Bayesian Network Visualization Functions ---

"""
BLOCK PURPOSE: Provides functions to create interactive Bayesian network visualizations
from DataFrame representations of ArgDown/BayesDown data.

This block implements the visualization pipeline described in the AMTAIR project, transformi
the structured DataFrame extracted from ArgDown/BayesDown into an interactive network graph
that displays nodes, relationships, and probability information. The visualization leverages
NetworkX for graph representation and PyVis for interactive display.

Key visualization features:
1. Color-coding of nodes based on probability values
2. Border styling to indicate node types (root, intermediate, leaf)
3. Interactive tooltips with probability information
4. Modal popups with detailed conditional probability tables
5. Physics-based layout for intuitive exploration

DEPENDENCIES: networkx, pyvis, HTML display from IPython
INPUTS: DataFrame with node information, relationships, and probabilities
OUTPUTS: Interactive HTML visualization of the Bayesian network
"""

from pyvis.network import Network
import networkx as nx
from IPython.display import HTML
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
import io
import base64
import colorsys
import json

def create_bayesian_network_with_probabilities(df):
    """
    Create an interactive Bayesian network visualization with enhanced probability visualiza
    and node classification based on network structure.

    Args:
        df (pandas.DataFrame): DataFrame containing node information, relationships, and pro

    Returns:
        IPython.display.HTML: Interactive HTML visualization of the Bayesian network
    """
    # PHASE 1: Create a directed graph representation
    G = nx.DiGraph()

    # Add nodes with proper attributes
    for idx, row in df.iterrows():
        title = row['Title']
        description = row['Description']

        # Process probability information
        priors = get_priors(row)
        instantiations = get_instantiations(row)

        # Add node with base information
        G.add_node(
            title,
            description=description,
            priors=priors,
            instantiations=instantiations,
            posteriors=get_posteriors(row)
        )

    # Add edges based on parent-child relationships
    for idx, row in df.iterrows():
        child = row['Title']
        parents = get_parents(row)
```

```
        # Add edges from each parent to this child
        for parent in parents:
            if parent in G.nodes():
                G.add_edge(parent, child)

    # PHASE 2: Classify nodes based on network structure
    classify_nodes(G)

    # PHASE 3: Create interactive network visualization
    net = Network(notebook=True, directed=True, cdn_resources="in_line", height="600px", wid

    # Configure physics for better layout
    net.force_atlas_2based(gravity=-50, spring_length=100, spring_strength=0.02)
    net.show_buttons(filter_=['physics'])  # Allow user to adjust physics settings

    # Add the graph to the network
    net.from_nx(G)

    # PHASE 4: Enhance node appearance with probability information
    for node in net.nodes:
        node_id = node['id']
        node_data = G.nodes[node_id]

        # Get node type and set border color
        node_type = node_data.get('node_type', 'unknown')
        border_color = get_border_color(node_type)

        # Get probability information
        priors = node_data.get('priors', {})
        true_prob = priors.get('true_prob', 0.5) if priors else 0.5

        # Get proper state names
        instantiations = node_data.get('instantiations', ["TRUE", "FALSE"])
        true_state = instantiations[0] if len(instantiations) > 0 else "TRUE"
        false_state = instantiations[1] if len(instantiations) > 1 else "FALSE"

        # Create background color based on probability
        background_color = get_probability_color(priors)

        # Create tooltip with probability information
        tooltip = create_tooltip(node_id, node_data)
```

```
        # Create a simpler node label with probability
        simple_label = f"{node_id}\np={true_prob:.2f}"

        # Store expanded content as a node attribute for use in click handler
        node_data['expanded_content'] = create_expanded_content(node_id, node_data)

        # Set node attributes
        node['title'] = tooltip  # Tooltip HTML
        node['label'] = simple_label  # Simple text label
        node['shape'] = 'box'
        node['color'] = {
            'background': background_color,
            'border': border_color,
            'highlight': {
                'background': background_color,
                'border': border_color
            }
        }

    # PHASE 5: Setup interactive click handling
    # Prepare data for click handler
    setup_data = {
        'nodes_data': {node_id: {
            'expanded_content': json.dumps(G.nodes[node_id].get('expanded_content', '')),
            'description': G.nodes[node_id].get('description', ''),
            'priors': G.nodes[node_id].get('priors', {}),
            'posteriors': G.nodes[node_id].get('posteriors', {})
        } for node_id in G.nodes()}
    }

    # JavaScript code for handling node clicks
    click_js = """
    // Store node data for click handling
    var nodesData = %s;

    // Add event listener for node clicks
    network.on("click", function(params) {
        if (params.nodes.length > 0) {
            var nodeId = params.nodes[0];
            var nodeInfo = nodesData[nodeId];

            if (nodeInfo) {
```

```
                // Create a modal popup for expanded content
                var modal = document.createElement('div');
                modal.style.position = 'fixed';
                modal.style.left = '50%%';
                modal.style.top = '50%%';
                modal.style.transform = 'translate(-50%%, -50%%)';
                modal.style.backgroundColor = 'white';
                modal.style.padding = '20px';
                modal.style.borderRadius = '5px';
                modal.style.boxShadow = '0 0 10px rgba(0,0,0,0.5)';
                modal.style.zIndex = '1000';
                modal.style.maxWidth = '80%%';
                modal.style.maxHeight = '80%%';
                modal.style.overflow = 'auto';

                // Add expanded content
                modal.innerHTML = nodeInfo.expanded_content || 'No detailed information avai

                // Add close button
                var closeBtn = document.createElement('button');
                closeBtn.innerHTML = 'Close';
                closeBtn.style.marginTop = '10px';
                closeBtn.style.padding = '5px 10px';
                closeBtn.style.cursor = 'pointer';
                closeBtn.onclick = function() {
                    document.body.removeChild(modal);
                };
                modal.appendChild(closeBtn);

                // Add modal to body
                document.body.appendChild(modal);
            }
        }
    });
    """ % json.dumps(setup_data['nodes_data'])

    # PHASE 6: Save the graph to HTML and inject custom click handling
    html_file = "bayesian_network.html"
    net.save_graph(html_file)

    # Inject custom click handling into HTML
    try:
```

```
        with open(html_file, "r") as f:
            html_content = f.read()

        # Insert click handling script before the closing body tag
        html_content = html_content.replace('</body>', f'<script>{click_js}</script></body>'

        # Write back the modified HTML
        with open(html_file, "w") as f:
            f.write(html_content)

        return HTML(html_content)
    except Exception as e:
        return HTML(f"<p>Error rendering HTML: {str(e)}</p><p>The network visualization has
```

1.3 Phase 2: Node Classification and Styling Module

```
# @title 4.1 --- Node Classification and Styling Functions ---

"""
BLOCK PURPOSE: Implements the visual classification and styling of nodes in the Bayesian net

This module handles the identification of node types based on their position in the network
and provides appropriate visual styling for each type. The functions:

1. Classify nodes as parents (causes), children (intermediate effects), or leaves (final eff
2. Assign appropriate border colors to visually distinguish node types
3. Calculate background colors based on probability values
4. Extract relevant information from DataFrame rows in a robust manner

The visual encoding helps users understand both the structure of the network
and the probability distributions at a glance.

DEPENDENCIES: colorsys for color manipulation
INPUTS: Graph structure and node data
OUTPUTS: Classification and styling information for visualization
"""

def classify_nodes(G):
    """
    Classify nodes as parent, child, or leaf based on network structure
```

```
    Args:
        G (networkx.DiGraph): Directed graph representation of the Bayesian network

    Effects:
        Adds 'node_type' attribute to each node in the graph:
        - 'parent': Root node with no parents but has children (causal source)
        - 'child': Node with both parents and children (intermediate)
        - 'leaf': Node with parents but no children (final effect)
        - 'isolated': Node with no connections (rare in Bayesian networks)
    """
    for node in G.nodes():
        predecessors = list(G.predecessors(node))  # Nodes pointing to this one (causes)
        successors = list(G.successors(node))      # Nodes this one points to (effects)

        if not predecessors:  # No parents
            if successors:  # Has children
                G.nodes[node]['node_type'] = 'parent'  # Root cause
            else:  # No children either
                G.nodes[node]['node_type'] = 'isolated'  # Disconnected node
        else:  # Has parents
            if not successors:  # No children
                G.nodes[node]['node_type'] = 'leaf'  # Final effect
            else:  # Has both parents and children
                G.nodes[node]['node_type'] = 'child'  # Intermediate node

def get_border_color(node_type):
    """
    Return border color based on node type

    Args:
        node_type (str): Type of node ('parent', 'child', 'leaf', or 'isolated')

    Returns:
        str: Hex color code for node border
    """
    if node_type == 'parent':
        return '#0000FF'  # Blue for root causes
    elif node_type == 'child':
        return '#800080'  # Purple for intermediate nodes
    elif node_type == 'leaf':
        return '#FF00FF'  # Magenta for final effects
    else:
```

```
        return '#000000'  # Default black for any other type

def get_probability_color(priors):
    """
    Create background color based on probability (red to green gradient)

    Args:
        priors (dict): Dictionary containing probability information

    Returns:
        str: Hex color code for node background, ranging from red (low probability)
             to green (high probability)
    """
    # Default to neutral color if no probability
    if not priors or 'true_prob' not in priors:
        return '#F8F8F8'  # Light grey

    # Get probability value
    prob = priors['true_prob']

    # Create color gradient from red (0.0) to green (1.0)
    hue = 120 * prob  # 0 = red, 120 = green (in HSL color space)
    saturation = 0.75
    lightness = 0.8  # Lighter color for better text visibility

    # Convert HSL to RGB
    r, g, b = colorsys.hls_to_rgb(hue/360, lightness, saturation)

    # Convert to hex format
    hex_color = "#{:02x}{:02x}{:02x}".format(int(r*255), int(g*255), int(b*255))

    return hex_color

def get_parents(row):
    """
    Extract parent nodes from row data, with safe handling for different data types

    Args:
        row (pandas.Series): Row from DataFrame containing node information

    Returns:
        list: List of parent node names
```

```
    """
    if 'Parents' not in row:
        return []

    parents_data = row['Parents']

    # Handle NaN, None, or empty list
    if isinstance(parents_data, float) and pd.isna(parents_data):
        return []

    if parents_data is None:
        return []

    # Handle different data types
    if isinstance(parents_data, list):
        # Return a list with NaN and empty strings removed
        return [p for p in parents_data if not (isinstance(p, float) and pd.isna(p)) and p !

    if isinstance(parents_data, str):
        if not parents_data.strip():
            return []

        # Remove brackets and split by comma, removing empty strings and NaN
        cleaned = parents_data.strip('[]"\'')
        if not cleaned:
            return []

        return [p.strip(' "\'') for p in cleaned.split(',') if p.strip()]

    # Default: empty list
    return []

def get_instantiations(row):
    """
    Extract instantiations with safe handling for different data types

    Args:
        row (pandas.Series): Row from DataFrame containing node information

    Returns:
        list: List of possible instantiations (states) for the node
    """
```

```
    if 'instantiations' not in row:
        return ["TRUE", "FALSE"]

    inst_data = row['instantiations']

    # Handle NaN or None
    if isinstance(inst_data, float) and pd.isna(inst_data):
        return ["TRUE", "FALSE"]

    if inst_data is None:
        return ["TRUE", "FALSE"]

    # Handle different data types
    if isinstance(inst_data, list):
        return inst_data if inst_data else ["TRUE", "FALSE"]

    if isinstance(inst_data, str):
        if not inst_data.strip():
            return ["TRUE", "FALSE"]

        # Remove brackets and split by comma
        cleaned = inst_data.strip('[]"\'')
        if not cleaned:
            return ["TRUE", "FALSE"]

        return [i.strip(' "\'') for i in cleaned.split(',') if i.strip()]

    # Default
    return ["TRUE", "FALSE"]

def get_priors(row):
    """
    Extract prior probabilities with safe handling for different data types

    Args:
        row (pandas.Series): Row from DataFrame containing node information

    Returns:
        dict: Dictionary of prior probabilities with 'true_prob' added for convenience
    """
    if 'priors' not in row:
        return {}
```

```
    priors_data = row['priors']

    # Handle NaN or None
    if isinstance(priors_data, float) and pd.isna(priors_data):
        return {}

    if priors_data is None:
        return {}

    result = {}

    # Handle dictionary
    if isinstance(priors_data, dict):
        result = priors_data
    # Handle string representation of dictionary
    elif isinstance(priors_data, str):
        if not priors_data.strip() or priors_data == '{}':
            return {}

        try:
            # Try to evaluate as Python literal
            import ast
            result = ast.literal_eval(priors_data)
        except:
            # Simple parsing for items like {'p(TRUE)': '0.2', 'p(FALSE)': '0.8'}
            if '{' in priors_data and '}' in priors_data:
                content = priors_data[priors_data.find('{')+1:priors_data.rfind('}')]
                items = [item.strip() for item in content.split(',')]

                for item in items:
                    if ':' in item:
                        key, value = item.split(':', 1)
                        key = key.strip(' \'\"')
                        value = value.strip(' \'\"')
                        result[key] = value

    # Extract main probability for TRUE state
    instantiations = get_instantiations(row)
    true_state = instantiations[0] if instantiations else "TRUE"
    true_key = f"p({true_state})"
```

```
    if true_key in result:
        try:
            result['true_prob'] = float(result[true_key])
        except:
            pass

    return result

def get_posteriors(row):
    """
    Extract posterior probabilities with safe handling for different data types

    Args:
        row (pandas.Series): Row from DataFrame containing node information

    Returns:
        dict: Dictionary of conditional probabilities
    """
    if 'posteriors' not in row:
        return {}

    posteriors_data = row['posteriors']

    # Handle NaN or None
    if isinstance(posteriors_data, float) and pd.isna(posteriors_data):
        return {}

    if posteriors_data is None:
        return {}

    result = {}

    # Handle dictionary
    if isinstance(posteriors_data, dict):
        result = posteriors_data
    # Handle string representation of dictionary
    elif isinstance(posteriors_data, str):
        if not posteriors_data.strip() or posteriors_data == '{}':
            return {}

        try:
            # Try to evaluate as Python literal
```

```
            import ast
            result = ast.literal_eval(posteriors_data)
        except:
            # Simple parsing
            if '{' in posteriors_data and '}' in posteriors_data:
                content = posteriors_data[posteriors_data.find('{')+1:posteriors_data.rfind(
                items = [item.strip() for item in content.split(',')]

                for item in items:
                    if ':' in item:
                        key, value = item.split(':', 1)
                        key = key.strip(' \'\"')
                        value = value.strip(' \'\"')
                        result[key] = value

    return result
```

1.4 Phase 3: HTML Content Generation Module

```
# @title 4.2 --- HTML Content Generation Functions ---

"""
BLOCK PURPOSE: Creates rich HTML content for the interactive Bayesian network visualization.

This module generates the HTML components that enhance the Bayesian network visualization:
1. Probability bars - Visual representation of probability distributions
2. Node tooltips - Rich information displayed on hover
3. Expanded content - Detailed probability information shown when clicking nodes

These HTML components make the mathematical concepts of Bayesian networks more
intuitive and accessible to users without requiring deep statistical knowledge.
The visual encoding of probabilities (colors, bars) and the progressive disclosure
of information (hover, click) help users build understanding at their own pace.

DEPENDENCIES: HTML generation capabilities
INPUTS: Node data from the Bayesian network
OUTPUTS: HTML content for visualization components
"""

def create_probability_bar(true_prob, false_prob, height="15px", show_values=True, value_pre
    """
    Creates a reusable HTML component to visualize probability distribution
```

```
    Args:
        true_prob (float): Probability of the true state (0.0-1.0)
        false_prob (float): Probability of the false state (0.0-1.0)
        height (str): CSS height of the bar
        show_values (bool): Whether to display numerical values
        value_prefix (str): Prefix to add before values (e.g., "p=")

    Returns:
        str: HTML for a horizontal bar showing probabilities
    """
    # Prepare display labels if showing values
    true_label = f"{value_prefix}{true_prob:.3f}" if show_values else ""
    false_label = f"{value_prefix}{false_prob:.3f}" if show_values else ""

    # Create the HTML for a horizontal stacked bar
    html = f"""
    <div style="width:100%; height:{height}; display:flex; border:1px solid #ccc; overflow:h
        <div style="flex-basis:{true_prob*100}%; background:linear-gradient(to bottom, rgba(
            <span style="font-size:10px; color:white; text-shadow:0px 0px 2px #000;">{true_l
        </div>
        <div style="flex-basis:{false_prob*100}%; background:linear-gradient(to bottom, rgba
            <span style="font-size:10px; color:white; text-shadow:0px 0px 2px #000;">{false_
        </div>
    </div>
    """
    return html

def create_tooltip(node_id, node_data):
    """
    Create rich HTML tooltip with probability information

    Args:
        node_id (str): Identifier of the node
        node_data (dict): Node attributes including probabilities

    Returns:
        str: HTML content for tooltip displayed on hover
    """
    # Extract node information
    description = node_data.get('description', '')
    priors = node_data.get('priors', {})
```

```
    instantiations = node_data.get('instantiations', ["TRUE", "FALSE"])

    # Start building the HTML tooltip
    html = f"""
    <div style="max-width:350px; padding:10px; background-color:#f8f9fa; border-radius:5px;
        <h3 style="margin-top:0; color:#202124;">{node_id}</h3>
        <p style="font-style:italic;">{description}</p>
    """

    # Add prior probabilities section
    if priors and 'true_prob' in priors:
        true_prob = priors['true_prob']
        false_prob = 1.0 - true_prob

        # Get proper state names
        true_state = instantiations[0] if len(instantiations) > 0 else "TRUE"
        false_state = instantiations[1] if len(instantiations) > 1 else "FALSE"

        html += f"""
        <div style="margin-top:10px; background-color:#fff; padding:8px; border-radius:4px;
            <h4 style="margin-top:0; font-size:14px;">Prior Probabilities:</h4>
            <div style="display:flex; justify-content:space-between; margin-bottom:4px;">
                <div style="font-size:12px;">{true_state}: {true_prob:.3f}</div>
                <div style="font-size:12px;">{false_state}: {false_prob:.3f}</div>
            </div>
            {create_probability_bar(true_prob, false_prob, "20px", True)}
        </div>
        """

    # Add click instruction
    html += """
    <div style="margin-top:8px; font-size:12px; color:#666; text-align:center;">
        Click node to see full probability details
    </div>
    </div>
    """

    return html

def create_expanded_content(node_id, node_data):
    """
    Create expanded content shown when a node is clicked
```

```
    Args:
        node_id (str): Identifier of the node
        node_data (dict): Node attributes including probabilities

    Returns:
        str: HTML content for detailed view displayed on click
    """
    # Extract node information
    description = node_data.get('description', '')
    priors = node_data.get('priors', {})
    posteriors = node_data.get('posteriors', {})
    instantiations = node_data.get('instantiations', ["TRUE", "FALSE"])

    # Get proper state names
    true_state = instantiations[0] if len(instantiations) > 0 else "TRUE"
    false_state = instantiations[1] if len(instantiations) > 1 else "FALSE"

    # Extract probabilities
    true_prob = priors.get('true_prob', 0.5)
    false_prob = 1.0 - true_prob

    # Start building the expanded content
    html = f"""
    <div style="max-width:500px; padding:15px; font-family:Arial, sans-serif;">
        <h2 style="margin-top:0; color:#333;">{node_id}</h2>
        <p style="font-style:italic; margin-bottom:15px;">{description}</p>

        <div style="margin-bottom:20px; padding:12px; border:1px solid #ddd; background-colo
            <h3 style="margin-top:0; color:#333;">Prior Probabilities</h3>
            <div style="display:flex; justify-content:space-between; margin-bottom:5px;">
                <div><strong>{true_state}:</strong> {true_prob:.3f}</div>
                <div><strong>{false_state}:</strong> {false_prob:.3f}</div>
            </div>
            {create_probability_bar(true_prob, false_prob, "25px", True)}
        </div>
    """

    # Add conditional probability table if available
    if posteriors:
        html += """
        <div style="padding:12px; border:1px solid #ddd; background-color:#f9f9f9; border-ra
```

```
            <h3 style="margin-top:0; color:#333;">Conditional Probabilities</h3>
            <table style="width:100%; border-collapse:collapse; font-size:13px;">
                <tr style="background-color:#eee;">
                    <th style="padding:8px; text-align:left; border:1px solid #ddd;">Conditi
                    <th style="padding:8px; text-align:center; border:1px solid #ddd; width:
                    <th style="padding:8px; text-align:center; border:1px solid #ddd;">Visua
                </tr>
        """

        # Sort posteriors to group by similar conditions
        posterior_items = list(posteriors.items())
        posterior_items.sort(key=lambda x: x[0])

        # Add rows for conditional probabilities
        for key, value in posterior_items:
            try:
                # Try to parse probability value
                prob_value = float(value)
                inv_prob = 1.0 - prob_value

                # Add row with probability visualization
                html += f"""
                <tr>
                    <td style="padding:8px; border:1px solid #ddd;">{key}</td>
                    <td style="padding:8px; text-align:center; border:1px solid #ddd;">{prob
                    <td style="padding:8px; border:1px solid #ddd;">
                        {create_probability_bar(prob_value, inv_prob, "20px", False)}
                    </td>
                </tr>
                """
            except:
                # Fallback for non-numeric values
                html += f"""
                <tr>
                    <td style="padding:8px; border:1px solid #ddd;">{key}</td>
                    <td style="padding:8px; text-align:center; border:1px solid #ddd;" colsp
                </tr>
                """

        html += """
            </table>
        </div>
```

```
        """

    html += "</div>"

    return html
```

1.5 Phase 4: Main Visualization Function

```
def create_bayesian_network_with_probabilities(df):
    """
    Create an interactive Bayesian network visualization with enhanced probability visualiza
    and node classification based on network structure.
    """
    # Create a directed graph
    G = nx.DiGraph()

    # Add nodes with proper attributes
    for idx, row in df.iterrows():
        title = row['Title']
        description = row['Description']

        # Process probability information
        priors = get_priors(row)
        instantiations = get_instantiations(row)

        # Add node with base information
        G.add_node(
            title,
            description=description,
            priors=priors,
            instantiations=instantiations,
            posteriors=get_posteriors(row)
        )

    # Add edges
    for idx, row in df.iterrows():
        child = row['Title']
        parents = get_parents(row)

        # Add edges from each parent to this child
        for parent in parents:
            if parent in G.nodes():
```

```
            G.add_edge(parent, child)

    # Classify nodes based on network structure
    classify_nodes(G)

    # Create network visualization
    net = Network(notebook=True, directed=True, cdn_resources="in_line", height="600px", wid

    # Configure physics for better layout
    net.force_atlas_2based(gravity=-50, spring_length=100, spring_strength=0.02)
    net.show_buttons(filter_=['physics'])

    # Add the graph to the network
    net.from_nx(G)

    # Enhance node appearance with probability information and classification
    for node in net.nodes:
        node_id = node['id']
        node_data = G.nodes[node_id]

        # Get node type and set border color
        node_type = node_data.get('node_type', 'unknown')
        border_color = get_border_color(node_type)

        # Get probability information
        priors = node_data.get('priors', {})
        true_prob = priors.get('true_prob', 0.5) if priors else 0.5

        # Get proper state names
        instantiations = node_data.get('instantiations', ["TRUE", "FALSE"])
        true_state = instantiations[0] if len(instantiations) > 0 else "TRUE"
        false_state = instantiations[1] if len(instantiations) > 1 else "FALSE"

        # Create background color based on probability
        background_color = get_probability_color(priors)

        # Create tooltip with probability information
        tooltip = create_tooltip(node_id, node_data)

        # Create a simpler node label with probability
        simple_label = f"{node_id}\np={true_prob:.2f}"
```

```
        # Store expanded content as a node attribute for use in click handler
        node_data['expanded_content'] = create_expanded_content(node_id, node_data)

        # Set node attributes
        node['title'] = tooltip  # Tooltip HTML
        node['label'] = simple_label  # Simple text label
        node['shape'] = 'box'
        node['color'] = {
            'background': background_color,
            'border': border_color,
            'highlight': {
                'background': background_color,
                'border': border_color
            }
        }

    # Set up the click handler with proper data
    setup_data = {
        'nodes_data': {node_id: {
            'expanded_content': json.dumps(G.nodes[node_id].get('expanded_content', '')),
            'description': G.nodes[node_id].get('description', ''),
            'priors': G.nodes[node_id].get('priors', {}),
            'posteriors': G.nodes[node_id].get('posteriors', {})
        } for node_id in G.nodes()}
    }

    # Add custom click handling JavaScript
    click_js = """
    // Store node data for click handling
    var nodesData = %s;

    // Add event listener for node clicks
    network.on("click", function(params) {
        if (params.nodes.length > 0) {
            var nodeId = params.nodes[0];
            var nodeInfo = nodesData[nodeId];

            if (nodeInfo) {
                // Create a modal popup for expanded content
                var modal = document.createElement('div');
                modal.style.position = 'fixed';
                modal.style.left = '50%%';
```

```
                modal.style.top = '50%%';
                modal.style.transform = 'translate(-50%%, -50%%)';
                modal.style.backgroundColor = 'white';
                modal.style.padding = '20px';
                modal.style.borderRadius = '5px';
                modal.style.boxShadow = '0 0 10px rgba(0,0,0,0.5)';
                modal.style.zIndex = '1000';
                modal.style.maxWidth = '80%%';
                modal.style.maxHeight = '80%%';
                modal.style.overflow = 'auto';

                // Parse the JSON string back to HTML content
                try {
                    var expandedContent = JSON.parse(nodeInfo.expanded_content);
                    modal.innerHTML = expandedContent;
                } catch (e) {
                    modal.innerHTML = 'Error displaying content: ' + e.message;
                }

                // Add close button
                var closeBtn = document.createElement('button');
                closeBtn.innerHTML = 'Close';
                closeBtn.style.marginTop = '10px';
                closeBtn.style.padding = '5px 10px';
                closeBtn.style.cursor = 'pointer';
                closeBtn.onclick = function() {
                    document.body.removeChild(modal);
                };
                modal.appendChild(closeBtn);

                // Add modal to body
                document.body.appendChild(modal);
            }
        }
    });
    """ % json.dumps(setup_data['nodes_data'])

    # Save the graph to HTML
    html_file = "bayesian_network.html"
    net.save_graph(html_file)

    # Inject custom click handling into HTML
```

```
    try:
        with open(html_file, "r") as f:
            html_content = f.read()

        # Insert click handling script before the closing body tag
        html_content = html_content.replace('</body>', f'<script>{click_js}</script></body>'

        # Write back the modified HTML
        with open(html_file, "w") as f:
            f.write(html_content)

        return HTML(html_content)
    except Exception as e:
        return HTML(f"<p>Error rendering HTML: {str(e)}</p><p>The network visualization has
```

Quickly check HTML Outputs

```
create_bayesian_network_with_probabilities(result_df)
```

```
# Use the function to create and display the visualization

print(result_df)
```

```
                         Title  \
0       Existential_Catastrophe
1          Human_Disempowerment
2        Scale_Of_Power_Seeking
3      Misaligned_Power_Seeking
4                   APS_Systems
5        Advanced_AI_Capability
6              Agentic_Planning
7           Strategic_Awareness
8       Difficulty_Of_Alignment
9      Instrumental_Convergence
10        Problems_With_Proxies
11         Problems_With_Search
12         Deployment_Decisions
13      Incentives_To_Build_APS
14            Usefulness_Of_APS
15         Competitive_Dynamics
16              Deception_By_AI
17          Corrective_Feedback
18                Warning_Shots
19  Rapid_Capability_Escalation
20    Barriers_To_Understanding
21         Adversarial_Dynamics
22              Stakes_Of_Error
```

	Description	line	line_numbers	\
0	The destruction of humanity's long-term potent...	0	[0]	
1	Permanent and collective disempowerment of hum...	1	[1]	
2	Power-seeking by AI systems scaling to the poi...	2	[2]	
3	Deployed AI systems seeking power in unintende...	3	[3, 21, 23, 25]	
4	AI systems with advanced capabilities, agentic...	4	[4]	
5	AI systems that outperform humans on tasks tha...	5	[5]	
6	AI systems making and executing plans based on...	6	[6]	
7	AI systems with models accurately representing...	7	[7]	
8	It is harder to build aligned systems than mis...	8	[8]	
9	AI systems with misaligned objectives tend to ...	9	[9]	
10	Optimizing for proxy objectives breaks correla...	10	[10]	
11	Search processes can yield systems pursuing di...	11	[11]	
12	Decisions to deploy potentially misaligned AI ...	12	[12]	
13	Strong incentives to build and deploy APS syst...	13	[13]	
14	APS systems are very useful for many valuable ...	14	[14]	
15	Competitive pressures between AI developers.	15	[15]	
16	AI systems deceiving humans about their true o...	16	[16]	
17	Human society implementing corrections after o...	17	[17]	
18	Observable failures in weaker systems before c...	18	[18]	
19	AI capabilities escalating very rapidly, allow...	19	[19]	
20	Difficulty in understanding the internal worki...	20	[20]	
21	Potentially adversarial relationships between ...	22	[22]	
22	The escalating impact of mistakes with power-s...	24	[24]	

	indentation	indentation_levels	\
0	0	[0]	
1	0	[0]	
2	4	[4]	
3	8	[8, 0, 0, 0]	
4	12	[12]	
5	16	[16]	
6	16	[16]	
7	16	[16]	
8	12	[12]	
9	16	[16]	
10	16	[16]	
11	16	[16]	
12	12	[12]	
13	16	[16]	
14	20	[20]	
15	20	[20]	

```
16          16               [16]
17           8                [8]
18          12               [12]
19          12               [12]
20           0                [0]
21           0                [0]
22           0                [0]

                                                 Parents  \
0                                                     []
1                              [Scale_Of_Power_Seeking]
2        [Misaligned_Power_Seeking, Corrective_Feedback]
3   [APS_Systems, Difficulty_Of_Alignment, Deploym...
4   [Advanced_AI_Capability, Agentic_Planning, Str...
5                                                     []
6                                                     []
7                                                     []
8   [Instrumental_Convergence, Problems_With_Proxi...
9                                                     []
10                                                    []
11                                                    []
12           [Incentives_To_Build_APS, Deception_By_AI]
13            [Usefulness_Of_APS, Competitive_Dynamics]
14                                                    []
15                                                    []
16                                                    []
17         [Warning_Shots, Rapid_Capability_Escalation]
18                                                    []
19                                                    []
20                                                    []
21                                                    []
22                                                    []

                       Children  \
0                             []
1                             []
2          [Human_Disempowerment]
3        [Scale_Of_Power_Seeking]
4   [Misaligned_Power_Seeking]
5                  [APS_Systems]
6                  [APS_Systems]
7                  [APS_Systems]
```

```
8   [Misaligned_Power_Seeking]
9    [Difficulty_Of_Alignment]
10   [Difficulty_Of_Alignment]
11   [Difficulty_Of_Alignment]
12  [Misaligned_Power_Seeking]
13     [Deployment_Decisions]
14   [Incentives_To_Build_APS]
15   [Incentives_To_Build_APS]
16     [Deployment_Decisions]
17    [Scale_Of_Power_Seeking]
18       [Corrective_Feedback]
19       [Corrective_Feedback]
20                          []
21                          []
22                          []

                                       instantiations  \
0   [existential_catastrophe_TRUE, existential_cat...
1   [human_disempowerment_TRUE, human_disempowerme...
2   [scale_of_power_seeking_TRUE, scale_of_power_s...
3   [misaligned_power_seeking_TRUE, misaligned_pow...
4               [aps_systems_TRUE, aps_systems_FALSE]
5   [advanced_ai_capability_TRUE, advanced_ai_capa...
6     [agentic_planning_TRUE, agentic_planning_FALSE]
7   [strategic_awareness_TRUE, strategic_awareness...
8   [difficulty_of_alignment_TRUE, difficulty_of_a...
9   [instrumental_convergence_TRUE, instrumental_c...
10  [problems_with_proxies_TRUE, problems_with_pro...
11  [problems_with_search_TRUE, problems_with_sear...
12  [deployment_decisions_DEPLOY, deployment_decis...
13  [incentives_to_build_aps_STRONG, incentives_to...
14    [usefulness_of_aps_HIGH, usefulness_of_aps_LOW]
15  [competitive_dynamics_STRONG, competitive_dyna...
16     [deception_by_ai_TRUE, deception_by_ai_FALSE]
17  [corrective_feedback_EFFECTIVE, corrective_fee...
18  [warning_shots_OBSERVED, warning_shots_UNOBSER...
19  [rapid_capability_escalation_TRUE, rapid_capab...
20  [barriers_to_understanding_HIGH, barriers_to_u...
21  [adversarial_dynamics_TRUE, adversarial_dynami...
22       [stakes_of_error_HIGH, stakes_of_error_LOW]

                                                priors  \
```

```
0   {'p(existential_catastrophe_TRUE)': '0.05', 'p...
1   {'p(human_disempowerment_TRUE)': '0.208', 'p(h...
2   {'p(scale_of_power_seeking_TRUE)': '0.208', 'p...
3   {'p(misaligned_power_seeking_TRUE)': '0.338', ...
4   {'p(aps_systems_TRUE)': '0.65', 'p(aps_systems...
5   {'p(advanced_ai_capability_TRUE)': '0.80', 'p(...
6   {'p(agentic_planning_TRUE)': '0.85', 'p(agenti...
7   {'p(strategic_awareness_TRUE)': '0.75', 'p(str...
8   {'p(difficulty_of_alignment_TRUE)': '0.40', 'p...
9   {'p(instrumental_convergence_TRUE)': '0.75', '...
10  {'p(problems_with_proxies_TRUE)': '0.80', 'p(p...
11  {'p(problems_with_search_TRUE)': '0.70', 'p(pr...
12  {'p(deployment_decisions_DEPLOY)': '0.70', 'p(...
13  {'p(incentives_to_build_aps_STRONG)': '0.80', ...
14  {'p(usefulness_of_aps_HIGH)': '0.85', 'p(usefu...
15  {'p(competitive_dynamics_STRONG)': '0.75', 'p(...
16  {'p(deception_by_ai_TRUE)': '0.50', 'p(decepti...
17  {'p(corrective_feedback_EFFECTIVE)': '0.60', '...
18  {'p(warning_shots_OBSERVED)': '0.70', 'p(warni...
19  {'p(rapid_capability_escalation_TRUE)': '0.45'...
20  {'p(barriers_to_understanding_HIGH)': '0.70', ...
21  {'p(adversarial_dynamics_TRUE)': '0.60', 'p(ad...
22  {'p(stakes_of_error_HIGH)': '0.85', 'p(stakes_...

                                          posteriors  No_Parent  No_Children  \
0   {'p(existential_catastrophe_TRUE|human_disempo...        True         True
1   {'p(human_disempowerment_TRUE|scale_of_power_s...       False         True
2   {'p(scale_of_power_seeking_TRUE|misaligned_pow...       False        False
3   {'p(misaligned_power_seeking_TRUE|aps_systems_...       False        False
4   {'p(aps_systems_TRUE|advanced_ai_capability_TR...       False        False
5                                                  {}        True        False
6                                                  {}        True        False
7                                                  {}        True        False
8   {'p(difficulty_of_alignment_TRUE|instrumental_...       False        False
9                                                  {}        True        False
10                                                 {}        True        False
11                                                 {}        True        False
12  {'p(deployment_decisions_DEPLOY|incentives_to_...       False        False
13  {'p(incentives_to_build_aps_STRONG|usefulness_...       False        False
14                                                 {}        True        False
15                                                 {}        True        False
16                                                 {}        True        False
```

17	{'p(corrective_feedback_EFFECTIVE|warning_shot...	False	False
18	{}	True	False
19	{}	True	False
20	{'p(barriers_to_understanding_HIGH|misaligned_...	True	True
21	{'p(adversarial_dynamics_TRUE|misaligned_power...	True	True
22	{'p(stakes_of_error_HIGH|misaligned_power_seek...	True	True

	parent_instantiations
0	[]
1	[[scale_of_power_seeking_TRUE, scale_of_power_...
2	[[misaligned_power_seeking_TRUE, misaligned_po...
3	[[aps_systems_TRUE, aps_systems_FALSE], [diffi...
4	[[advanced_ai_capability_TRUE, advanced_ai_cap...
5	[]
6	[]
7	[]
8	[[instrumental_convergence_TRUE, instrumental_...
9	[]
10	[]
11	[]
12	[[incentives_to_build_aps_STRONG, incentives_t...
13	[[usefulness_of_aps_HIGH, usefulness_of_aps_LO...
14	[]
15	[]
16	[]
17	[[warning_shots_OBSERVED, warning_shots_UNOBSE...
18	[]
19	[]
20	[]
21	[]
22	[]

Conclusion: From Prototype to Production

3.1 Summary of Achievements

This notebook has successfully demonstrated the core AMTAIR extraction pipeline, transforming structured argument representations into interactive Bayesian network visualizations through the following steps:

1. **Environment Setup**: Established a reproducible environment with necessary libraries and data access
2. **Argument Extraction**: Processed structured ArgDown representations preserving the hierarchical relationships
3. **Probability Integration**: Enhanced arguments with probability information to create BayesDown
4. **Data Transformation**: Converted BayesDown into structured DataFrame representation
5. **Visualization & Analysis**: Created interactive Bayesian network visualizations with probability encoding

The rain-sprinkler-lawn example, though simple, demonstrates all the key components of the extraction pipeline that can be applied to more complex AI safety arguments.

3.2 Limitations and Future Work

While this prototype successfully demonstrates the core pipeline, several limitations and opportunities for future work remain:

1. **LLM Extraction**: The current implementation focuses on processing pre-formatted ArgDown rather than performing extraction directly from unstructured text. Future work will integrate LLM-powered extraction.

2. **Scalability**: The system has been tested on small examples; scaling to larger, more complex arguments will require additional optimization and handling of computational complexity.

3. **Policy Evaluation**: The current implementation focuses on representation and visualization; future work will add policy evaluation capabilities by implementing intervention modeling.

4. **Prediction Market Integration**: Future versions will integrate with forecasting platforms to incorporate live data into the models.

3.3 Connection to AMTAIR Project

This prototype represents just one component of the broader AMTAIR project described in the project documentation (see PY_AMTAIRDescription and PY_AMTAIR_SoftwareToolsNMilestones). The full project includes:

1. **AI Risk Pathway Analyzer (ARPA)**: The core extraction and visualization system demonstrated in this notebook
2. **Worldview Comparator**: Tools for comparing different perspectives on AI risk
3. **Policy Impact Evaluator**: Systems for evaluating intervention effects across scenarios
4. **Strategic Intervention Generator**: Tools for identifying robust governance strategies

Together, these components aim to address the coordination crisis in AI governance by providing computational tools that make implicit models explicit, identify cruxes of disagreement, and evaluate policy impacts across diverse worldviews.

By transforming unstructured text into formal, analyzable representations, the AMTAIR project helps bridge the gaps between technical researchers, policy specialists, and other stakeholders, enabling more effective coordination in addressing existential risks from advanced AI.

6.0 Save Outputs

6. Saving and Exporting Results

This section provides tools for saving the notebook results and visualizations in various formats:

1. **HTML Export**: Creates a self-contained HTML version of the notebook with all visualizations
2. **Markdown Export**: Generates documentation-friendly Markdown version of the notebook
3. **PDF Export**: Creates a PDF document for formal sharing (requires LaTeX installation)

These exports are essential for: - Sharing analysis results with colleagues and stakeholders - Including visualizations in presentations and reports - Creating documentation for the AMTAIR project - Preserving results for future reference

The different formats serve different purposes, from interactive exploration (HTML) to documentation (Markdown) to formal presentation (PDF).

Instruction:

Download the ipynb, which you want to convert, on your local computer. Run the code below to upload the ipynb.

The html version will be downloaded automatically on your local machine. Enjoy it!

```
# @title 6.0 --- Save Visualization and Notebook Outputs as .HTML---

"""
BLOCK PURPOSE: Provides tools for saving the notebook results in various formats.

This block offers functions to:
1. Convert the notebook to HTML for easy sharing and viewing
2. Convert the notebook to Markdown for documentation purposes
3. Save the visualization outputs for external use

These tools are essential for preserving the analysis results and making them
accessible outside the notebook environment, supporting knowledge transfer
and integration with other AMTAIR project components.
```

```
DEPENDENCIES: nbformat, nbconvert modules
INPUTS: Current notebook state
OUTPUTS: HTML, Markdown, or other format versions of the notebook
"""

import nbformat
from nbconvert import HTMLExporter
import os

# Repository URL variable for file access
repo_url = "https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype/main/data/ex
notebook_name = "AMTAIR_Prototype_example_carlsmith"  # Change when working with different e

# Download the notebook file
!wget {repo_url}{notebook_name}.ipynb -O {notebook_name}.ipynb

# Load the notebook
try:
  with open(f"{notebook_name}.ipynb") as f:
    nb = nbformat.read(f, as_version=4)
  print(f" Successfully loaded notebook: {notebook_name}.ipynb")
except FileNotFoundError:
  print(f" Error: File '{notebook_name}.ipynb' not found. Please check if it was downloaded

# Initialize the HTML exporter
exporter = HTMLExporter()

# Convert the notebook to HTML
try:
    (body, resources) = exporter.from_notebook_node(nb)

    # Save the HTML to a file
    with open(f"{notebook_name}IPYNB.html", "w") as f:
        f.write(body)
    print(f" Successfully saved HTML version to: {notebook_name}IPYNB.html")
except Exception as e:
    print(f" Error converting notebook to HTML: {str(e)}")
```

```
--2025-04-26 22:34:17--  https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443...
HTTP request sent, awaiting response... 200 OK
```

```
Length: 1120047 (1.1M) [text/plain]
Saving to: ‘AMTAIR_Prototype_example_rain-sprinkler-lawn.ipynb’

          AMTAIR_Pr   0%[                    ]       0  --.-KB/s               AMTAIR_Protot

2025-04-26 22:34:17 (16.4 MB/s) - ‘AMTAIR_Prototype_example_rain-sprinkler-lawn.ipynb’ saved

  Successfully loaded notebook: AMTAIR_Prototype_example_rain-sprinkler-lawn.ipynb
  Successfully saved HTML version to: AMTAIR_Prototype_example_rain-sprinkler-lawnIPYNB.html
```

5.1 Convert .ipynb Notebook to MarkDown

```
# @title --- Convert .ipynb Notebook to MarkDown ---

import nbformat
from nbconvert import MarkdownExporter
import os

# repo_url = "https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype/main/data/
notebook_name = "AMTAIR_Prototype_example_carlsmith"  #Change Notebook name and path when wo

# Download the notebook file
!wget {repo_url}{notebook_name}.ipynb -O {notebook_name}.ipynb  # Corrected line

# Load the notebook
# add error handling for file not found
try:
  with open(f"{notebook_name}.ipynb") as f:
    nb = nbformat.read(f, as_version=4)
except FileNotFoundError:
  print(f"Error: File '{notebook_name}.ipynb' not found. Please check if it was downloaded c


# Initialize the Markdown exporter
exporter = MarkdownExporter(exclude_output=True)  # Correct initialization

# Convert the notebook to Markdown
(body, resources) = exporter.from_notebook_node(nb)

# Save the Markdown to a file
with open(f"{notebook_name}IPYNB.md", "w") as f:
    f.write(body)
```

```
--2025-04-26 22:33:43--  https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443...
HTTP request sent, awaiting response... 200 OK
Length: 1120047 (1.1M) [text/plain]
Saving to: ‘AMTAIR_Prototype_example_rain-sprinkler-lawn.ipynb’

          AMTAIR_Pr   0%[                    ]       0  --.-KB/s               AMTAIR_Protot

2025-04-26 22:33:43 (18.1 MB/s) - ‘AMTAIR_Prototype_example_rain-sprinkler-lawn.ipynb’ saved
```

```
# @title 6.1 --- Convert Notebook to Markdown Documentation ---

"""
BLOCK PURPOSE: Converts the notebook to Markdown format for documentation purposes.

Markdown is a lightweight markup language that is widely used for documentation
and is easily readable in both plain text and rendered formats. This conversion:

1. Preserves the structure and content of the notebook
2. Creates a format suitable for inclusion in documentation systems
3. Excludes code outputs to focus on the process and methodology
4. Supports version control and collaboration on GitHub

The resulting Markdown file can be used in project documentation, GitHub wikis,
or as a standalone reference guide to the AMTAIR extraction pipeline.

DEPENDENCIES: nbformat, nbconvert.MarkdownExporter modules
INPUTS: Current notebook state
OUTPUTS: Markdown version of the notebook
"""

import nbformat
from nbconvert import MarkdownExporter
import os

# Repository URL variable for file access
# repo_url = "https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype/main/data/
notebook_name = "AMTAIR_Prototype_example_carlsmith"  # Change when working with different e

# Download the notebook file
!wget {repo_url}{notebook_name}.ipynb -O {notebook_name}.ipynb
```

```
# Load the notebook
try:
  with open(f"{notebook_name}.ipynb") as f:
    nb = nbformat.read(f, as_version=4)
  print(f" Successfully loaded notebook: {notebook_name}.ipynb")
except FileNotFoundError:
  print(f" Error: File '{notebook_name}.ipynb' not found. Please check if it was downloaded


# Initialize the Markdown exporter
exporter = MarkdownExporter(exclude_output=True)  # Exclude outputs for cleaner documentatio

# Convert the notebook to Markdown
try:
    (body, resources) = exporter.from_notebook_node(nb)

    # Save the Markdown to a file
    with open(f"{notebook_name}IPYNB.md", "w") as f:
        f.write(body)
    print(f" Successfully saved Markdown version to: {notebook_name}IPYNB.md")
except Exception as e:
    print(f" Error converting notebook to Markdown: {str(e)}")
```

```
--2025-04-26 22:31:45--  https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443...
HTTP request sent, awaiting response... 200 OK
Length: 1120047 (1.1M) [text/plain]
Saving to: ‘AMTAIR_Prototype_example_rain-sprinkler-lawn.ipynb’

          AMTAIR_Pr   0%[                    ]       0  --.-KB/s               AMTAIR_Prototy

2025-04-26 22:31:45 (18.0 MB/s) - ‘AMTAIR_Prototype_example_rain-sprinkler-lawn.ipynb’ saved

  Successfully loaded notebook: AMTAIR_Prototype_example_rain-sprinkler-lawn.ipynb
  Successfully saved Markdown version to: AMTAIR_Prototype_example_rain-sprinkler-lawnIPYNB.
```

```
import nbformat
from nbconvert import PDFExporter
import os
import subprocess
import re
```

```
def escape_latex_special_chars(text):
  """Escapes special LaTeX characters in a string."""
  latex_special_chars = ['&', '%', '#', '_', '{', '}', '~', '^', '\\']
  replacement_patterns = [
      (char, '\\' + char) for char in latex_special_chars
  ]

  # Escape reserved characters
  for original, replacement in replacement_patterns:
    text = text.replace(original, replacement) # This is the fix
  return text

# Function to check if a command is available
def is_command_available(command):
    try:
        subprocess.run([command], capture_output=True, check=True)
        return True
    except (subprocess.CalledProcessError, FileNotFoundError):
        return False

# Check if xelatex is installed, and install if necessary
if not is_command_available("xelatex"):
    print("Installing necessary TeX packages...")
    !apt-get install -y texlive-xetex texlive-fonts-recommended texlive-plain-generic
    print("TeX packages installed successfully.")
else:
    print("xelatex is already installed. Skipping installation.")

# repo_url = "https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype/main/data/
notebook_name = "AMTAIR_Prototype_example_carlsmith"  #Change Notebook name and path when wo

# Download the notebook file
!wget {repo_url}{notebook_name}.ipynb -O {notebook_name}.ipynb  # Corrected line

# Load the notebook
# add error handling for file not found
try:
  with open(f"{notebook_name}.ipynb") as f:
    nb = nbformat.read(f, as_version=4)
except FileNotFoundError:
  print(f"Error: File '{notebook_name}.ipynb' not found. Please check if it was downloaded c
```

```
# Initialize the PDF exporter
exporter = PDFExporter(exclude_output=True)  # Changed to PDFExporter

# Sanitize notebook cell titles to escape special LaTeX characters like '&'
for cell in nb.cells:
    if 'cell_type' in cell and cell['cell_type'] == 'markdown':
        if 'source' in cell and isinstance(cell['source'], str):
            # Replace '&' with '\protect&' in markdown cell titles AND CONTENT
            # Updated to use escape_latex_special_chars function
            cell['source'] = escape_latex_special_chars(cell['source'])
            # Additionally, escape special characters in headings
            cell['source'] = re.sub(r'(#+)\s*(.*)', lambda m: m.group(1) + ' ' + escape_late



# Convert the notebook to PDF
(body, resources) = exporter.from_notebook_node(nb)


# Save the PDF to a file
with open(f"{notebook_name}IPYNB.pdf", "wb") as f:  # Changed to 'wb' for binary writing
    f.write(body)
```

```
Installing necessary TeX packages...
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono
  fonts-texgyre fonts-urw-base35 libapache-pom-java libcommons-logging-java
  libcommons-parent-java libfontbox-java libgs9 libgs9-common libidn12
  libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1 libruby3.0
  libsynctex2 libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13
  lmodern poppler-data preview-latex-style rake ruby ruby-net-telnet
  ruby-rubygems ruby-webrick ruby-xmlrpc ruby3.0 rubygems-integration t1utils
  teckit tex-common tex-gyre texlive-base texlive-binaries texlive-latex-base
  texlive-latex-extra texlive-latex-recommended texlive-pictures tipa
  xfonts-encodings xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java
  poppler-utils ghostscript fonts-japanese-mincho | fonts-ipafont-mincho
```

```
  fonts-japanese-gothic | fonts-ipafont-gothic fonts-arphic-ukai
  fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper gv
  | postscript-viewer perl-tk xpdf | pdf-viewer xzdec
  texlive-fonts-recommended-doc texlive-latex-base-doc python3-pygments
  icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl
  texlive-latex-extra-doc texlive-latex-recommended-doc texlive-luatex
  texlive-pstricks dot2tex prerex texlive-pictures-doc vprerex
  default-jre-headless tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono
  fonts-texgyre fonts-urw-base35 libapache-pom-java libcommons-logging-java
  libcommons-parent-java libfontbox-java libgs9 libgs9-common libidn12
  libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1 libruby3.0
  libsynctex2 libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13
  lmodern poppler-data preview-latex-style rake ruby ruby-net-telnet
  ruby-rubygems ruby-webrick ruby-xmlrpc ruby3.0 rubygems-integration t1utils
  teckit tex-common tex-gyre texlive-base texlive-binaries
  texlive-fonts-recommended texlive-latex-base texlive-latex-extra
  texlive-latex-recommended texlive-pictures texlive-plain-generic
  texlive-xetex tipa xfonts-encodings xfonts-utils
0 upgraded, 53 newly installed, 0 to remove and 34 not upgraded.
Need to get 182 MB of archives.
After this operation, 571 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-droid-fallback all 1:6.0.1r16-
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1 [2,696 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 poppler-data all 0.4.11-1 [2,171 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-common all 6.17 [33.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-urw-base35 all 20200910-1 [6,3
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9-common all 9.55.0~dfs
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libidn12 amd64 1.38-4ubuntu1
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libijs-0.35 amd64 0.35-15build2 [16.
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjbig2dec0 amd64 0.19-3build2 [64.
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9 amd64 9.55.0~dfsg1-0
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libkpathsea6 amd64 2021.202
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libwoff1 amd64 1.0.2-1build4 [45.2
Get:13 http://archive.ubuntu.com/ubuntu jammy/universe amd64 dvisvgm amd64 2.13.1-1 [1,221 k
Get:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-lmodern all 2.004.5-6.1 [
Get:15 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-noto-mono all 20201225-1build
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-texgyre all 20180621-3.1
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libapache-pom-java all 18-1 [4,
Get:18 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-parent-java all 43-1
Get:19 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-logging-java all 1.2
```

```
Get:20 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libptexenc1 amd64 2021.2021
Get:21 http://archive.ubuntu.com/ubuntu jammy/main amd64 rubygems-integration all 1.18 [5,33
Get:22 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby3.0 amd64 3.0.2-7ubuntu
Get:23 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-rubygems all 3.3.5-2 [228 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby amd64 1:3.0~exp1 [5,100 B]
Get:25 http://archive.ubuntu.com/ubuntu jammy/main amd64 rake all 13.0.6-2 [61.7 kB]
Get:26 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-net-telnet all 0.1.1-2 [12.6 k
Get:27 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby-webrick all 1.7.0-3ubu
Get:28 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby-xmlrpc all 0.3.2-1ubun
Get:29 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libruby3.0 amd64 3.0.2-7ubu
Get:30 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libsynctex2 amd64 2021.2021
Get:31 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libteckit0 amd64 2.5.11+ds1-1 [
Get:32 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexlua53 amd64 2021.2021
Get:33 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexluajit2 amd64 2021.20
Get:34 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libzzip-0-13 amd64 0.13.72+dfsg
Get:35 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-encodings all 1:1.0.5-0ubunt
Get:36 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-utils amd64 1:7.7+6build2 [9
Get:37 http://archive.ubuntu.com/ubuntu jammy/universe amd64 lmodern all 2.004.5-6.1 [9,471
Get:38 http://archive.ubuntu.com/ubuntu jammy/universe amd64 preview-latex-style all 12.2-1u
Get:39 http://archive.ubuntu.com/ubuntu jammy/main amd64 t1utils amd64 1.41-4build2 [61.3 kB
Get:40 http://archive.ubuntu.com/ubuntu jammy/universe amd64 teckit amd64 2.5.11+ds1-1 [699
Get:41 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-gyre all 20180621-3.1 [6,20
Get:42 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 texlive-binaries amd64
Get:43 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-base all 2021.20220204-
Get:44 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-fonts-recommended all 2
Get:45 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-base all 2021.202
Get:46 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfontbox-java all 1:1.8.16-2
Get:47 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libpdfbox-java all 1:1.8.16-2 [
Get:48 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-recommended all 2
Get:49 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-pictures all 2021.20220
Get:50 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra all 2021.20
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-generic all 2021.
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21 [2,967 kB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all 2021.20220204
Fetched 182 MB in 3s (69.8 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages ...
Selecting previously unselected package fonts-droid-fallback.
(Reading database ... 126558 files and directories currently installed.)
Preparing to unpack .../00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb ...
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) ...
Selecting previously unselected package fonts-lato.
```

```
Preparing to unpack .../01-fonts-lato_2.0-2.1_all.deb ...
Unpacking fonts-lato (2.0-2.1) ...
Selecting previously unselected package poppler-data.
Preparing to unpack .../02-poppler-data_0.4.11-1_all.deb ...
Unpacking poppler-data (0.4.11-1) ...
Selecting previously unselected package tex-common.
Preparing to unpack .../03-tex-common_6.17_all.deb ...
Unpacking tex-common (6.17) ...
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack .../04-fonts-urw-base35_20200910-1_all.deb ...
Unpacking fonts-urw-base35 (20200910-1) ...
Selecting previously unselected package libgs9-common.
Preparing to unpack .../05-libgs9-common_9.55.0~dfsg1-0ubuntu5.11_all.deb ...
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.11) ...
Selecting previously unselected package libidn12:amd64.
Preparing to unpack .../06-libidn12_1.38-4ubuntu1_amd64.deb ...
Unpacking libidn12:amd64 (1.38-4ubuntu1) ...
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack .../07-libijs-0.35_0.35-15build2_amd64.deb ...
Unpacking libijs-0.35:amd64 (0.35-15build2) ...
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack .../08-libjbig2dec0_0.19-3build2_amd64.deb ...
Unpacking libjbig2dec0:amd64 (0.19-3build2) ...
Selecting previously unselected package libgs9:amd64.
Preparing to unpack .../09-libgs9_9.55.0~dfsg1-0ubuntu5.11_amd64.deb ...
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.11) ...
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack .../10-libkpathsea6_2021.20210626.59705-1ubuntu0.2_amd64.deb ...
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libwoff1:amd64.
Preparing to unpack .../11-libwoff1_1.0.2-1build4_amd64.deb ...
Unpacking libwoff1:amd64 (1.0.2-1build4) ...
Selecting previously unselected package dvisvgm.
Preparing to unpack .../12-dvisvgm_2.13.1-1_amd64.deb ...
Unpacking dvisvgm (2.13.1-1) ...
Selecting previously unselected package fonts-lmodern.
Preparing to unpack .../13-fonts-lmodern_2.004.5-6.1_all.deb ...
Unpacking fonts-lmodern (2.004.5-6.1) ...
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack .../14-fonts-noto-mono_20201225-1build1_all.deb ...
Unpacking fonts-noto-mono (20201225-1build1) ...
Selecting previously unselected package fonts-texgyre.
```

```
Preparing to unpack .../15-fonts-texgyre_20180621-3.1_all.deb ...
Unpacking fonts-texgyre (20180621-3.1) ...
Selecting previously unselected package libapache-pom-java.
Preparing to unpack .../16-libapache-pom-java_18-1_all.deb ...
Unpacking libapache-pom-java (18-1) ...
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack .../17-libcommons-parent-java_43-1_all.deb ...
Unpacking libcommons-parent-java (43-1) ...
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack .../18-libcommons-logging-java_1.2-2_all.deb ...
Unpacking libcommons-logging-java (1.2-2) ...
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack .../19-libptexenc1_2021.20210626.59705-1ubuntu0.2_amd64.deb ...
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package rubygems-integration.
Preparing to unpack .../20-rubygems-integration_1.18_all.deb ...
Unpacking rubygems-integration (1.18) ...
Selecting previously unselected package ruby3.0.
Preparing to unpack .../21-ruby3.0_3.0.2-7ubuntu2.10_amd64.deb ...
Unpacking ruby3.0 (3.0.2-7ubuntu2.10) ...
Selecting previously unselected package ruby-rubygems.
Preparing to unpack .../22-ruby-rubygems_3.3.5-2_all.deb ...
Unpacking ruby-rubygems (3.3.5-2) ...
Selecting previously unselected package ruby.
Preparing to unpack .../23-ruby_1%3a3.0~exp1_amd64.deb ...
Unpacking ruby (1:3.0~exp1) ...
Selecting previously unselected package rake.
Preparing to unpack .../24-rake_13.0.6-2_all.deb ...
Unpacking rake (13.0.6-2) ...
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack .../25-ruby-net-telnet_0.1.1-2_all.deb ...
Unpacking ruby-net-telnet (0.1.1-2) ...
Selecting previously unselected package ruby-webrick.
Preparing to unpack .../26-ruby-webrick_1.7.0-3ubuntu0.1_all.deb ...
Unpacking ruby-webrick (1.7.0-3ubuntu0.1) ...
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack .../27-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb ...
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) ...
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack .../28-libruby3.0_3.0.2-7ubuntu2.10_amd64.deb ...
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.10) ...
Selecting previously unselected package libsynctex2:amd64.
```

```
Preparing to unpack .../29-libsynctex2_2021.20210626.59705-1ubuntu0.2_amd64.deb ...
Unpacking libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack .../30-libteckit0_2.5.11+ds1-1_amd64.deb ...
Unpacking libteckit0:amd64 (2.5.11+ds1-1) ...
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack .../31-libtexlua53_2021.20210626.59705-1ubuntu0.2_amd64.deb ...
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack .../32-libtexluajit2_2021.20210626.59705-1ubuntu0.2_amd64.deb ...
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack .../33-libzzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb ...
Unpacking libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) ...
Selecting previously unselected package xfonts-encodings.
Preparing to unpack .../34-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb ...
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) ...
Selecting previously unselected package xfonts-utils.
Preparing to unpack .../35-xfonts-utils_1%3a7.7+6build2_amd64.deb ...
Unpacking xfonts-utils (1:7.7+6build2) ...
Selecting previously unselected package lmodern.
Preparing to unpack .../36-lmodern_2.004.5-6.1_all.deb ...
Unpacking lmodern (2.004.5-6.1) ...
Selecting previously unselected package preview-latex-style.
Preparing to unpack .../37-preview-latex-style_12.2-1ubuntu1_all.deb ...
Unpacking preview-latex-style (12.2-1ubuntu1) ...
Selecting previously unselected package t1utils.
Preparing to unpack .../38-t1utils_1.41-4build2_amd64.deb ...
Unpacking t1utils (1.41-4build2) ...
Selecting previously unselected package teckit.
Preparing to unpack .../39-teckit_2.5.11+ds1-1_amd64.deb ...
Unpacking teckit (2.5.11+ds1-1) ...
Selecting previously unselected package tex-gyre.
Preparing to unpack .../40-tex-gyre_20180621-3.1_all.deb ...
Unpacking tex-gyre (20180621-3.1) ...
Selecting previously unselected package texlive-binaries.
Preparing to unpack .../41-texlive-binaries_2021.20210626.59705-1ubuntu0.2_amd64.deb ...
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package texlive-base.
Preparing to unpack .../42-texlive-base_2021.20220204-1_all.deb ...
Unpacking texlive-base (2021.20220204-1) ...
Selecting previously unselected package texlive-fonts-recommended.
```

```
Preparing to unpack .../43-texlive-fonts-recommended_2021.20220204-1_all.deb ...
Unpacking texlive-fonts-recommended (2021.20220204-1) ...
Selecting previously unselected package texlive-latex-base.
Preparing to unpack .../44-texlive-latex-base_2021.20220204-1_all.deb ...
Unpacking texlive-latex-base (2021.20220204-1) ...
Selecting previously unselected package libfontbox-java.
Preparing to unpack .../45-libfontbox-java_1%3a1.8.16-2_all.deb ...
Unpacking libfontbox-java (1:1.8.16-2) ...
Selecting previously unselected package libpdfbox-java.
Preparing to unpack .../46-libpdfbox-java_1%3a1.8.16-2_all.deb ...
Unpacking libpdfbox-java (1:1.8.16-2) ...
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack .../47-texlive-latex-recommended_2021.20220204-1_all.deb ...
Unpacking texlive-latex-recommended (2021.20220204-1) ...
Selecting previously unselected package texlive-pictures.
Preparing to unpack .../48-texlive-pictures_2021.20220204-1_all.deb ...
Unpacking texlive-pictures (2021.20220204-1) ...
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack .../49-texlive-latex-extra_2021.20220204-1_all.deb ...
Unpacking texlive-latex-extra (2021.20220204-1) ...
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack .../50-texlive-plain-generic_2021.20220204-1_all.deb ...
Unpacking texlive-plain-generic (2021.20220204-1) ...
Selecting previously unselected package tipa.
Preparing to unpack .../51-tipa_2%3a1.3-21_all.deb ...
Unpacking tipa (2:1.3-21) ...
Selecting previously unselected package texlive-xetex.
Preparing to unpack .../52-texlive-xetex_2021.20220204-1_all.deb ...
Unpacking texlive-xetex (2021.20220204-1) ...
Setting up fonts-lato (2.0-2.1) ...
Setting up fonts-noto-mono (20201225-1build1) ...
Setting up libwoff1:amd64 (1.0.2-1build4) ...
Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up libijs-0.35:amd64 (0.35-15build2) ...
Setting up libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up libfontbox-java (1:1.8.16-2) ...
Setting up rubygems-integration (1.18) ...
Setting up libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) ...
Setting up fonts-urw-base35 (20200910-1) ...
Setting up poppler-data (0.4.11-1) ...
Setting up tex-common (6.17) ...
update-language: texlive-base not installed and configured, doing nothing!
```

```
Setting up libjbig2dec0:amd64 (0.19-3build2) ...
Setting up libteckit0:amd64 (2.5.11+ds1-1) ...
Setting up libapache-pom-java (18-1) ...
Setting up ruby-net-telnet (0.1.1-2) ...
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) ...
Setting up t1utils (1.41-4build2) ...
Setting up libidn12:amd64 (1.38-4ubuntu1) ...
Setting up fonts-texgyre (20180621-3.1) ...
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up ruby-webrick (1.7.0-3ubuntu0.1) ...
Setting up fonts-lmodern (2.004.5-6.1) ...
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) ...
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) ...
Setting up libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.11) ...
Setting up teckit (2.5.11+ds1-1) ...
Setting up libpdfbox-java (1:1.8.16-2) ...
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.11) ...
Setting up preview-latex-style (12.2-1ubuntu1) ...
Setting up libcommons-parent-java (43-1) ...
Setting up dvisvgm (2.13.1-1) ...
Setting up libcommons-logging-java (1.2-2) ...
Setting up xfonts-utils (1:7.7+6build2) ...
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.2) ...
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin (xdvi.bin) in auto
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex (bibtex) in a
Setting up lmodern (2.004.5-6.1) ...
Setting up texlive-base (2021.20220204-1) ...
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST...
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN...
mktexlsr: Updating /var/lib/texmf/ls-R...
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4: /var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4: /var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-ini-files/pdft
Setting up tex-gyre (20180621-3.1) ...
```

```
Setting up texlive-plain-generic (2021.20220204-1) ...
Setting up texlive-latex-base (2021.20220204-1) ...
Setting up texlive-latex-recommended (2021.20220204-1) ...
Setting up texlive-pictures (2021.20220204-1) ...
Setting up texlive-fonts-recommended (2021.20220204-1) ...
Setting up tipa (2:1.3-21) ...
Setting up texlive-latex-extra (2021.20220204-1) ...
Setting up texlive-xetex (2021.20220204-1) ...
Setting up rake (13.0.6-2) ...
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.10) ...
Setting up ruby3.0 (3.0.2-7ubuntu2.10) ...
Setting up ruby (1:3.0~exp1) ...
Setting up ruby-rubygems (3.3.5-2) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
/sbin/ldconfig.real: /usr/local/lib/libhwloc.so.15 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm_debug.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libumf.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link
```

```
Processing triggers for tex-common (6.17) ...
Running updmap-sys. This may take some time... done.
Running mktexlsr /var/lib/texmf ... done.
Building format(s) --all.
    This may take some time... done.
TeX packages installed successfully.
--2025-04-26 22:32:56--  https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443...
HTTP request sent, awaiting response... 200 OK
Length: 1120047 (1.1M) [text/plain]
Saving to: ‘AMTAIR_Prototype_example_rain-sprinkler-lawn.ipynb’

AMTAIR_Prototype_ex 100%[===================>]   1.07M  --.-KB/s    in 0.06s

2025-04-26 22:32:56 (17.0 MB/s) - ‘AMTAIR_Prototype_example_rain-sprinkler-lawn.ipynb’ saved
```

UNIVERSITÄT BAYREUTH

– P&E Master's Programme –
Chair of Philosophy, Computer Science & Artificial Intelligence

Affidavit

Declaration of Academic Honesty

Hereby, I attest that I have composed and written the presented thesis

Automating the Modelling of Transformative Artificial Intelligence Risks

independently on my own, without the use of other than the stated aids and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

This paper has neither been previously submitted in the same or a similar form to another authority nor has it been published yet.

Bayreuth on the
May 23, 2025

Valentin Meyer