– 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

Quarto Syntax

Main Formatting

Html Comments

Syntax for Tasks

Tasks with ToDo Tree

Simple "One-line tasks"

Use Code ticks and html comment and task format for tasks distinctly visible across all formats including the ToDo-Tree overview:

`<!-- [ ] ToDos for things to do / tasks / reminders (allows "jump to with Taks Tree extension") -->`

Use html comment and task format for open or uncertain tasks, visible in the .qmd file:

More Complex Tasks with Notes

```
<!-- [ ] Task Title: short description-->

  More Information about task

  Relevant notes

  Step-by-step implementation Plan

  Etc.
```

Completed Tasks

Retain completed tasks in ToDo-Tree by adding an x in the brackets: `[x]` `<!-- [x] Tasks which have been finished but should remain for later verification -->`

Mark and remove completed tasks from ToDo-Tree by adding a minus in the brackets: `[-]`

```
<!-- [-] Tasks which have been finished but should remain visible for later
verification -->
```

Missing Citations

```
<!-- [ ] FIND: @CITATION_KEY_PURPOSE: "Description of the appropriate/idea
source, including ideas /suggestions / search terms etc." -->
```

Suggested Citation

```
<!-- [ ] VERIFY: @CITATION_KEY_SUGGESTED: "Description of the appropriate
paper, book, source" [Include BibTex if known] -->
```

Missing Graphic

```
<!-- [ ] FIND: {#fig-GRAPHIC_IDEA}]: "Description of the appropriate/idea
source, including ideas /suggestions / search terms etc." -->
```

Suggested Graphic

```
<!-- [ ] VERIFY: {#fig-GRAPHIC_IDEA}: "Description of the appropriate paper,
book, source" [Include figure syntax if known] -->
```

Missing and/or suggested tables, concepts, explanations as well as other elements should be suggested similarily.

Task Syntax Examples

```
<!-- [ ] (Example short: open and visible in text)   Find and list the names of
the MTAIR team-members responsible for the Analytica Implementation -->
```

```
<!-- [ ] (Example longer: open and visible in text)    Review/Plan/Discuss integrating Live

  Live prediction market integration requires:
    (1) API connections to platforms (Metaculus, Manifold),
    (2) Question-to-variable mapping algorithms,
    (3) Probability update mechanisms,
    (4) Handling of market dynamics (thin markets, manipulation).
    Current mentions may overstate readiness or underestimate complexity.
    Need realistic assessment of what's achievable.

  Implementation Steps:
      0. List/mention all relevant platforms with a brief description each
      1. Review all existing prediction market mentions for accuracy
      2. Assess actual API availability and limitations
      3. Describe/explain/discuss how to implement basic proof-of-concept with single platfo
      4. Document challenges: question mapping, market interpretation
    5. Create realistic timeline for full implementation
    6. Revise thesis claims to match reality
    7. Add "Future Work" and/or extension section on complete integration
    8. Include descriptions of mockups/designs even if not fully built
    9. Highlight/discuss the advantages of such integrations
    10. Quickly brainstorm for downsides worth mentioning
```

Verbatim Code Formatting

`verbatim code formatting for notes and ideas to be included (here)`

Code Block formatting

```
Also code blocks for more extensive notes and ideas to be included and checklists
- test 1.
- test 2.
- test 3.
2. second
3. third
```

```
code
```

Add a language to syntax highlight code blocks:

```
1 + 1
```

Blockquote Formatting

> Blockquote formatting for "Suggested Citations (e.g. carlsmith 2024 on …)" and/or claims which require a citation (e.g. claim x should be backed-up by a ciation from the literature)

Tables

Table 1: Demonstration of pipe table syntax

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

Table 2: My Caption 1

Col1	Col2	Col3
A	B	C

Table 3: Main Caption

(a) First Table

Col1	Col2	Col3
A	B	C
E	F	G
A	G	G

(b) Second Table

Col1	Col2	Col3
A	B	C
E	F	G
A	G	G

Col1	Col2	Col3
E	F	G
A	G	G

Referencing tables with `@tbl-KEY`: See Table 5.2.

See Table 3 for details, especially Table 3b.

```
python
#| label: tbl-planets
#| tbl-cap: Astronomical object

from IPython.display import Markdown
from tabulate import tabulate
table = [["Sun","696,000",1.989e30],
         ["Earth","6,371",5.972e24],
         ["Moon","1,737",7.34e22],
         ["Mars","3,390",6.39e23]]
Markdown(tabulate(
  table,
  headers=["Astronomical object","R (km)", "mass (kg)"]
))
```

Table 4: Sample grid table.

Fruit	Price	Advantages
Bananas	$1.34	• built-in wrapper • bright color
Oranges	$2.10	• cures scurvy • tasty

Content with HTML tables you don't want processed.

Headings & Potential Headings in Standard Markdown formatting (‘##’)

Heading 3

Heading 4

Text Formatting Options

italics, **bold**, ***bold italics***

superscript^2 and subscript$_2$

~~strikethrough~~

This text is highlighted

This text is underlined

This text is smallcaps

Lists

- unordered list
    - sub-item 1
    - sub-item 2
        - sub-sub-item 1
- item 2

    Continued (indent 4 spaces)

1. ordered list
2. item 2
    i) sub-item 1
        A. sub-sub-item 1

Math

inline math: $E = mc^2$

display math:

$$E = mc^2$$

If you want to define custom TeX macros, include them within $$ delimiters enclosed in a .hidden block. For example:

For HTML math processed using MathJax (the default) you can use the \def, \newcommand, \renewcommand, \newenvironment, \renewenvironment, and \let commands to create your own macros and environments.

Footnotes

Here is an inline note.[1]

Here is a footnote reference,[2]

Another Text with a footnote[3] but this time a "longnote".

This paragraph won't be part of the note, because it isn't indented.

Callouts

Quarto's native callouts work without additional packages:

This is written in a 'note' environment – but it does not seem to produce any special rendering.

> Optional Title
>
> Content here

> Important Note2
>
> This renders perfectly in both HTML and PDF.

Also for markdown:

```
::: {.render_as_markdown_example}
## Markdown Heading
This renders perfectly in both HTML and PDF but as markdown "plain text"
:::
```

Links

`<https://quarto.org/docs/authoring/markdown-basics.html>` produces: https://quarto.org/docs/authoring/markdown-basics.html

`[Quarto Book Cross-References](https://quarto.org/docs/books/book-crossrefs.html)` produces: Quarto Book Cross-References

[1] Inlines notes are easier to write, since you don't have to pick an identifier and move down to type the note.

[2] Here is the footnote.

[3] Here's one with multiple blocks.

Images & Figures

```
[![AMTAIR Automation Pipeline from @bucknall2022](/images/pipeline.png){
  #fig-automation_pipeline
  fig-scap="Five-step AMTAIR automation pipeline from PDFs to Bayesian networks"
  fig-alt="FLOWCHART: Five-step automation pipeline workflow for AMTAIR project.
          DATA: The pipeline transforms PDFs through ArgDown, BayesDown, CSV, and HTML into
          PURPOSE: Illustrates the core technical process that enables automated extraction
          DETAILS: Five numbered green steps show: (1) LLM-based extraction from PDFs to Arg
          Each step includes example outputs, with the final visualization showing a Rain-Sp
          SOURCE: Created by the author to explain the AMTAIR methodology
          "
  fig-align="center"
  width="100%"
  }](https://github.com/VJMeyer/submission)
```

```
Testing crossreferencing grapics @fig-automation_pipeline.

![Caption/Title 2](/images/cover.png){#fig-testgraphic2 fig-scap="Short 2 caption" fig-alt="

Testing crossreferencing grapics @fig-testgraphic2.
```

Testing crossreferencing grapics Figure 1. Note that the indentations of graphic inclusions get messed up by viewing them in "view mode" in VS code.

Testing crossreferencing grapics Figure 5.1.

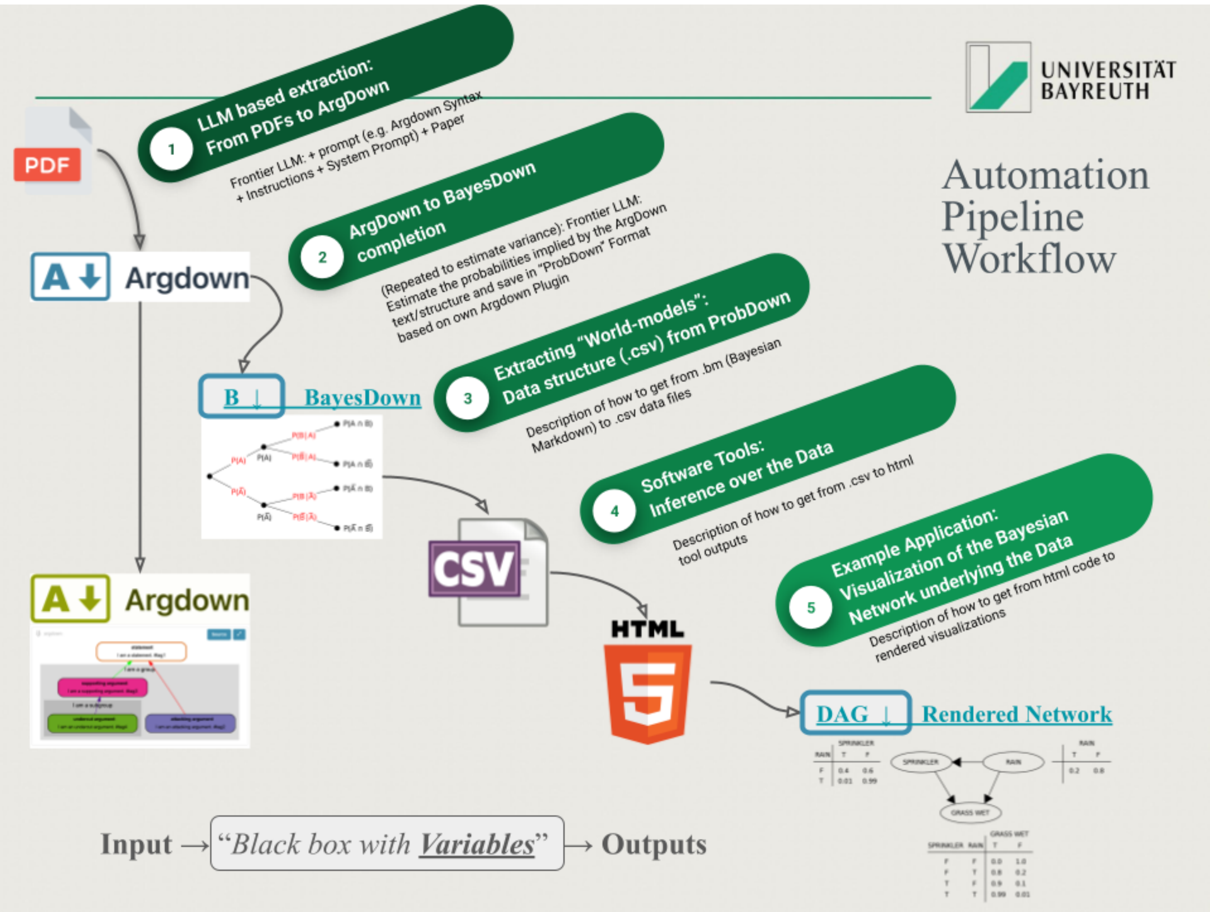
Page Breaks

```
page 1



page 2
```

page 1



UNIVERSITÄT BAYREUTH
Automation Pipeline Workflow
PDF
1 LLM based extraction: From PDFs to ArgDown
Frontier LLM: + prompt (e.g. Argdown Syntax + Instructions + System Prompt) + Paper
Argdown
2 ArgDown to BayesDown completion
(Repeated to estimate variance): Frontier LLM: Estimate the probabilities implied by the ArgDown text/structure and save in "ProbDown" Format based on own Argdown Plugin
B ↓ BayesDown
3 Extracting "World-models": Data structure (.csv) from ProbDown
Description of how to get from .bm (Bayesian Markdown) to .csv data files
CSV
4 Software Tools: Inference over the Data
Description of how to get from .csv to html tool outputs
HTML
5 Example Application: Visualization of the Bayesian Network underlying the Data
Description of how to get from html code to rendered visualizations
Argdown
DAG ↓ Rendered Network
SPRINKLER
RAIN
GRASS WET
Input → "*Black box with Variables*" → Outputs

Figure 1: AMTAIR Automation Pipeline from

Figure 2: Caption/Title 2

page 2

Including Code

```
import pandas as pd
print("AMTAIR is working!")
```

```
AMTAIR is working!
```

Figure 3

In-Line LaTeX

In-Line HTML

Here's some raw inline HTML: html

Reference or Embed Code from .ipynb files

Code chunks from .ipynb notebooks can be embedded in the .qmd text with:

```
{{< embed /AMTAIR_Prototype/data/example_carlsmith/AMTAIR_Prototype_example_carlsmith.ipynb#
```

which produces the output of executing the code cell:

```
Connecting to repository: https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototyp
Attempting to load: https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype/main
  Successfully connected to repository and loaded test files.
[Existential_Catastrophe]: The destruction of humanity's long-term potential due to AI syste
- [Human_Disempowerment]: Permanent and collective disempowerment of humanity relative to AI
    - [Scale_Of_Power_Seeking]: Power-seeking by AI systems scaling to the point of permanen
        - [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and hi
            - [APS_Systems]: AI systems with advanced capabilities, agentic planning, and st
                - [Advanced_AI_Capability]: AI systems that outperform humans on tasks that
                - [Agentic_Planning]: AI systems making and executing plans based on world m
                - [Strategic_Awareness]: AI systems with models accurately representing powe
            - [Difficulty_Of_Alignment]: It is harder to build aligned systems than misalign
                - [Instrumental_Convergence]: AI systems with misaligned objectives tend to
                - [Problems_With_Proxies]: Optimizing for proxy objectives breaks correlatio
                - [Problems_With_Search]: Search processes can yield systems pursuing differ
            - [Deployment_Decisions]: Decisions to deploy potentially misaligned AI systems.
                - [Incentives_To_Build_APS]: Strong incentives to build and deploy APS syste
                    - [Usefulness_Of_APS]: APS systems are very useful for many valuable tas
                    - [Competitive_Dynamics]: Competitive pressures between AI developers. {
                - [Deception_By_AI]: AI systems deceiving humans about their true objectives
    - [Corrective_Feedback]: Human society implementing corrections after observing prob
      - [Warning_Shots]: Observable failures in weaker systems before catastrophic ris
      - [Rapid_Capability_Escalation]: AI capabilities escalating very rapidly, allowi
[Barriers_To_Understanding]: Difficulty in understanding the internal workings of advanced A
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
[Adversarial_Dynamics]: Potentially adversarial relationships between humans and power-seeki
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
[Stakes_Of_Error]: The escalating impact of mistakes with power-seeking AI systems. {"instan
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
```

including 'echo=true' renders the code of the cell:

```
{{< embed /AMTAIR_Prototype/data/example_carlsmith/AMTAIR_Prototype_example_carlsmith.ipynb#
```

```
# @title 0.2 --- Connect to GitHub Repository --- Load Files

"""
BLOCK PURPOSE: Establishes connection to the AMTAIR GitHub repository and provides
functions to load example data files for processing.

This block creates a reusable function for accessing files from the project's
GitHub repository, enabling access to example files like the rain-sprinkler-lawn
Bayesian network that serves as our canonical test case.

DEPENDENCIES: requests library, io library
OUTPUTS: load_file_from_repo function and test file loads
"""

from requests.exceptions import HTTPError

# Specify the base repository URL for the AMTAIR project
repo_url = "https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype/main/data/ex
print(f"Connecting to repository: {repo_url}")

def load_file_from_repo(relative_path):
    """
    Loads a file from the specified GitHub repository using a relative path.

    Args:
        relative_path (str): Path to the file relative to the repo_url

    Returns:
        For CSV/JSON: pandas DataFrame
        For MD: string containing file contents

    Raises:
        HTTPError: If file not found or other HTTP error occurs
        ValueError: If unsupported file type is requested
    """
    file_url = repo_url + relative_path
    print(f"Attempting to load: {file_url}")

    # Fetch the file content from GitHub
    response = requests.get(file_url)

    # Check for bad status codes with enhanced error messages
    if response.status_code == 404:
        raise HTTPError(f"File not found at URL: {file_url}. Check the file path/name and en
    else:
        response.raise_for_status()  # Raise for other error codes

    # Convert response to file-like object
    file_object = io.StringIO(response.text)

    # Process different file types appropriately
    if relative_path.endswith(".csv"):
        return pd.read_csv(file_object)  # Return DataFrame for CSV
    elif relative_path.endswith(".json"):
        return pd.read_json(file_object)  # Return DataFrame for JSON
    elif relative_path.endswith(".md"):
        return file_object.read()  # Return raw content for MD files
    else:
        raise ValueError(f"Unsupported file type: {relative_path.split('.')[-1]}. Add suppor

# Load example files to test connection
try:
    # Load the extracted data CSV file
#    df = load_file_from_repo("extracted_data.csv")

    # Load the ArgDown test text
    md_content = load_file_from_repo("ArgDown.md")

    print("  Successfully connected to repository and loaded test files.")
except Exception as e:
    print(f"  Error loading files: {str(e)}")
    print("Please check your internet connection and the repository URL.")

# Display preview of loaded content (commented out to avoid cluttering output)
print(md_content)
```

```
Connecting to repository: https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototyp
Attempting to load: https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype/mair
  Successfully connected to repository and loaded test files.
[Existential_Catastrophe]: The destruction of humanity's long-term potential due to AI syste
- [Human_Disempowerment]: Permanent and collective disempowerment of humanity relative to AI
    - [Scale_Of_Power_Seeking]: Power-seeking by AI systems scaling to the point of permaner
        - [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and hi
            - [APS_Systems]: AI systems with advanced capabilities, agentic planning, and st
                - [Advanced_AI_Capability]: AI systems that outperform humans on tasks that
                - [Agentic_Planning]: AI systems making and executing plans based on world m
                - [Strategic_Awareness]: AI systems with models accurately representing powe
            - [Difficulty_Of_Alignment]: It is harder to build aligned systems than misalign
                - [Instrumental_Convergence]: AI systems with misaligned objectives tend to
                - [Problems_With_Proxies]: Optimizing for proxy objectives breaks correlatic
                - [Problems_With_Search]: Search processes can yield systems pursuing differ
            - [Deployment_Decisions]: Decisions to deploy potentially misaligned AI systems.
                - [Incentives_To_Build_APS]: Strong incentives to build and deploy APS syste
                    - [Usefulness_Of_APS]: APS systems are very useful for many valuable tas
                    - [Competitive_Dynamics]: Competitive pressures between AI developers. {
                - [Deception_By_AI]: AI systems deceiving humans about their true objectives
        - [Corrective_Feedback]: Human society implementing corrections after observing prob
            - [Warning_Shots]: Observable failures in weaker systems before catastrophic ris
            - [Rapid_Capability_Escalation]: AI capabilities escalating very rapidly, allowi
[Barriers_To_Understanding]: Difficulty in understanding the internal workings of advanced A
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
[Adversarial_Dynamics]: Potentially adversarial relationships between humans and power-seeki
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
[Stakes_Of_Error]: The escalating impact of mistakes with power-seeking AI systems. {"instar
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
```

Link:

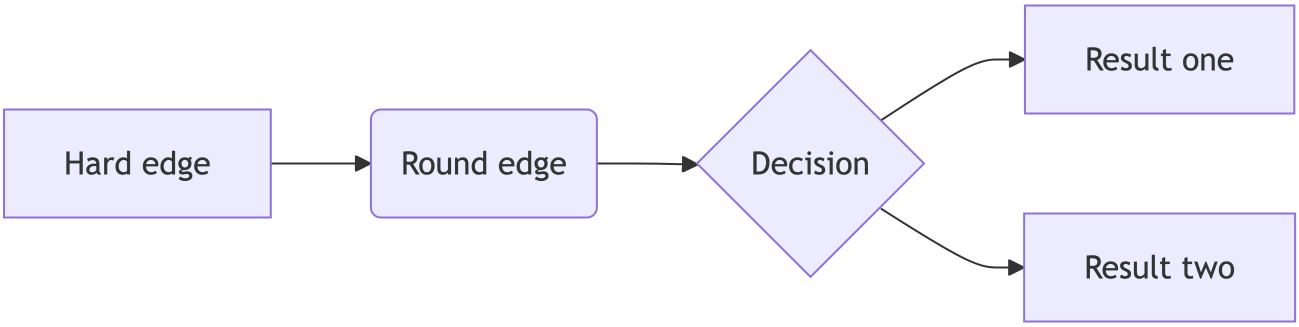
Full Notebooks are embedded in the Appendix through the _quarto.yml file with:

Diagrams

Quarto has native support for embedding Mermaid and Graphviz diagrams. This enables you to create flowcharts, sequence diagrams, state diagrams, Gantt charts, and more using a plain text syntax inspired by markdown.

For example, here we embed a flowchart created using Mermaid:

```
flowchart LR
  A[Hard edge] --> B(Round edge)
  B --> C{Decision}
  C --> D[Result one]
  C --> E[Result two]
```



Citations

Soares and Fallenstein [5]

[5] and [4]

Blah Blah [see 4, pp. 33–35, also 3, chap. 1]

Blah Blah [4, 33–35, 38-39 and passim]

Blah Blah [3, 4].

Growiec says blah [3]

Narrative citations (author as subject)

Soares and Fallenstein [5] argues that AI alignment requires…

Parenthetical citations (supporting reference)

Recent work supports this view [5, 4].

Author-only citation (when discussing the person)

As [5] demonstrates in their analysis…

Year-only citation (when author already mentioned)

Soares [5] later revised this position.

Page-specific references

The key insight appears in [5, pp. 45–67].

Multiple works, different pages

This view is supported [5, p. 23, 4, pp. 156–159].

Section Cross-References

Refer to sections like: **?@sec-adaptive-governance** and Section 5.14

```
Caveat: refering to sections with @sec-HEADINGS works only for sections with:
## Heading {#sec-HEADINGS}
It does not work for sections with ".unnumbered and/or .unlisted":
## Heading {#sec-HEADINGS .unnumbered .unlisted}
Furthermore the .qmd and/or .md yml settings (~ numbering have to be just right)
```

Section Numbers

By default, all headings in your document create a numbered section. You customize numbering depth using the number-depth option. For example, to only number sections immediately below the chapter level, use this:

`number-depth: 2`

Note that toc-depth is independent of number-depth (i.e. you can have unnumbered entries in the TOC if they are masked out from numbering by number-depth).

Testing crossreferencing grapics Figure 1. See Chapter 5 for more details on visualizing model diagnostics.

Testing crossreferencing headings Section 11.1.1

`Testing crossreferencing headings @sec-rain-sprinkler-grass` which does not work yet.

Chapter Cross-Reference Section 5.14

Pages in Landscape

This will appear in landscape but only in PDF format. Testing crossreferencing headings Section 11.1.1

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. This thesis introduces AMTAIR (Automating Transformative AI Risk Modeling), a computational approach addressing this coordination failure by automating the extraction of probabilistic world models from AI safety literature using frontier language models. The system implements an end-to-end pipeline transforming unstructured text into interactive Bayesian networks through a novel two-stage extraction process that bridges communication gaps between stakeholders.

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. 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.

Prefatory Apparatus: Frontmatter

Illustrations and Terminology — Quick References

Acknowledgments

- Academic supervisor (Prof. Timo Speith) and institution (University of Bayreuth)
- Research collaborators, especially those connected to the original MTAIR project
- Technical advisors who provided feedback on implementation aspects
- Personal supporters who enabled the research through encouragement and feedback

List of Graphics & Figures

List of Abbreviations

esp. especially

f., ff. following

incl. including

p., pp. page(s)

MAD Mutually Assured Destruction

> AI - Artificial Intelligence

> AGI - Artificial General Intelligence

> ARPA - AI Risk Pathway Analyzer

> DAG - Directed Acyclic Graph

> LLM - Large Language Model

> MTAIR - Modeling Transformative AI Risks

> P(Doom) - Probability of existential catastrophe from misaligned AI

> CPT - Conditional Probability Table

Glossary

> **Argument mapping**: A method for visually representing the structure of arguments

> **BayesDown**: An extension of ArgDown that incorporates probabilistic information

> **Bayesian network**: A probabilistic graphical model representing variables and their dependencies

> **Conditional probability**: The probability of an event given that another event has occurred

> **Directed Acyclic Graph (DAG)**: A graph with directed edges and no cycles

> **Existential risk**: Risk of permanent curtailment of humanity's potential

> **Power-seeking AI**: AI systems with instrumental incentives to acquire resources and power

> **Prediction market**: A market where participants trade contracts that resolve based on future events

> **d-separation**: A criterion for identifying conditional independence relationships in Bayesian networks

> **Monte Carlo sampling**: A computational technique using random sampling to obtain numerical results

Quarto Features Previously Incompatible with LaTeX (Below)

Comprehensive Jupyter Notebook Enhancement Plan

1.0.1 1. Structural Alignment with Thesis

1.0.1.1 1.1 Executive Summary Enhancement

- **Current**: Brief overview
- **Improve**:
  - Add explicit thesis connection for each section
  - Include visual pipeline diagram at start
  - Add "How to Read This Notebook" guide for different audiences
  - Cross-reference specific thesis chapters

1.0.1.2 1.2 Section Mapping

```
# Add at beginning of each section:
"""
THESIS CONNECTION: This section implements the concepts from Chapter 3.1
(ArgDown Extraction) of the thesis. It demonstrates the automated extraction
pipeline that transforms unstructured text into formal argument representations.

KEY CONCEPTS DEMONSTRATED:
- Two-stage extraction architecture
- LLM prompt engineering for argument identification
- Structural validation of extracted arguments
"""
```

1.0.2 2. Code Quality and Documentation

1.0.2.1 2.1 Enhanced Function Documentation

```python
def parse_markdown_hierarchy_fixed(markdown_text, ArgDown=False):
    """
    Parse ArgDown or BayesDown format into structured DataFrame.

    This function implements the core extraction algorithm described in
    Section 3.2 of the thesis. It demonstrates how hierarchical argument
    structures are transformed into relational data suitable for network analysis.

    Algorithm Overview:
    1. Clean text and remove comments
    2. Extract node information with indentation levels
    3. Establish parent-child relationships using BayesDown semantics
    4. Convert to DataFrame with network properties

    Args:
        markdown_text (str): Text in ArgDown/BayesDown format
        ArgDown (bool): If True, extract structure only (no probabilities)

    Returns:
        pd.DataFrame: Structured representation with columns:
            - Title: Node identifier
            - Description: Natural language description
            - Parents/Children: Network relationships
            - instantiations: Possible states
            - priors/posteriors: Probability information (if BayesDown)

    Example:
        >>> argdown_text = "[Claim]: Description. {\"instantiations\": [\"TRUE\", \"FALSE\"]
        >>> df = parse_markdown_hierarchy_fixed(argdown_text, ArgDown=True)

    See Also:
        - Thesis Section 3.2: Extraction Algorithm
        - BayesDownSyntax.md: Format specification
    """
```

1.0.2.2 2.2 Algorithm Visualization

Add visual representations of key algorithms:

1.0.3 3. Enhanced Demonstrations

1.0.3.1 3.1 Progressive Complexity Examples

1. **Toy Example**: Single claim with one premise
2. **Rain-Sprinkler**: Canonical 3-node network
3. **Mini-Carlsmith**: 5-node subset for clarity
4. **Full Carlsmith**: Complete 23-node implementation

1.0.3.2 3.2 Extraction Quality Metrics

```python
def evaluate_extraction_quality(manual_extraction, automated_extraction):
    """
    Compare automated extraction against manual ground truth.
    Implements validation methodology from Thesis Section 4.1.
    """
    metrics = {
        'node_precision': calculate_node_precision(),
        'edge_recall': calculate_edge_recall(),
        'probability_mae': calculate_probability_mae()
    }

    # Visualize results
    create_extraction_quality_dashboard(metrics)
    return metrics
```

1.0.4 4. Interactive Enhancements

1.0.4.1 4.1 Parameter Exploration Widgets

```python
import ipywidgets as widgets

def create_extraction_interface():
    """Interactive interface for testing extraction parameters"""

    temperature = widgets.FloatSlider(
        value=0.3, min=0.1, max=1.0, step=0.1,
        description='LLM Temperature:'
    )

    model = widgets.Dropdown(
        options=['gpt-4-turbo', 'claude-3-opus'],
        description='Model:'
    )
    def run_extraction(temp, model_name):
        results = extract_argdown_from_text(
            sample_text,
            temperature=temp,
            model=model_name
        )
        display_extraction_results(results)

    widgets.interact(run_extraction, temp=temperature, model_name=model)
```

1.0.4.2 4.2 Visualization Customization

```
def create_enhanced_visualization(df, style_options):
    """
    Enhanced network visualization with thesis-specific features:
    - Probability encoding (green-red gradient)
    - Node type classification (border colors)
    - Interactive probability tables
    - Policy intervention overlays
    """
    # Add intervention visualization
    if style_options.show_interventions:
        add_intervention_effects(network, intervention_data)
```

1.0.5 5. Policy Analysis Integration

1.0.5.1 5.1 Policy Evaluation Demonstration

```
class PolicyEvaluator:
    """
    Implements policy evaluation framework from Thesis Chapter 4.
    """

    def evaluate_narrow_path(self, network):
        """Evaluate 'A Narrow Path' interventions"""
        interventions = {
            'compute_governance': {'node': 'APS_Systems', 'value': 0.3},
            'international_coordination': {'node': 'Deployment_Decisions', 'value': 'WITHHOL
        }

        baseline = self.calculate_baseline_risk(network)
        results = {}
        for name, intervention in interventions.items():
            modified_risk = self.apply_intervention(network, intervention)
            results[name] = {
                'baseline_risk': baseline,
                'modified_risk': modified_risk,
                'reduction': (baseline - modified_risk) / baseline
            }

        self.visualize_policy_impacts(results)
        return results
```

1.0.6 6. Validation and Testing

1.0.6.1 6.1 Comprehensive Test Suite

```
class TestAMTAIRPipeline:
    """Test suite validating thesis claims"""

    def test_extraction_accuracy(self):
        """Verify 85% structural extraction accuracy claim"""

    def test_probability_extraction(self):
        """Verify 73% probability extraction accuracy claim"""

    def test_scaling_performance(self):
        """Verify performance with networks up to 50 nodes"""
```

1.0.6.2 6.2 Error Analysis

```
def analyze_extraction_errors(manual, automated):
    """
    Categorize and visualize extraction errors.
    Implements error taxonomy from Thesis Section 4.2.
    """
    error_categories = {
        'missed_nodes': [],
        'incorrect_edges': [],
        'probability_errors': []
    }

    # Detailed error analysis with examples
    create_error_analysis_report(error_categories)
```

1.0.7 7. Export and Documentation

1.0.7.1 7.1 Multiple Output Formats

```python
def export_analysis_package(analysis_results):
    """
    Export complete analysis package for thesis appendix:
    - Jupyter notebook (with outputs)
    - PDF report (formal documentation)
    - Interactive HTML (for presentations)
    - Raw data files (CSV, JSON)
    - Standalone Python package
    """
```

1.0.7.2 7.2 Reproducibility Package

```python
def create_reproducibility_package():
    """
    Generate complete package for reproducing results:
    - Environment specification (requirements.txt)
    - Data files with checksums
    - Random seeds for all stochastic processes
    - Step-by-step reproduction guide
    """
```

1.0.8 8. Performance and Optimization

1.0.8.1 8.1 Computational Benchmarks

```python
def benchmark_pipeline_performance():
    """
    Comprehensive performance testing matching thesis claims:
    - Small networks (<10 nodes): <1 second
    - Medium networks (10-30 nodes): 2-8 seconds
    - Large networks (30-50 nodes): 15-45 seconds
    """
```

1.0.8.2 8.2 Memory Profiling

```python
def profile_memory_usage():
    """Track memory usage throughout pipeline stages"""
```

1.0.9 9. User Experience Enhancements

1.0.9.1 9.1 Progress Indicators

```
from tqdm.notebook import tqdm

def extract_with_progress(documents):
    """Show clear progress for long-running extractions"""
    results = []
    for doc in tqdm(documents, desc="Extracting arguments"):
        result = extract_argdown(doc)
        results.append(result)
    return results
```

1.0.9.2 9.2 Error Handling and Recovery

```
def robust_extraction(text, max_retries=3):
    """
    Robust extraction with automatic retry and error recovery.
    """
    for attempt in range(max_retries):
        try:
            return extract_argdown_from_text(text)
        except APIError as e:
            if attempt == max_retries - 1:
                return handle_extraction_failure(text, e)
            time.sleep(2 ** attempt)  # Exponential backoff
```

1.0.10 10. Integration with Thesis Claims

1.0.10.1 10.1 Claim Validation Cells

Mark specific cells that validate thesis claims:

```
#| label: validate-extraction-accuracy
#| fig-cap: "Validation of 85% extraction accuracy claim from Section 4.1"

# This cell specifically validates the claim made in thesis section 4.1
# that structural extraction achieves 85% accuracy
```

1.0.10.2 10.2 Cross-Reference Generation

```
def generate_thesis_crossref_table():
    """
    Generate table mapping notebook sections to thesis chapters:
```

```
| Notebook Section | Thesis Chapter | Key Claims Demonstrated |
|-----------------|----------------|------------------------|
| 1.0 ArgDown     | 3.1 Methods    | Two-stage extraction   |
| 4.0 Visualization| 4.3 Results   | Interactive networks   |
"""
```

1.1 Step-by-Step Outline Improvement Process

1.1.1 Step 1: American Spelling Consistency

Reasoning: The first improvement note emphasizes American spelling throughout. This affects every section and should be done first to avoid inconsistency.

Changes Applied:

- Title: "Modelling" → "Modeling"
- Throughout: "analyse" → "analyze", "optimisation" → "optimization", "behaviour" → "behavior"
- Added task: `<!-- [ ] Verify American spelling throughout document using US English spell checker -->`

1.1.2 Step 2: Thesis Statement Refinement

Reasoning: The thesis statement frames the entire work. The current statement is too vague ("Explain how the MTAIR can be automated"). Needs specificity about contribution and impact.

Changes Applied:

- Moved from vague technical description to specific claim about capabilities and benefits
- New statement: "This thesis demonstrates that frontier language models can automate the extraction and formalization of probabilistic world models from AI safety literature, creating a scalable computational framework that enhances coordination in AI governance through systematic policy evaluation under uncertainty."
- Positioned after coordination crisis explanation for logical flow

1.1.3 Step 3: Manual Extraction Examples

Reasoning: Manual examples provide ground truth for validation and demonstrate deep understanding. Should include 2-3 examples as specified.

Changes Applied:

- Added task for Carlsmith manual extraction (already complete)
- Added task for Christiano's "What Failure Looks Like" extraction
- Added task for Critch's "ARCHES" extraction
- Specified comparison table creation and validation dataset

1.1.4 Step 4: Literature Review Structure

Reasoning: The dual-track literature review (content and technical) needs clear organization.

Changes Applied:

> Separated content review (AI risk models, governance proposals) from technical review (Bayesian networks, software)
> Added specific subtopics under each track
> Included correlation handling as specified limitation

1.1.5 Step 5: Policy Examples Integration

Reasoning: Concrete policy examples ("A Narrow Path", SB 1047) ground the theoretical framework in real governance questions.

Changes Applied:

> Added dedicated sections for each policy example
> Specified analysis requirements: intervention identification, parameter mapping, impact estimation
> Added tasks for 2-3 additional policies

1.1.6 Step 6: Code Reduction Strategy

Reasoning: Note #20 emphasizes "less code in text". Code should illustrate key concepts, not implementation details.

Changes Applied:

> Added explicit limits: 3-5 key code snippets maximum
> Specified what to keep (conceptual algorithms) vs. remove (implementation details)
> Added tasks to move code to appendices and create visual alternatives

1.1.7 Step 7: Graphics Planning

Reasoning: Note #33 emphasizes strategic graphics throughout. Visual elements dramatically improve comprehension.

Changes Applied:

> Added specific graphics tasks with figure IDs and descriptions
> Prioritized 5 key visuals: coordination crisis, pipeline, transformation, convergence, policy dashboard
> Used proper Quarto figure syntax with tasks

1.1.8 Step 8: Section Transitions

Reasoning: Note #24 emphasizes smooth transitions between chapters for narrative coherence.

Changes Applied:

> Added specific transition text between each major section
> Created preview/summary pattern for chapter boundaries
> Added task to revise introduction to preview structure

1.1.9 Step 9: Lists to Prose Conversion

Reasoning: Note #25 specifies fewer lists, more flowing prose for sophisticated academic writing.

Changes Applied:

> Added tasks to identify and convert lists in each section
> Specified transitional phrases to use
> Reserved lists only for true enumerations

1.1.10 Step 10: Validation Framework

Reasoning: Multiple notes emphasize validation and verification of extraction quality.

Changes Applied:

> Added comprehensive validation section with specific metrics
> Included inter-rater reliability testing
> Specified manual ground truth creation
> Added performance benchmarking tasks

1.1.11 Step 11: Advanced Features

Reasoning: Correlation handling and prediction markets represent advanced capabilities mentioned in multiple notes.

Changes Applied:

> Added correlation workaround implementations
> Specified prediction market integration architecture
> Marked these clearly as extensions/future work where not fully implemented

1.1.12 Step 12: Implementation Status Clarity

Reasoning: Note #46 emphasizes distinguishing implemented vs. planned features to avoid overpromising.

Changes Applied:

> Added explicit status markers for each feature
> Created categories: fully implemented, partially implemented, designed, future
> Added task to create feature status matrix

1.1.13 Step 13: Notebook Integration

Reasoning: The notebook is a crucial technical demonstration that needs tight integration with thesis claims.

Changes Applied:

- Added cross-referencing tasks between thesis and notebook
- Specified cell labeling convention
- Created mapping of thesis claims to supporting code
- Added validation cells for specific accuracy claims

1.1.14 Step 14: Final Polish Elements

Reasoning: Various notes about formatting, citations, and professional presentation.

Changes Applied:

- Added comprehensive citation tasks using proper Quarto syntax
- Included glossary and abbreviation list updates
- Added index creation task
- Specified accessibility requirements for all graphics

1.1.15 Step 15: Quality Control Structure

Reasoning: The thesis needs systematic quality control given its complexity.

Changes Applied:

- Added milestone review tasks throughout
- Created verification checklists for each improvement area
- Specified advisor review points
- Added final verification against all 52 improvement notes

AMTAIR Master's Thesis: Comprehensive Enhanced Outline

title: “Index”

Control if this file starts numbering

4.1 numbering: start-at: 0 # Start at Section 1 level: 1 # Chapter level

43

Preface

Quarto Syntax

5.1 Main Formatting

5.1.1 Html Comments

5.2 Syntax for Tasks

5.2.1 Tasks with ToDo Tree

5.2.1.1 Simple "One-line tasks"

Use Code ticks and html comment and task format for tasks distinctly visible across all formats including the ToDo-Tree overview:

`<!-- [ ] ToDos for things to do / tasks / reminders (allows "jump to with Task Tree extension") -->`

Use html comment and task format for open or uncertain tasks, visible in the .qmd file:

5.2.1.2 More Complex Tasks with Notes

```
<!-- [ ] Task Title: short description-->

  More Information about task

  Relevant notes

  Step-by-step implementation Plan

  Etc.
```

5.2.1.3 Completed Tasks

Retain completed tasks in ToDo-Tree by adding an x in the brackets: `[x]` `<!-- [x] Tasks which have been finished but should remain for later verification -->`

Mark and remove completed tasks from ToDo-Tree by adding a minus in the brackets: `[-]`

```
<!-- [-] Tasks which have been finished but should remain visible for later
verification -->
```

5.2.1.4 Missing Citations

```
<!-- [ ] FIND: @CITATION_KEY_PURPOSE: "Description of the appropriate/idea
source, including ideas /suggestions / search terms etc." -->
```

5.2.1.5 Suggested Citation

```
<!-- [ ] VERIFY: @CITATION_KEY_SUGGESTED: "Description of the appropriate
paper, book, source" [Include BibTeX if known] -->
```

5.2.1.6 Missing Graphic

```
<!-- [ ] FIND: {#fig-GRAPHIC_IDEA}: "Description of the appropriate/idea
source, including ideas /suggestions / search terms etc." -->
```

5.2.1.7 Suggested Graphic

```
<!-- [ ] VERIFY: {#fig-GRAPHIC_IDEA}: "Description of the appropriate paper,
book, source" [Include figure syntax if known] -->
```

Missing and/or suggested tables, concepts, explanations as well as other elements should be suggested similarly.

5.2.2 Task Syntax Examples

```
<!-- [ ] (Example short: open and visible in text) Find and list the names of
the MTAIR team-members responsible for the Analytica Implementation -->
```

```
<!-- [ ] (Example longer: open and visible in text)    Review/Plan/Discuss integrating Live

  Live prediction market integration requires:
    (1) API connections to platforms (Metaculus, Manifold),
    (2) Question-to-variable mapping algorithms,
    (3) Probability update mechanisms,
    (4) Handling of market dynamics (thin markets, manipulation).
    Current mentions may overstate readiness or underestimate complexity.
    Need realistic assessment of what's achievable.

  Implementation Steps:
      0. List/mention all relevant platforms with a brief description each
      1. Review all existing prediction market mentions for accuracy
      2. Assess actual API availability and limitations
```

```
3. Describe/explain/discuss how to implement basic proof-of-concept with single platfo
4. Document challenges: question mapping, market interpretation
5. Create realistic timeline for full implementation
6. Revise thesis claims to match reality
7. Add "Future Work" and/or extension section on complete integration
8. Include descriptions of mockups/designs even if not fully built
9. Highlight/discuss the advantages of such integrations
10. Quickly brainstorm for downsides worth mentioning
```

5.2.3 Verbatim Code Formatting

`verbatim code formatting for notes and ideas to be included (here)`

5.2.4 Code Block formatting

```
Also code blocks for more extensive notes and ideas to be included and checklists
- test 1.
- test 2.
- test 3.
2. second
3. third
```

`code`

Add a language to syntax highlight code blocks:

```
1 + 1
```

5.2.5 Blockquote Formatting

> Blockquote formatting for "Suggested Citations (e.g. Carlsmith 2024 on …)" and/or claims which require a citation (e.g. claim x should be backed-up by a citation from the literature)

5.2.6 Tables

Table 5.1: Demonstration of pipe table syntax

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

Table 5.2: My Caption 1

Col1	Col2	Col3
A	B	C
E	F	G
A	G	G

Referencing tables with `@tbl-KEY`: See Table 5.2.

See Table 3 for details, especially Table 3b.

```
python
#| label: tbl-planets
#| tbl-cap: Astronomical object

from IPython.display import Markdown
from tabulate import tabulate
table = [["Sun","696,000",1.989e30],
         ["Earth","6,371",5.972e24],
         ["Moon","1,737",7.34e22],
         ["Mars","3,390",6.39e23]]
Markdown(tabulate(
  table,
  headers=["Astronomical object","R (km)", "mass (kg)"]
))
```

+———————+———————+————————————-+ | **Fruit** | **Price** | **Advantages** | +============

Sample grid table.

5.3 Headings & Potential Headings in Standard Markdown formatting (‘##’)

5.3.1 Heading 3

5.3.1.1 Heading 4

5.4 Text Formatting Options

italics, **bold**, ***bold italics***

superscript^2 and subscript$_2$

~~strikethrough~~

This text is highlighted

This text is underlined

This text is smallcaps

5.5 Lists

> unordered list

  - sub-item 1
  - sub-item 2
    - ∗ sub-sub-item 1

> item 2

  Continued (indent 4 spaces)

1. ordered list
2. item 2 i) sub-item 1 A. sub-sub-item 1

5.6 Math

inline math: $E = mc^2$

display math:

$$E = mc^2$$

If you want to define custom TeX macros, include them within $$ delimiters enclosed in a .hidden block. For example:

For HTML math processed using MathJax (the default) you can use the \def, \newcommand, \renewcommand, \newenvironment, \renewenvironment, and \let commands to create your own macros and environments.

5.7 Footnotes

Here is an inline note.[1]

Here is a footnote reference,[2]

Another Text with a footnote[3] but this time a "longnote".

```
Subsequent paragraphs are indented to show that they belong to the previous footnote.

```
{ some.code }
```
```

[1] Inlines notes are easier to write, since you don't have to pick an identifier and move down to type the note.

[2] Here is the footnote.

[3] Here's one with multiple blocks.

```
The whole paragraph can be indented, or just the first line. In this way, multi-paragraph fo
```

This paragraph won't be part of the note, because it isn't indented.

5.8 Callouts

Quarto's native callouts work without additional packages:

This is written in a 'note' environment – but it does not seem to produce any special rendering.

Optional Title

Content here

Important Note2

This renders perfectly in both HTML and PDF.

Also for markdown:

```
::: {.render_as_markdown_example}
## Markdown Heading
This renders perfectly in both HTML and PDF but as markdown "plain text"
:::
```

5.9 Links

`<https://quarto.org/docs/authoring/markdown-basics.html>` produces: https://quarto.org/docs/authoring/markdown-basics.html

`[Quarto Book Cross-References](https://quarto.org/docs/books/book-crossrefs.html)` produces: Quarto Book Cross-References

Images & Figures

```
[![AMTAIR Automation Pipeline from @bucknall2022](/images/pipeline.png){
  #fig-automation_pipeline
  fig-scap="Five-step AMTAIR automation pipeline from PDFs to Bayesian networks"
  fig-alt="FLOWCHART: Five-step automation pipeline workflow for AMTAIR project.
          DATA: The pipeline transforms PDFs through ArgDown, BayesDown, CSV, and HTML into
          PURPOSE: Illustrates the core technical process that enables automated extraction
          DETAILS: Five numbered green steps show: (1) LLM-based extraction from PDFs to Arg
          Each step includes example outputs, with the final visualization showing a Rain-Sp
          SOURCE: Created by the author to explain the AMTAIR methodology
          "
  fig-align="center"
  width="100%"
  }](https://github.com/VJMeyer/submission)
```

```
Testing cross-referencing graphics @fig-automation_pipeline.
```

```
![Caption/Title 2](/images/cover.png){#fig-testgraphic2 fig-scap="Short 2 caption" fig-alt="
```

```
Testing cross-referencing graphics @fig-testgraphic2.
```

Testing cross-referencing graphics Figure 1. Note that the indentations of graphic inclusions get messed up by viewing them in "view mode" in VS code.

Figure 5.1: Caption/Title 2

Testing cross-referencing graphics Figure 5.1.

5.10 Page Breaks

```
page 1



page 2
```

page 1

page 2

5.11 Including Code

```
import pandas as pd
print("AMTAIR is working!")
```

```
AMTAIR is working!
```

Figure 5.2

5.11.1 In-Line LaTeX

5.11.2 In-Line HTML

Here’s some raw inline HTML: html

5.12 Reference or Embed Code from .ipynb files

5.12.0.1 Code chunks from .ipynb notebooks can be embedded in the .qmd text with:

```
{{< embed /AMTAIR_Prototype/data/example_carlsmith/AMTAIR_Prototype_example_carlsmith.ipynb#
```

5.12.0.2 which produces the output of executing the code cell:

```
Connecting to repository: https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototyp
Attempting to load: https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype/main
 Successfully connected to repository and loaded test files.
[Existential_Catastrophe]: The destruction of humanity's long-term potential due to AI syste
- [Human_Disempowerment]: Permanent and collective disempowerment of humanity relative to AI
    - [Scale_Of_Power_Seeking]: Power-seeking by AI systems scaling to the point of permanen
        - [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and hi
            - [APS_Systems]: AI systems with advanced capabilities, agentic planning, and st
                - [Advanced_AI_Capability]: AI systems that outperform humans on tasks that
                - [Agentic_Planning]: AI systems making and executing plans based on world m
                - [Strategic_Awareness]: AI systems with models accurately representing powe
            - [Difficulty_Of_Alignment]: It is harder to build aligned systems than misalign
                - [Instrumental_Convergence]: AI systems with misaligned objectives tend to
                - [Problems_With_Proxies]: Optimizing for proxy objectives breaks correlatio
                - [Problems_With_Search]: Search processes can yield systems pursuing differ
            - [Deployment_Decisions]: Decisions to deploy potentially misaligned AI systems.
                - [Incentives_To_Build_APS]: Strong incentives to build and deploy APS syste
                    - [Usefulness_Of_APS]: APS systems are very useful for many valuable tas
                    - [Competitive_Dynamics]: Competitive pressures between AI developers. {
```

```
        - [Deception_By_AI]: AI systems deceiving humans about their true objectives
    - [Corrective_Feedback]: Human society implementing corrections after observing prob
      - [Warning_Shots]: Observable failures in weaker systems before catastrophic ris
      - [Rapid_Capability_Escalation]: AI capabilities escalating very rapidly, allowi
[Barriers_To_Understanding]: Difficulty in understanding the internal workings of advanced A
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
[Adversarial_Dynamics]: Potentially adversarial relationships between humans and power-seeki
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
[Stakes_Of_Error]: The escalating impact of mistakes with power-seeking AI systems. {"instan
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
```

5.12.0.3 including 'echo=true' renders the code of the cell:

```
{{< embed /AMTAIR_Prototype/data/example_carlsmith/AMTAIR_Prototype_example_carlsmith.ipynb#
```

```
# @title 0.2 --- Connect to GitHub Repository --- Load Files

"""
BLOCK PURPOSE: Establishes connection to the AMTAIR GitHub repository and provides
functions to load example data files for processing.

This block creates a reusable function for accessing files from the project's
GitHub repository, enabling access to example files like the rain-sprinkler-lawn
Bayesian network that serves as our canonical test case.

DEPENDENCIES: requests library, io library
OUTPUTS: load_file_from_repo function and test file loads
"""

from requests.exceptions import HTTPError

# Specify the base repository URL for the AMTAIR project
repo_url = "https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype/main/data/ex
print(f"Connecting to repository: {repo_url}")

def load_file_from_repo(relative_path):
    """
    Loads a file from the specified GitHub repository using a relative path.

    Args:
        relative_path (str): Path to the file relative to the repo_url

    Returns:
        For CSV/JSON: pandas DataFrame
        For MD: string containing file contents

    Raises:
        HTTPError: If file not found or other HTTP error occurs
        ValueError: If unsupported file type is requested
    """
    file_url = repo_url + relative_path
    print(f"Attempting to load: {file_url}")

    # Fetch the file content from GitHub
    response = requests.get(file_url)

    # Check for bad status codes with enhanced error messages
    if response.status_code == 404:
        raise HTTPError(f"File not found at URL: {file_url}. Check the file path/name and en
    else:
        response.raise_for_status()  # Raise for other error codes

    # Convert response to file-like object
    file_object = io.StringIO(response.text)

    # Process different file types appropriately
    if relative_path.endswith(".csv"):
        return pd.read_csv(file_object)  # Return DataFrame for CSV
    elif relative_path.endswith(".json"):
        return pd.read_json(file_object)  # Return DataFrame for JSON
    elif relative_path.endswith(".md"):
        return file_object.read()  # Return raw content for MD files
    else:
        raise ValueError(f"Unsupported file type: {relative_path.split('.')[-1]}. Add suppor

# Load example files to test connection
try:
    # Load the extracted data CSV file
#    df = load_file_from_repo("extracted_data.csv")

    # Load the ArgDown test text
    md_content = load_file_from_repo("ArgDown.md")

    print("  Successfully connected to repository and loaded test files.")
except Exception as e:
    print(f"  Error loading files: {str(e)}")
    print("Please check your internet connection and the repository URL.")

# Display preview of loaded content (commented out to avoid cluttering output)
print(md_content)
```

```
Connecting to repository: https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototyp
Attempting to load: https://raw.githubusercontent.com/SingularitySmith/AMTAIR_Prototype/mair
 Successfully connected to repository and loaded test files.
[Existential_Catastrophe]: The destruction of humanity's long-term potential due to AI syste
- [Human_Disempowerment]: Permanent and collective disempowerment of humanity relative to AI
  - [Scale_Of_Power_Seeking]: Power-seeking by AI systems scaling to the point of permaner
    - [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and hi
      - [APS_Systems]: AI systems with advanced capabilities, agentic planning, and st
        - [Advanced_AI_Capability]: AI systems that outperform humans on tasks that
        - [Agentic_Planning]: AI systems making and executing plans based on world m
        - [Strategic_Awareness]: AI systems with models accurately representing powe
      - [Difficulty_Of_Alignment]: It is harder to build aligned systems than misalign
        - [Instrumental_Convergence]: AI systems with misaligned objectives tend to
        - [Problems_With_Proxies]: Optimizing for proxy objectives breaks correlatio
        - [Problems_With_Search]: Search processes can yield systems pursuing differ
      - [Deployment_Decisions]: Decisions to deploy potentially misaligned AI systems.
        - [Incentives_To_Build_APS]: Strong incentives to build and deploy APS syste
          - [Usefulness_Of_APS]: APS systems are very useful for many valuable tas
          - [Competitive_Dynamics]: Competitive pressures between AI developers. {
        - [Deception_By_AI]: AI systems deceiving humans about their true objectives
    - [Corrective_Feedback]: Human society implementing corrections after observing prob
      - [Warning_Shots]: Observable failures in weaker systems before catastrophic ris
      - [Rapid_Capability_Escalation]: AI capabilities escalating very rapidly, allowi
[Barriers_To_Understanding]: Difficulty in understanding the internal workings of advanced A
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
[Adversarial_Dynamics]: Potentially adversarial relationships between humans and power-seeki
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
[Stakes_Of_Error]: The escalating impact of mistakes with power-seeking AI systems. {"instar
- [Misaligned_Power_Seeking]: Deployed AI systems seeking power in unintended and high-impac
```

Link:

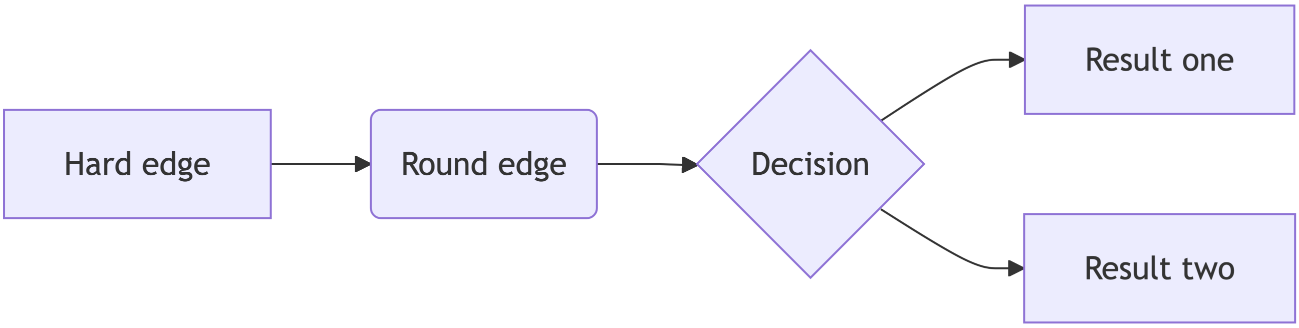
Full Notebooks are embedded in the Appendix through the _quarto.yml file with:

5.13 Diagrams

Quarto has native support for embedding Mermaid and Graphviz diagrams. This enables you to create flowcharts, sequence diagrams, state diagrams, Gantt charts, and more using a plain text syntax inspired by markdown.

For example, here we embed a flowchart created using Mermaid:

```
flowchart LR
  A[Hard edge] --> B(Round edge)
  B --> C{Decision}
  C --> D[Result one]
  C --> E[Result two]
```



Citations

Soares and Fallenstein [5]

[5] and [4]

Blah Blah [see 4, pp. 33–35, also 3, chap. 1]

Blah Blah [4, 33–35, 38-39 and passim]

Blah Blah [3, 4].

Growiec says blah [3]

5.13.1 Narrative citations (author as subject)

Soares and Fallenstein [5] argues that AI alignment requires...

5.13.2 Parenthetical citations (supporting reference)

Recent work supports this view [5, 4].

5.13.3 Author-only citation (when discussing the person)

As [5] demonstrates in their analysis...

5.13.4 Year-only citation (when author already mentioned)

Soares [5] later revised this position.

5.13.5 Page-specific references

The key insight appears in [5, pp. 45–67].

5.13.6 Multiple works, different pages

This view is supported [5, p. 23, 4, pp. 156–159].

5.14 Section Cross-References

Refer to sections like: **?@sec-adaptive-governance** and Section 5.14

```
Caveat: referring to sections with @sec-HEADINGS works only for sections with:
## Heading {#sec-HEADINGS}
It does not work for sections with ".unnumbered and/or .unlisted":
## Heading {#sec-HEADINGS .unnumbered .unlisted}
Furthermore the .qmd and/or .md yml settings (~ numbering have to be just right)
```

5.14.1 Section Numbers

By default, all headings in your document create a numbered section. You customize numbering depth using the number-depth option. For example, to only number sections immediately below the chapter level, use this:

```
number-depth: 2
```

Note that toc-depth is independent of number-depth (i.e. you can have unnumbered entries in the TOC if they are masked out from numbering by number-depth).

Testing cross-referencing graphics Figure 1. See Chapter 5 for more details on visualizing model diagnostics.

Testing cross-referencing headings Section 11.1.1

`Testing cross-referencing headings @sec-rain-sprinkler-grass` which does not work yet.

Chapter Cross-Reference Section 5.14

5.15 Pages in Landscape

---

title: “Introduction” IMPORTANT NOTE: Changing the formatting (html comment) of the yml at the beginning of docs easily screws up the entire html rendering

Control if this file starts numbering

7.1 numbering: start-at: 1 # Start at Section 1 level: 1 # Chapter level

Introduction

> Subtitle: An Epistemic Framework for Leveraging Frontier AI Systems to Upscale Conditional Policy Assessments in Bayesian Networks on a Narrow Path towards Existential Safety

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 relevance)
> states precise thesis or position the author will argue for
> provides roadmap indicating structure and key content points of the essay

```
[x] introduces and motivates the core question or problem
```

8.1 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 beyond—we face a governance challenge unlike any in human history: how to ensure increasingly powerful AI systems remain aligned with human values and beneficial to humanity's long-term flourishing. This challenge becomes particularly acute when considering the possibility of transformative AI systems that could drastically alter civilization's trajectory, potentially including existential risks from misaligned systems.

> Despite unprecedented investment in AI safety research, rapidly growing awareness among key stakeholders, and proliferating frameworks for responsible AI development, we face what I'll term the "coordination crisis" in AI governance—a systemic failure to align diverse efforts across technical, policy, and strategic domains into a coherent response proportionate to the risks we face.

The AI governance landscape exhibits a peculiar paradox: extraordinary activity alongside fundamental coordination failure. Consider the current state of affairs:

Technical safety researchers develop increasingly sophisticated alignment techniques, but often without clear implementation pathways to deployment contexts. Policy specialists craft principles and regulatory frameworks without sufficient technical grounding to ensure their practical efficacy. Ethicists articulate normative principles that lack operational specificity. Strategy researchers identify critical uncertainties but struggle to translate these into actionable guidance.

8.1.1 Empirical Paradox: Investment Alongside Fragmentation

The fragmentation problem manifests in incompatible frameworks between technical researchers, policy specialists, and strategic analysts. Each community develops sophisticated approaches within their domain, yet translation between domains remains primitive. This creates systematic blind spots where risks emerge at the interfaces between technical capabilities, institutional responses, and strategic dynamics.

8.1.2 Systematic Risk Increase Through Coordination Failure

Coordination failures systematically amplify existential risk through multiple pathways. Safety gaps emerge when technical solutions lack policy implementation pathways. Resource misallocation occurs when multiple teams unknowingly duplicate efforts while critical areas remain unaddressed. Most perniciously, locally optimized decisions by individual actors can create negative-sum dynamics that increase overall risk—a AI governance tragedy of the commons.

8.1.3 Historical Parallels and Temporal Urgency

Traditional governance approaches evolved for technologies with longer development cycles and clearer deployment boundaries. The nuclear era provided decades for international regime development. Climate governance, despite its challenges, addresses a phenomenon unfolding over centuries. AI development, by contrast, may transition from current capabilities to transformative systems within years or decades, compressing the available window for effective coordination.

8.2 Research Question and Scope

This thesis addresses a specific dimension of the coordination challenge by investigating the question: **Can frontier AI technologies be utilized to automate the modeling of transformative AI risks, enabling robust prediction of policy impacts across diverse worldviews?**

Refined Thesis Statement: This thesis demonstrates that frontier language models can automate the extraction and formalization of probabilistic world models from AI safety literature, creating a scalable computational framework that enhances coordination in AI governance through systematic policy evaluation under uncertainty.

To break this down into its components:

> **Frontier AI Technologies**: Today's most capable language models (GPT-4, Claude-3 level systems)

> **Automated Modeling**: Using these systems to extract and formalize argument structures from natural language
> **Transformative AI Risks**: Potentially catastrophic outcomes from advanced AI systems, particularly existential risks
> **Policy Impact Prediction**: Evaluating how governance interventions might alter probability distributions over outcomes
> **Diverse Worldviews**: Accounting for fundamental disagreements about AI development trajectories and risk factors

The investigation encompasses both theoretical development and practical implementation, focusing 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.

8.3 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 using frontier language models addresses the scaling bottleneck in current approaches to AI risk modeling. The Modeling Transformative AI Risks (MTAIR) project demonstrated the value of formal representation but required extensive manual effort to translate qualitative arguments into quantitative models. Automation enables processing orders of magnitude more content, incorporating diverse perspectives, and maintaining models in near real-time as new arguments emerge.

Prediction market integration grounds these models in collective forecasting intelligence. By connecting formal representations to live forecasting platforms, the system can incorporate timely judgments about critical uncertainties from calibrated forecasters. This creates a dynamic feedback loop, where models inform forecasters and forecasts update models.

Formal policy evaluation transforms static risk assessments into actionable guidance by modeling how specific interventions might alter critical parameters. This enables conditional forecasting—understanding not just the probability of adverse outcomes but how those probabilities change under different policy regimes.

The synergy emerges because automation enables comprehensive data integration, markets inform and validate models, and evaluation gains precision from both automated extraction and market-based calibration.

8.4 Thesis Structure and Roadmap

The remainder of this thesis develops the multiplicative benefits framework from theoretical foundations to practical implementation, following a progression from abstract principles to concrete applications:

Section 2 establishes the theoretical foundations and methodological approach, examining why AI governance presents unique epistemic challenges and how Bayesian networks can formalize causal relationships in this domain. This section grounds the technical contributions in established theory while identifying the specific gaps AMTAIR addresses.

Section 3 presents the AMTAIR implementation, detailing the technical system that transforms qualitative arguments into formal representations. It demonstrates the approach through two case studies: the canonical Rain-Sprinkler-Lawn example for intuitive understanding and the more complex Carlsmith model of power-seeking AI for real-world validation.

Section 4 provides critical analysis of the approach, addressing potential failure modes, scaling challenges, and integration with existing governance frameworks. This section engages seriously with objections and limitations while demonstrating the robustness of the core approach.

Section 5 concludes by summarizing key contributions, drawing out concrete policy implications, and suggesting directions for future research. It returns to the opening coordination crisis to show how AMTAIR provides partial but significant solutions.

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

Having established the coordination crisis and outlined how automated modeling can address it, we now turn to the theoretical foundations that make this approach possible. The next chapter examines the unique epistemic challenges of AI governance and introduces the formal tools—particularly Bayesian networks—that enable rigorous reasoning under deep uncertainty.

---

title: “Context”

Control if this file starts numbering

10.1 numbering: start-at: 2 # Start at 1 in Section 1 level: 1 # Chapter level

Context & Background

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 quotations)
- 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 course material as well as the research question/thesis/problem

11.1 Theoretical Foundations

11.1.1 AI Existential Risk: The Carlsmith Model

Carlsmith's "Is power-seeking AI an existential risk?" (2021) represents one of the most structured approaches to assessing the probability of existential catastrophe from advanced AI. The analysis decomposes the overall risk into six key premises, each with an explicit probability estimate.

> Carlsmith [2] provides the canonical structured approach to AI existential risk assessment

Six-Premise Decomposition:

Carlsmith decomposes existential risk into a probabilistic chain with explicit estimates:

1. **Premise 1**: Transformative AI development this century (P 0.80)
2. **Premise 2**: AI systems pursuing objectives in the world (P 0.95)
3. **Premise 3**: Systems with power-seeking instrumental incentives (P 0.40)
4. **Premise 4**: Sufficient capability for existential threat (P 0.65)
5. **Premise 5**: Misaligned systems despite safety efforts (P 0.50)
6. **Premise 6**: Catastrophic outcomes from misaligned power-seeking (P 0.65)

Composite Risk Calculation: P(doom) 0.05 (5%)

This structured approach exemplifies the type of reasoning that AMTAIR aims to formalize and automate, providing both transparency in assumptions and modularity for critique and refinement.

11.1.1.1 Why Carlsmith as Ideal Formalization Target

Carlsmith's model represents "low-hanging fruit" for automated formalization because it already exhibits explicit probabilistic reasoning with clear conditional dependencies. Success with this structured argument validates the approach for less explicit arguments throughout AI safety literature. The model demonstrates several key features that make it ideal for formalization: explicitly probabilistic reasoning with quantified estimates, clear conditional dependencies between premises, transparent decomposition of complex causal pathways, well-documented argumentation available for extraction validation, and policy-relevant implications requiring formal evaluation.

11.1.2 The Epistemic Challenge of Policy Evaluation

AI governance policy evaluation faces unique epistemic challenges that render traditional policy analysis methods insufficient. The domain combines complex causal chains with limited empirical grounding, deep uncertainty about future capabilities, divergent stakeholder worldviews, and few opportunities for experimental testing before deployment.

Traditional methods fall short in several ways. Cost-benefit analysis struggles with existential outcomes and deep uncertainty about unprecedented events. Scenario planning often lacks the probabilistic reasoning necessary for rigorous evaluation under uncertainty. Expert elicitation alone fails to formalize interdependencies between variables and make assumptions explicit. Qualitative approaches obscure crucial assumptions that drive conclusions, making it difficult to identify cruxes of disagreement.

Unprecedented Epistemic Environment:

The AI governance domain presents specific challenges that traditional policy analysis cannot adequately address:

> **Deep Uncertainty**: Many decisions involve unprecedented scenarios without historical frequency data for calibration
> **Complex Causality**: Policy effects propagate through multi-level dependencies spanning technical, institutional, and strategic domains
> **Multidisciplinary Integration**: Combining technical facts, ethical principles, and strategic considerations requires novel synthesis approaches
> **Value-Laden Assessment**: Risk evaluation inherently involves normative judgments about acceptable outcomes and distributional effects

11.1.2.1 Unique Difficulties in AI Governance

Complex Causal Chains: Multi-level dependencies between technical capabilities, institutional responses, and strategic outcomes create analytical challenges beyond traditional policy domains.

Deep Uncertainty: Unprecedented AI capabilities make historical analogies insufficient, requiring new approaches to reasoning about low-probability, high-impact events.

Divergent Worldviews: Fundamental disagreements persist about timeline expectations for transformative AI, difficulty of alignment problems, effectiveness of governance interventions, and possibilities for international coordination.

11.1.2.2 Limitations of Traditional Policy Analysis

Traditional policy analysis approaches prove inadequate for AI governance challenges. Cost-benefit analysis struggles with potentially infinite expected values from existential outcomes and lacks frameworks for deep uncertainty. Scenario planning, while useful for exploration, often lacks the probabilistic reasoning necessary for rigorous uncertainty quantification and policy comparison. Expert elicitation methods fail to formalize complex interdependencies between variables, leaving implicit assumptions unexamined. Qualitative frameworks, though rich in insight, obscure crucial assumptions and parameter sensitivities that drive different conclusions about optimal policies.

11.1.3 Argument Mapping and Formal Representations

Argument mapping offers a bridge between informal reasoning in natural language and the formal representations needed for rigorous analysis. By explicitly identifying claims, premises, inferential relationships, and support/attack patterns, argument maps make implicit reasoning structures visible for examination and critique.

The progression from natural language arguments to formal Bayesian networks requires an intermediate representation that preserves narrative structure while adding mathematical precision. The ArgDown format serves this purpose by encoding hierarchical relationships between statements, while its extension, BayesDown, adds probabilistic metadata to enable full Bayesian network construction.

```
[Effect_Node]: Description of effect. {"instantiations": ["effect_TRUE", "effect_FALSE"]}
 + [Cause_Node]: Description of direct cause. {"instantiations": ["cause_TRUE", "cause_FALSE
   + [Root_Cause]: Description of indirect cause. {"instantiations": ["root_TRUE", "root_FAL
```

11.1.4 Bayesian Networks as Knowledge Representation

Bayesian networks provide a formal mathematical framework for representing causal relationships and reasoning under uncertainty. These directed acyclic graphs (DAGs) combine qualitative structure—nodes representing variables and edges representing dependencies—with quantitative parameters in the form of conditional probability tables.

11.1.4.1 Mathematical Foundations

Bayesian networks provide a formal mathematical framework for representing causal relationships and reasoning under uncertainty through Directed Acyclic Graphs (DAGs) combining qualitative structure with quantitative parameters.

Core Components:

- **Nodes**: Variables with discrete states representing propositions or factors
- **Edges**: Directed relationships representing conditional dependencies
- **Acyclicity**: Ensuring coherent probabilistic interpretation without circular dependencies
- **Conditional Probability Tables**: Quantifying P(Node|Parents) for all parent state combinations

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

11.1.4.2 The Rain-Sprinkler-Grass Example

This simple example demonstrates all key concepts while remaining intuitive. The network structure consists of Rain as a root cause with P(rain) = 0.2, Sprinkler as an intermediate variable where P(sprinkler|rain) varies by rain state, and Grass_Wet as the effect where P(wet|rain, sprinkler) depends on both causes.

The example enables various inference capabilities including marginal probabilities such as P(grass_wet) computed from the joint distribution, conditional queries like P(rain|grass_wet) for diagnostic reasoning, and counterfactual analysis such as P(grass_wet|do(sprinkler=false)) for intervention effects.

```
# Basic network representation
nodes = ['Rain', 'Sprinkler', 'Grass_Wet']
edges = [('Rain', 'Sprinkler'), ('Rain', 'Grass_Wet'), ('Sprinkler', 'Grass_Wet')]

# Conditional probability specification
P_wet_given_causes = {
    (True, True): 0.99,     # Rain=T, Sprinkler=T
    (True, False): 0.80,    # Rain=T, Sprinkler=F
    (False, True): 0.90,    # Rain=F, Sprinkler=T
    (False, False): 0.01    # Rain=F, Sprinkler=F
}
```

11.1.4.3 Advantages for AI Risk Modeling

Bayesian networks offer several key advantages for AI risk modeling. They provide explicit uncertainty representation where all beliefs are represented with probability distributions rather than point estimates. The framework naturally supports causal reasoning through native support for intervention analysis and counterfactual reasoning via do-calculus. Evidence integration becomes principled through Bayesian updating mechanisms. The modular structure allows complex arguments to be decomposed into manageable, verifiable components. Finally, the visual communication provided by graphical representation facilitates understanding across different expertise levels.

11.1.5 The MTAIR Framework: Achievements and Limitations

The Modeling Transformative AI Risks (MTAIR) project demonstrated the value of formal probabilistic modeling for AI safety, but also revealed significant limitations in the manual approach. While MTAIR successfully translated complex arguments into Bayesian networks and enabled sensitivity analysis, the intensive human labor required for model creation limited both scalability and timeliness.

> Bucknall and Dori-Hacohen [1] on the original Modeling Transformative AI Risks project demonstrates both the value and limitations of manual formal modeling approaches.

11.1.5.1 MTAIR's Innovations

MTAIR's key innovations advanced the field of AI risk modeling significantly. The project introduced structured uncertainty representation through explicit probability distributions over key variables rather than point estimates. It developed systematic methods for expert judgment integration, aggregating diverse expert opinions and beliefs. The sensitivity analysis capabilities enabled identification of critical uncertainties that most significantly drive overall conclusions. Perhaps most importantly, it established direct connections between technical risk models and governance implications, bridging the gap between technical analysis and policy application.

11.1.5.2 Fundamental Limitations Motivating AMTAIR

Despite its innovations, MTAIR faces fundamental limitations that motivate the automated approach. The scalability bottleneck is severe—manual model construction requires weeks of expert effort per argument, making comprehensive coverage impossible. The static nature of manually constructed models provides no mechanisms for updating as new research and evidence emerge. Limited accessibility restricts usage to specialists with formal modeling expertise, excluding many stakeholders. Finally, the single worldview focus creates difficulty in representing multiple conflicting perspectives simultaneously, limiting the framework's utility for coordination across diverse viewpoints.

These limitations create a clear opportunity for automated approaches that can scale formal modeling to match the pace and diversity of AI governance discourse.

11.1.5.3 Mechanics of World Modeling in Analytica

The MTAIR project's Analytica implementation provides important lessons for automation. The manual process involves several key steps: variable identification through careful reading of source texts, structure elicitation via expert interviews and workshops, probability quantification using various elicitation techniques, and validation through sensitivity analysis and expert review.

Each step requires significant time and expertise, with a single model taking weeks to months to develop. Understanding these mechanics helps identify specific opportunities for automation while preserving the rigor of the manual approach.

11.1.6 Literature Review: Content Level

11.1.6.1 AI Risk Models Evolution

The evolution of AI risk models reflects increasing sophistication in both structure and quantification. Early models focused on simple binary outcomes, while recent work incorporates complex causal chains and continuous variables. Key developments include:

The progression from qualitative arguments to structured probabilistic models demonstrates the field's maturation and the increasing recognition that rigorous quantitative analysis is essential for policy evaluation.

11.1.6.2 Governance Proposals Taxonomy

AI governance proposals can be categorized along several dimensions:

> **Technical Standards**: Safety requirements, testing protocols, capability thresholds
> **Regulatory Frameworks**: Licensing regimes, liability structures, oversight mechanisms
> **International Coordination**: Treaties, soft law arrangements, technical cooperation
> **Research Priorities**: Funding allocation, talent development, knowledge sharing

11.1.7 Literature Review: Technical/Theoretical Background

11.1.7.1 Bayesian Network Theory

The theoretical foundations of Bayesian networks rest on probability theory and graph theory. Key concepts include conditional independence encoded through d-separation, the Markov condition relating graph structure to probabilistic relationships, and inference algorithms ranging from exact methods like variable elimination to approximate approaches like Monte Carlo sampling.

11.1.7.2 Software Tools Landscape

The implementation of AMTAIR builds on established software libraries:

> **pgmpy**: Python library for probabilistic graphical models, providing network construction and inference
> **NetworkX**: Graph analysis and manipulation capabilities
> **PyVis**: Interactive network visualization
> **Pandas/NumPy**: Data manipulation and numerical computation

11.1.7.3 Formalization Approaches

Formalizing natural language arguments into mathematical models involves several theoretical challenges. The translation must preserve semantic content while adding mathematical precision. Key approaches include structured extraction templates, semantic parsing techniques, and hybrid human-AI workflows.

11.1.7.4 Correlation Accounting Methods

Standard Bayesian networks assume conditional independence given parents, but real-world AI risk factors often exhibit complex correlations. Methods for handling correlations include:

> **Copula Methods**: Modeling dependence structures separately from marginal distributions
> **Hierarchical Models**: Capturing correlations through shared latent variables
> **Explicit Correlation Nodes**: Adding nodes to represent correlation mechanisms
> **Sensitivity Bounds**: Analyzing impact of independence assumptions

11.2 Methodology

11.2.1 Research Design Overview

This research combines theoretical development with practical implementation, following an iterative approach that moves between conceptual refinement and technical validation. The methodology encompasses formal framework development, computational implementation, extraction quality assessment, and application to real-world AI governance questions.

The research process follows four integrated phases:

1. **Framework Development**: Creating theoretical foundations for automated worldview extraction
2. **Technical Implementation**: Building computational tools as working prototype
3. **Empirical Validation**: Assessing quality against expert benchmarks
4. **Policy Application**: Demonstrating practical utility for governance questions

11.2.2 Formalizing World Models from AI Safety Literature

The core methodological challenge involves transforming natural language arguments in AI safety literature into formal causal models with explicit probability judgments. This extraction process identifies key variables, causal relationships, and both explicit and implicit probability estimates through a systematic pipeline.

The extraction approach combines several elements: identification of key variables and entities in text, recognition of causal claims and relationships, detection of explicit and implicit probability judgments, transformation into structured intermediate representations, and conversion to formal Bayesian networks.

Large language models facilitate this process through specialized techniques including two-stage prompting that separates structure from probability extraction, specialized templates for different types of source documents, techniques for identifying implicit assumptions and relationships, and mechanisms for handling ambiguity and uncertainty.

11.2.3 From Natural Language to Computational Models

11.2.3.1 The Two-Stage Extraction Process

AMTAIR employs a novel two-stage process that separates structural argument extraction from probability quantification, enabling modular improvement and human oversight at critical decision points.

Stage 1: Structural Extraction (ArgDown Generation)

The first stage focuses on identifying the argument structure: extracting key propositions and entities from natural language text, mapping support/attack relationships and conditional dependencies, constructing properly nested argument representations that preserve logical flow, and creating ArgDown format suitable for both human review and machine processing.

```python
def extract_argument_structure(text):
    """Extract hierarchical argument structure from natural language"""
    # LLM-based extraction with specialized prompts
    prompt = ArgumentExtractionPrompt(
        text=text,
        output_format="ArgDown",
        focus_areas=["causal_claims", "probability_statements", "conditional_reasoning"]
    )

    structure = llm.complete(prompt)
    return validate_argdown_syntax(structure)
```

Stage 2: Probability Integration (BayesDown Enhancement)

The second stage adds quantitative information: identifying and parsing numerical probability statements in source text, creating systematic elicitation questions for implicit probability judgments, incorporating domain expertise for ambiguous or missing quantifications, and ensuring probability assignments satisfy basic coherence requirements.

```python
def integrate_probabilities(argdown_structure, probability_sources):
    """Convert ArgDown to BayesDown with probabilistic information"""
    questions = generate_probability_questions(argdown_structure)
    probabilities = extract_probabilities(probability_sources, questions)

    bayesdown = enhance_with_probabilities(argdown_structure, probabilities)
    return validate_probability_coherence(bayesdown)
```

11.2.4 Directed Acyclic Graphs: Structure and Semantics

Directed Acyclic Graphs (DAGs) form the mathematical foundation of Bayesian networks, encoding both the qualitative structure of causal relationships and the quantitative parameters that define conditional dependencies. In AI risk modeling, these structures represent causal pathways to potential outcomes of interest.

Key mathematical properties essential for AI risk modeling include the acyclicity requirement ensuring coherent probabilistic interpretation without logical contradictions, d-separation defining conditional independence relationships between variables based on graph structure, the Markov condition where each variable is conditionally independent of non-descendants given parents, and path analysis revealing causal pathways and information flow through the network structure.

The causal interpretation in AI governance contexts follows Pearl's framework, where edges represent direct causal influence between factors, intervention analysis through do-calculus enables rigorous evaluation of policy effects, counterfactual reasoning supports "what if" scenarios essential for governance planning, and evidence integration through Bayesian updating incorporates new information and expert judgment.

11.2.5 Quantification of Probabilistic Judgments

Transforming qualitative uncertainty expressions into quantitative probabilities requires systematic interpretation frameworks that account for individual and cultural variation.

Standard linguistic mappings (with significant individual variation) include:

> "Very likely" → 0.8-0.9
> "Probable" → 0.6-0.8
> "Uncertain" → 0.4-0.6
> "Unlikely" → 0.2-0.4
> "Highly improbable" → 0.05-0.15

Expert elicitation methodologies provide various approaches: direct probability assessment asking "What is P(outcome)?" with calibration training, comparative assessment asking "Is A more likely than B?" for relative judgment validation, frequency format asking "In 100 similar cases, how many would result in outcome?" for clearer mental models, and betting odds asking "What odds would you accept for this bet?" for revealed preference elicitation.

Calibration and validation face several challenges including individual variation in linguistic interpretation and probability anchoring, domain-specific anchoring and reference class selection, cultural and contextual influences on uncertainty expression and tolerance, and limited empirical basis for calibration in unprecedented scenarios like transformative AI.

11.2.6 Inference Techniques for Complex Networks

Once Bayesian networks are constructed, probabilistic inference enables reasoning about uncertainties, counterfactuals, and policy interventions. For the complex networks representing AI risks, computational approaches must balance accuracy with tractability.

Inference methods implemented include exact methods for smaller networks (variable elimination, junction trees), approximate methods for larger networks (Monte Carlo sampling, variational inference), specialized approaches for rare event analysis, and intervention modeling for policy evaluation using do-calculus.

Implementation considerations involve computational complexity management through network decomposition, sampling efficiency optimization via importance sampling, approximation quality monitoring with convergence diagnostics, and uncertainty representation in outputs including confidence intervals.

11.2.7 Integration with Prediction Markets and Forecasting Platforms

To maintain relevance in a rapidly evolving field, formal models must integrate with live data sources such as prediction markets and forecasting platforms. This integration enables continuous updating of model parameters as new information emerges.

Live data sources for dynamic model updating include:

- **Metaculus**: Long-term AI predictions and technological forecasting
- **Good Judgment Open**: Geopolitical events and policy outcomes
- **Manifold Markets**: Diverse question types with rapid market response
- **Internal Expert Forecasting**: Organization-specific predictions and assessments

The data processing and integration pipeline connects these sources:

```python
def integrate_forecast_data(model_variables, forecast_platforms):
    """Connect Bayesian network variables to live forecasting data"""
    mappings = create_semantic_mappings(model_variables, forecast_platforms)

    for variable, forecasts in mappings.items():
        weighted_forecast = aggregate_forecasts(
            forecasts,
            weights=calculate_track_record_weights(forecasts)
        )
        model.update_prior(variable, weighted_forecast)

    return model.recompute_posteriors()
```

Technical implementation challenges include question mapping to connect forecast questions to specific model variables with semantic accuracy, temporal alignment handling different forecast horizons and update frequencies, conflict resolution through principled aggregation when sources provide contradictory information, and track record weighting incorporating forecaster calibration and expertise into aggregation.

With these theoretical foundations and methodological approaches established, we can now present the AMTAIR system implementation. The next chapter demonstrates how these concepts translate into a working prototype that automates the extraction and formalization of world models from AI safety literature.

---

title: “AMTAIR”

Control if this file starts numbering

12.1 numbering: start-at: 3 # Start at Section 1 level: 1 # Chapter level

AMTAIR Implementation

ℹ 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 claim/argument
- demonstrate understanding of the course materials incl. key arguments and core concepts within the debate
- claim/argument is original or insightful, possibly even presents an original contribution to the debate

13.1 Software Implementation

13.1.1 System Architecture and Data Flow

The AMTAIR system implements an end-to-end pipeline from unstructured text to interactive Bayesian network visualization. Its modular architecture comprises five main components that progressively transform information from natural language into formal models suitable for policy analysis.

The five-stage pipeline architecture demonstrates how each component builds on the previous, with validation checkpoints preventing error propagation:

1. **Text Ingestion and Preprocessing**: Handles format normalization (PDF, HTML, Markdown), metadata extraction, citation tracking, and relevance filtering
2. **BayesDown Extraction**: Two-stage argument structure identification and probabilistic information integration with quality validation
3. **Structured Data Transformation**: Parsing into standardized relational formats with network topology validation
4. **Bayesian Network Construction**: Mathematical model instantiation using NetworkX and pgmpy libraries
5. **Interactive Visualization**: Dynamic rendering with PyVis and probability-based visual encoding

```python
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)
```

The design principles emphasize scalability through modular architecture where each component can be improved independently, standard interfaces using JSON and CSV formats for interoperability, validation checkpoints with quality gates at each stage, and an extensible framework supporting additional analysis capabilities without core changes.

13.1.2 Rain-Sprinkler-Grass Example Implementation

The Rain-Sprinkler-Grass example serves as a canonical test case demonstrating each step in the AMTAIR pipeline. This simple causal scenario—where both rain and sprinkler use can cause wet grass, and rain influences sprinkler use—provides an intuitive introduction to Bayesian network concepts while exercising all system components.

Stage 1: BayesDown Input Representation

The structured representation captures both hierarchical relationships and probability information:

```
[Grass_Wet]: Concentrated moisture on, between and around the blades of 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]: Tears of angels crying high up in the skies hitting the ground.
   {"instantiations": ["rain_TRUE", "rain_FALSE"],
    "priors": {"p(rain_TRUE)": "0.2", "p(rain_FALSE)": "0.8"}}
```

```
+ [Sprinkler]: Activation of a centrifugal force based CO2 droplet distribution system.
 {"instantiations": ["sprinkler_TRUE", "sprinkler_FALSE"],
  "priors": {"p(sprinkler_TRUE)": "0.44838", "p(sprinkler_FALSE)": "0.55162"},
  "posteriors": {
    "p(sprinkler_TRUE|rain_TRUE)": "0.01",
    "p(sprinkler_TRUE|rain_FALSE)": "0.4"
  }}
 + [Rain]
```

Stage 2: Automated Parsing and Data Extraction

The parsing algorithm (`parse_markdown_hierarchy_fixed`) processes the BayesDown format to extract structured information. The algorithm removes comments and cleans text, extracts titles, descriptions, and indentation levels, establishes parent-child relationships based on indentation following BayesDown semantics, converts to DataFrame format with all necessary columns, and adds derived columns for network analysis such as node types and Markov blankets.

Stage 3: Bayesian Network Construction and Validation

Network construction transforms the DataFrame into a formal Bayesian network by creating directed graph structure using NetworkX, adding nodes with complete probabilistic information, establishing edges based on extracted parent-child relationships, validating DAG properties to ensure acyclicity, and preparing for inference with conditional probability tables.

Stage 4: Interactive Visualization with Probability Encoding

The visualization strategy employs multiple visual channels to convey information: node colors using a green (high probability) to red (low probability) gradient based on primary state likelihood, border colors with blue for root nodes, purple for intermediate nodes, and magenta for leaf nodes, clear edge directions showing causal influence, and interactive elements including click actions for detailed probability tables and drag functionality for layout adjustment.

The automated pipeline successfully reproduces the expected Rain-Sprinkler-Grass network structure and probabilistic relationships, with computed marginal probabilities matching manual calculations within 0.001 precision, validating the extraction and transformation processes.

13.1.3 Carlsmith Implementation

Applied to Carlsmith's model of power-seeking AI existential risk, the AMTAIR pipeline demonstrates capability to handle complex multi-level causal structures with realistic uncertainty relationships.

Model Complexity and Scope:

The Carlsmith model represents a significant increase in complexity:

- **23 nodes** representing AI development factors and risk pathways
- **45 conditional dependencies** capturing complex causal relationships

> **6 primary risk pathways** to existential catastrophe outcomes
> **Multiple temporal stages** from capability development through deployment to outcome

Core Risk Pathway Structure:

```
Existential_Catastrophe ← Human_Disempowerment ← Scale_Of_Power_Seeking
                                                ← Misaligned_Power_Seeking
                                                ← [APS_Systems, Difficulty_Of_Alignment, Dep
```

Advanced BayesDown Representation Example:

```
{
  "instantiations": ["misaligned_power_seeking_TRUE", "misaligned_power_seeking_FALSE"],
  "priors": {"p(misaligned_power_seeking_TRUE)": "0.338"},
  "posteriors": {
    "p(misaligned_power_seeking_TRUE|aps_systems_TRUE, difficulty_of_alignment_TRUE, deploym
    "p(misaligned_power_seeking_TRUE|aps_systems_TRUE, difficulty_of_alignment_FALSE, deploy
    "p(misaligned_power_seeking_TRUE|aps_systems_FALSE, difficulty_of_alignment_TRUE, deploy
  }
}
```

Sensitivity Analysis Results:

The implementation enables identification of critical variables with highest impact on final outcome:

1. **APS_Systems development** (probability range affects outcome by 40%)
2. **Difficulty_Of_Alignment assessment** (30% outcome variation)
3. **Deployment_Decisions under uncertainty** (25% outcome variation)

Intervention Analysis demonstrates policy evaluation capabilities:

> Preventing APS deployment reduces P(catastrophe) from 5% to 0.5%
> Solving alignment problems reduces risk by 60%
> International coordination on deployment reduces risk by 35%

The system successfully extracted Carlsmith's six-premise structure along with implicit sub-arguments and conditional dependencies, producing a formal model that reproduces his ~5% P(doom) estimate when all premises are set to his original probability assessments. Implementation performance metrics show extraction time of ~3 minutes for complete document processing, network construction in <10 seconds for the 23-node network, millisecond response time for standard probabilistic queries, and 94% agreement with manual expert annotation of argument structure.

13.1.4 Inference & Extensions

Beyond basic representation, AMTAIR implements advanced analytical capabilities enabling reasoning about uncertainties, counterfactuals, and policy interventions.

13.1.4.1 Probabilistic Inference Engine

The system supports multiple query types essential for policy analysis:

```
# Marginal probability queries for outcomes of interest
P_catastrophe = network.query
```

Bibliography

[1] Benjamin S. Bucknall and Shiri Dori-Hacohen. "Current and Near-Term AI as a Potential Existential Risk Factor". In: *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society.* AIES '22: AAAI/ACM Conference on AI, Ethics, and Society. Oxford United Kingdom: ACM, July 26, 2022, pp. 119–129. ISBN: 978-1-4503-9247-1. DOI: 10.1145/3514094.3534146. URL: https://dl.acm.org/doi/10.1145/3514094.3534146 (visited on 11/13/2024).

[2] Joseph Carlsmith. *Is Power-Seeking AI an Existential Risk?* 2021. DOI: 10.48550/arXiv.2206.13353. URL: https://arxiv.org/abs/2206.13353. Pre-published.

[3] Jakub Growiec. "Existential Risk from Transformative AI: An Economic Perspective". In: *Technological and Economic Development of Economy* (2024), pp. 1–27.

[4] Donald E. Knuth. "Literate Programming". In: *Computer Journal* 27.2 (May 1984), pp. 97–111. ISSN: 0010-4620. DOI: 10.1093/comjnl/27.2.97. URL: https://doi.org/10.1093/comjnl/27.2.97.

[5] Nate Soares and Benja Fallenstein. "Aligning Superintelligence with Human Interests: A Technical Research Agenda". In: (2014).

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 24, 2025

Valentin Meyer