

Baylor Environmental AI Research System (BEARS): An Agentic AI Project to Combat Climate Change

Abanisenioluwa Orojo^{1*}, Bikram Khanal^{1*}, Emmanuelli El-Mahmoud^{1*},
Joseph Yu^{1*}, Maisha Binte Rashid^{1*}, Paapa Quansah^{2*},
Sri Manjusha Tella^{1*}, Tianqi (Kirk) Ding^{2*}, Xiang Fang^{1*},
Pablo Rivas¹, and Maryam Samami¹

¹ Department of Computer Science, Baylor University, Waco, TX 76798, USA

² Dept. of Electrical & Computer Eng., Baylor University, Waco, TX 76798, USA
{abanisenioluwa_oroj1, bikram_khanal1, emmanuelli_elmahmou1, joseph_yu1,
maisha_rashid1, paapa_quansah1, srimanjusha_tella1, kirk_ding1,
xiang_fang1, pablo_rivas, maryam_samami1}@baylor.edu

Abstract. Agentic AI architectures, long theorized but limited by compute, have regained feasibility with large language models. Here we present the Baylor Environmental AI Research System (BEARS), an autonomous multi-agent pipeline in which nine collaborating agents exchange structured JSON via a shared key-value store to generate, assess, and rank deep learning research ideas addressing climate change. Agents perform tasks such as literature retrieval, idea generation, evidence synthesis, feasibility analysis, carbon auditing, impact estimation, risk evaluation, utility scoring, and control, iterating until defined thresholds are met. BEARS illustrates how modular agentic design can deliver transparent, reproducible, and ethically aware AI-driven research workflows.

Keywords: agentic AI · multi-agent pipeline · climate change research.

Regular Research Paper

1 Introduction

Climate change is a defining challenge of the twenty-first century, with rising temperatures and more frequent extreme events disrupting carbon cycles, biodiversity, and human societies [19,1]. Deep learning has advanced environmental modeling, improving weather forecasts with convolutional networks [2], correcting biases in circulation models [18], and enhancing prediction of phenomena such as ENSO and MJO dynamics [16,12]. Yet, the proliferation of methods makes it difficult to choose approaches that balance predictive power, resource demands, and ethical considerations [11,17].

* Alphabetical. These authors contributed equally to this work.

To address this gap, we introduce the Baylor Environmental AI Research System (BEARS), an agentic framework built in **ollama** that orchestrates nine collaborating agents. Each agent performs a focused task, ranging from automated keyword generation and literature retrieval to evidence synthesis, feasibility scoring, carbon auditing, impact estimation, risk assessment, utility calculation, and iterative control, exchanging structured JSON records via a shared key-value store. Over up to three refinement rounds, BEARS screens proposals against target thresholds (feasibility ≥ 0.6 , risk $\leq$ medium, utility ≥ 0.5), preserving a clear audit trail for each idea’s evolution.

We validate BEARS on a collection of climate-focused concepts, demonstrating how modular agents can systematically surface research directions that are technically viable, environmentally responsible, and ethically informed. Our contributions are a scalable agent-based pipeline that unifies idea generation with multi-criteria evaluation, the integration of carbon and ethical assessments into an automated workflow, and a proof-of-concept for transparent, reproducible AI-driven research prioritization. The BEARS code is publicly available at <https://github.com/Rivas-AI/BEARS.git>.

The remainder of this paper is structured as follows. Section 2 reviews prior work on deep learning in environmental science and agent-based AI pipelines. Section 3 details the BEARS architecture, describing the nine collaborating agents and their interactions. In Section 4, we present the results of our evaluation, including the ranking of candidate ideas against feasibility, utility, risk, and carbon metrics. Section 5 examines the key limitations of our approach, while Section 6 discusses ethical considerations arising from model bias, transparency, and environmental impact. Section 7 outlines intended use and responsible deployment guidance. Finally, Section 8 concludes and proposes directions for future work.

2 Related Work

2.1 Deep Learning for Climate and Environmental Science

Deep learning has delivered notable advances in environmental forecasting and analysis. Eraliev and Lee apply time-series neural models to predict microclimate variables in indoor hydroponic greenhouses, improving accuracy over varying intervals [8]. Wang and Tian employ deep residual networks for bias correction and downscaling of general circulation models, reducing errors in surface temperature projections [18]. Shin et al. explore neural approaches to model ENSO dynamics, though they point out persistent challenges in explaining model decisions [16]. Kim et al. use bias-corrected deep learning to enhance Madden-Julian Oscillation forecasts, demonstrating better seasonal prediction skill [12]. These studies advance predictive methods but stop short of end-to-end frameworks for idea generation and systematic evaluation in climate contexts.

2.2 Agent-Based AI Systems and Pipelines

Agent architectures have been used to coordinate multiple AI tasks in complex settings. Moon and Ahn describe feedback loops for continuous AI system improvement on web platforms, stressing change management requirements [14]. Chen et al. integrate human oversight into agent pipelines to enforce ethical checks at runtime [3]. Morley et al. propose an ethics-as-a-service model, noting difficulties when policy constraints shift dynamically [15]. Building on these ideas, our framework deploys nine specialized agents that collaborate via a shared key-value store, covering retrieval, generation, scoring, auditing, and iteration.

2.3 Automated Literature Retrieval

Automated retrieval systems streamline the gathering of relevant studies. Huang et al. review API-driven methods for extracting AI and climate research, highlighting biases in keyword-based queries that can omit niche work [10]. Other tools may reinforce citation loops or favor established topics. Our approach augments semantic search with novelty scoring to surface underrepresented methods and ensure broad coverage.

2.4 Evaluation of Feasibility, Carbon Impact, and Ethical Risk

Responsible AI deployment requires quantitative evaluation across multiple dimensions. Siau and Wang outline core AI ethics principles but lack specific metrics for environmental applications [17]. Guan et al. identify risk factors in AI decision processes and suggest mitigation tactics [9]. McGrath et al. assess gaps in enterprise risk management for AI, finding a need for tighter integration of ethical guidelines [13]. The EAIIFT framework offers structured ethical reasoning but remains untested in climate scenarios [7]. Our work unifies feasibility analysis, carbon auditing, impact estimation, and ethical checks into a cohesive scoring pipeline.

3 Design

In this section, we describe the design of each of the agents depicted in Fig. 1. Each agent is described in the following paragraphs, along with its corresponding pseudocode description and the underlying model choice.

3.1 Agent 1: Paper Retriever

Agent 1 acts as the reconnaissance unit of our pipeline. It forges keyword combinations at the intersection of climate change and deep learning, interrogates the Semantic Scholar API, and filters returned publications by novelty metrics to surface under-explored avenues. This agent employs structured search and metadata heuristics, requiring minimal inferential depth but demanding high throughput and precision.

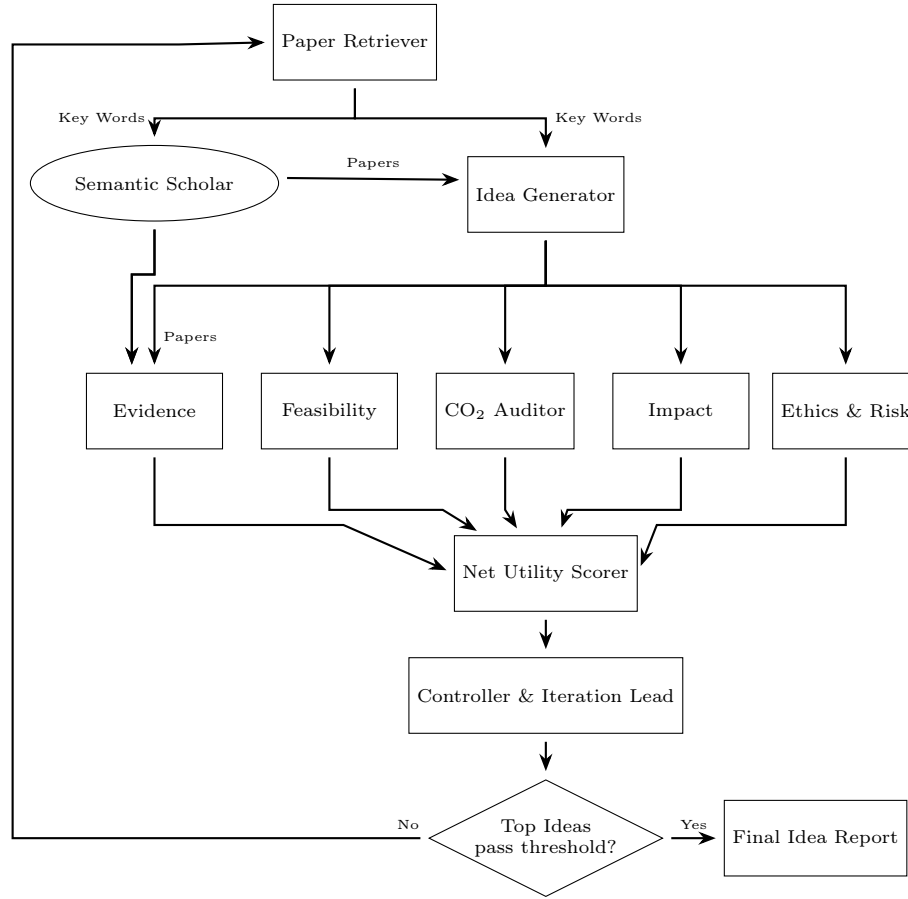


Fig. 1: Workflow diagram of the BEARS pipeline.

Model Specification We provision Ollama with the LLaMA 3.2 model, ≈ 7 B parameters and a 14 GB memory footprint, enabling rapid execution on consumer-grade GPUs or well-equipped CPUs.

Further architectural details, prompt templates, and JSON schemas are in Appendix A.

3.2 Agent 2: Idea Generator

Agent 2 is our ideation engine. It ingests 50 curated keywords and associated abstracts from Agent 1, then synthesizes 15–20 actionable deep learning concepts aimed at climate mitigation. Each concept is output as a structured object with a concise title, a 2–3 sentence technical description, and its data requirements (e.g., satellite imagery, time-series sensors).

Agent 1 Paper Retriever

Input: Topic (e.g., “Climate Change and Deep Learning”)

Output: List of 50 climate-related keywords

```

1: Begin Agent_1(topic)
2:   kb ← InitializeKnowledgeBaseModel(LLaMA 3.2)
3:   keywords ← kb.GenerateClimateKeywords(topic)
4:   papers ← QuerySemanticScholar(keywords)
5:   filtered_papers ← FilterPapers(papers)
6:   final_keywords ← ExtractKeywords(filtered_papers, 50)
7:   return final_keywords
8: End
    
```

Agent 2 Idea Generator

Input: keywords : List of 50 keyword strings

Input: papers : List of paper metadata and abstracts

Output: List of 15–20 structured project ideas

```

1: Begin Agent_2(keywords, papers)
2:   model ← InitializeReasoningModel(LLaMA 3.2)
3:   prompt ← BuildPrompt(keywords, papers)
4:   response ← model.GenerateIdeas(prompt)
5:   ideas ← ParseIdeas(response)
6:   for all idea in ideas do
7:     idea.title ← ExtractTitle(idea)
8:     idea.description ← ExtractDescription(idea, 2–3 sentences)
9:     idea.data_needs ← ExtractDataNeeds(idea)
10:    idea.idea_id ← GenerateID()
11:  end for
12:  return ideas
13: End
    
```

Model Specification Ollama with LLaMA 3.2 (≈ 7 B parameters, 14 GB memory) provides the creative reasoning capacity needed for high-quality idea formulation while remaining performant. Complete design diagrams, sample prompts, and I/O schemas appear in Appendix B.

3.3 Agent 3: Evidence Summarizer

Agent 3 functions as a rapid literature synthesizer. Given an idea and its candidate papers, it performs targeted retrieval (via the same API logic as Agent 1) and distills 5–7 bullet points that capture the core findings and relevance to the idea. Its role is to convert raw abstracts into concise evidence summaries, supporting downstream evaluation.

Model Specification Ollama with LLaMA 3.2 (≈ 7 B parameters, 14 GB memory) delivers efficient summarization capacity on standard hardware. See Appendix C for full flowcharts, prompt text, and JSON definitions.

Agent 3 Evidence Summarizer

Input: *idea* : a project idea object**Input:** *papers*: list of related paper abstracts**Output:** *evidence_summary*: list of 5–7 bullet points

```

1: Begin Agent_3(idea, papers)
2:   model  $\leftarrow$  InitializeReasoningModel(LLaMA 3.2)
3:   prompt  $\leftarrow$  BuildEvidencePrompt(idea, papers)
4:   response  $\leftarrow$  model.GenerateSummary(prompt)
5:   evidence_summary  $\leftarrow$  ExtractBulletPoints(response)
6:   return evidence_summary
7: End

```

3.4 Agent 4: Feasibility Analyst

Agent 4 serves as the pipeline’s technical auditor. It evaluates each idea along four dimensions, data availability, model suitability, compute demands, and deployment complexity, and assigns a normalized feasibility score in $[0.0, 1.0]$. The agent also provides concise justification notes to document key constraints and enablers.

Model Specification Ollama with LLaMA 3.2 (≈ 7 B parameters, 14 GB memory) executes rule-based reasoning with consistency and speed. Extended details are available in Appendix D.

Agent 4 Feasibility Analyst

Input: *idea_list*: list of ideas (title, description, data_needs, idea_id)**Output:** *results*: list of {*idea_id*, *feasibility_score*, *feasibility_notes*}

```

1: Begin Agent_4(idea_list)
2:   model  $\leftarrow$  InitializeReasoningModel(LLaMA 3.2)
3:   results  $\leftarrow$  []
4:   for all idea in idea_list do
5:     prompt  $\leftarrow$  BuildFeasibilityPrompt(idea)
6:     response  $\leftarrow$  model.AnalyzeFeasibility(prompt)
7:     score, notes  $\leftarrow$  ParseFeasibilityOutput(response)
8:     results.append({ "idea_id": idea.idea_id, "feasibility_score": score, "feasibility_notes": notes })
9:   end for
10:  return results
11: End

```

3.5 Agent 5: Carbon Auditor

Agent 5 quantifies the environmental cost of each idea by estimating annual CO₂ emissions from training and inference. It multiplies projected compute hours and

hardware power draw by standard emission factors, producing a kg CO₂/year metric for each project.

Model Specification We deploy Qwen3:32b for its fast, accurate numerical reasoning suited to structured estimation tasks. The full diagram, prompt chain, and JSON schemas reside in Appendix E.

Agent 5 Carbon Auditor

Input: `idea_list`: list of ideas (with `idea_id`)

Output: `results`: list of `{idea_id, co2_kg_per_year}`

```

1: Begin Agent_5(idea_list)
2:   model  $\leftarrow$  InitializeBasicModel(Qwen3:32b)
3:   results  $\leftarrow$  []
4:   for all idea in idea_list do
5:     prompt  $\leftarrow$  FormatCarbonPrompt(idea.idea_id)
6:     response  $\leftarrow$  model.EstimateCO2(prompt)
7:     co2_kg  $\leftarrow$  ParseJSONResponse(response)
8:     if co2_kg > 0 then
9:       results.append({ "idea_id": idea.idea_id, "co2_kg_per_year": co2_kg })
10:    end if
11:  end for
12:  return results
13: End

```

3.6 Agent 6: Impact Estimator

Agent 6 appraises the positive effects of each idea across three axes: emissions reduction potential, policy applicability, and societal benefit. It integrates these metrics into a unified impact score in $[0.0, 1.0]$, accompanied by a brief summary of its reasoning process.

Model Specification Qwen3:32b provides the mixed qualitative–quantitative reasoning ability necessary for nuanced impact estimation. Refer to Appendix F for supplementary materials.

3.7 Agent 7: Ethics & Risk Checker

Agent 7 performs a comprehensive ethical audit. It examines bias, privacy vulnerabilities, misuse potential, and explainability, then assigns a categorical risk level (low/medium/high) and proposes three targeted mitigation strategies.

Model Specification Qwen3:32b’s advanced reasoning capabilities enable deep contextual analysis of abstract ethical considerations at scale. Detailed flow diagrams and JSON specifications can be found in Appendix G.

Agent 6 Impact Estimator

Input: `idea_list`: list of ideas (with `idea_id`, title, description)**Output:** `results`: list of `{idea_id, impact_score, summary}`

```

1: Begin Agent_6(idea_list)
2:   model  $\leftarrow$  InitializeReasoningModel(Qwen3:32b)
3:   results  $\leftarrow$  []
4:   for all idea in idea_list do
5:     prompt  $\leftarrow$  BuildImpactPrompt(idea)
6:     response  $\leftarrow$  model.EstimateImpact(prompt)
7:     impact_score, summary  $\leftarrow$  ParseImpactOutput(response)
8:     results.append({ "idea_id": idea.idea_id, "impact_score": impact_score,
       "summary": summary })
9:   end for
10:  return results
11: End

```

Agent 7 Ethics & Risk Checker

Input: `idea_list`: list of ideas (with `idea_id`, title, description)**Output:** `results`: list of `{idea_id, risk_level, mitigation_suggestions}`

```

1: Begin Agent_7(idea_list)
2:   model  $\leftarrow$  InitializeReasoningModel(Qwen3:32b)
3:   results  $\leftarrow$  []
4:   for all idea in idea_list do
5:     prompt  $\leftarrow$  BuildRiskPrompt(idea)
6:     response  $\leftarrow$  model.EvaluateEthicalRisks(prompt)
7:     risk_level, mitigations  $\leftarrow$  ParseRiskOutput(response)
8:     results.append({ "idea_id": idea.idea_id, "risk_level": risk_level, "mitiga-
       tion_suggestions": mitigations })
9:   end for
10:  return results
11: End

```

3.8 Agent 8: Net Utility Scorer

Agent 8 synthesizes all prior metrics into a single utility score:

$$U = w_f F + w_i I - w_c C - w_r R,$$

with $F, I \in [0, 1]$, C normalized CO₂, $R \in \{1, 3, 5\}$, and user-defined weights w_f, w_i, w_c, w_r . This aims to produce a fair, transparent ranking of project ideas.

Model Specification We leverage Ollama with LLaMA 3.2 (≈ 7 B parameters, 14 GB memory) to perform the scoring computation and format outputs. See Appendix H for complete specifications.

Agent 8 Net Utility Scorer

Input: `idea_list`: list of ideas with fields $\{\text{idea_id}, F, I, C, R\}$

Output: each idea augmented with `utility_score`

```

1: Begin Agent_8(idea_list)
2:   model  $\leftarrow$  InitializeScoringModel(LLaMA 3.2)
3:   results  $\leftarrow []$ 
4:   for all idea in idea_list do
5:     R  $\leftarrow$  ConvertRiskLevel(idea.R)
6:     C  $\leftarrow$  NormalizeCO2(idea.C)
7:     U  $\leftarrow w_f \times F + w_i \times I - w_c \times C - w_r \times R$ 
8:     idea.utility_score  $\leftarrow U$ 
9:     results.append(idea)
10:  end for
11:  return results
12: End

```

3.9 Agent 9: Controller & Iteration Lead

Agent 9 integrates results from Agents 4–8 to select, report on, or re-iterate ideas. It applies the weighted utility ranking (Section 3.8), then filters for:

$$F \geq 0.6, \quad R \in \{\text{low, medium}\}, \quad U \geq 0.5.$$

Qualified ideas are rendered into project-specific Markdown reports; if none qualify, the agent calls Agent 1 to restart the pipeline. This control logic and automated report generation use Mistral’s moderate reasoning capacity.

Model Qwen3:32b is used for structured decision logic and automated Markdown report generation. Further details are archived in Appendix I.

4 Results and Discussion

We applied BEARS to an initial set of generated project ideas, evaluating each according to four dimensions: utility, feasibility, ethical risk, and annual CO₂ emissions. The pipeline leveraged open-weight models, LLaMA 3.2 for retrieval, summarization, and scoring; Qwen3:32b for numerical estimation and ethical analysis; and Mistral for final report generation, thereby balancing computational efficiency with analytical rigor.

From the full pool, we selected the top three candidates meeting our pre-defined criteria (utility ≥ 0.5 , feasibility ≥ 0.6 , and risk $\leq$ medium). Table 1 summarizes their key metrics.

4.1 Arctic Anomaly Detection with Computer Vision (ID: 8)

This project employs convolutional neural networks (CNNs) to detect Arctic climate anomalies from satellite imagery, providing real-time actionable insights

Agent 9 Controller & Iteration Lead

Input: `idea_data_list`: list of ideas with fields $\{\text{idea_id}, F, I, C, R, U\}$ (U = utility score)**Output:** Generate Markdown reports for qualified ideas or restart pipeline

```

1: Begin Agent_9(idea_data_list)
2:   model  $\leftarrow$  InitializeReportModel(Qwen3:32b)
3:   qualified  $\leftarrow$  []
4:   for all idea in idea_data_list do
5:     if idea.feasibility_score  $\geq$  0.6 and idea.utility_score  $\geq$  0.5 and
       idea.risk_level  $\in$  {"low", "medium"} then
6:       qualified.append(idea)
7:     end if
8:   end for
9:   if qualified  $\neq$  [] then
10:    for all idea in qualified do
11:      prompt  $\leftarrow$  BuildMarkdownPrompt(idea)
12:      report  $\leftarrow$  model.GenerateMarkdown(prompt)
13:      SaveReport(report, idea.idea_id)
14:    end for
15:  else
16:    RestartPipelineFromAgent1()
17:  end if
18: End

```

Table 1: Summary of Key Metrics for Top Project Ideas

Project ID	Utility Score	Feasibility	Risk Level	CO ₂ (kg/yr)
8	12.02	0.65	Medium	1,047,500
9	15.90	0.70	High	10,000
10	-10.50	0.70	Medium	3,500,000

crucial for environmental monitoring. Its strength lies in its balanced profile of high utility and moderate feasibility, offering substantial potential to enhance monitoring capabilities in a critical climate-sensitive region. The primary challenge is its considerable environmental footprint, necessitating computational optimizations through model pruning and energy-efficient hardware.

The whole generated idea report can be accessed [here](#).

4.2 Climate Resilience Prediction using GANs (ID: 9)

This approach uses Generative Adversarial Networks (GANs) to generate synthetic, realistic future climate scenarios supporting proactive adaptation in urban and agricultural planning. It offers high utility and minimal environmental footprint, positioning it as a promising solution. However, its high ethical risk requires careful management through transparent validation processes, stakeholder collaboration, and strict misuse controls.

The complete idea report can be found [here](#).

4.3 Sea Level Rise Prediction with Deep Learning (ID: 10)

This project utilizes transformer networks integrating diverse data sources to provide high-resolution forecasts for sea-level rise, crucial for regional planning. While technically feasible, significant challenges include its negative utility score and exceptionally high CO₂ emissions. These issues emphasize the urgent need for substantial computational efficiency improvements or the adoption of less resource-intensive alternatives.

The complete generated idea report can be accessed [here](#).

Collectively, these case studies illustrate the trade-offs inherent in AI-driven climate solutions: balancing impact and feasibility against environmental and ethical constraints. BEARS demonstrates its capacity to surface high-value projects and to flag those requiring further refinement, validating its role as a rigorous prioritization framework for climate-focused deep learning research.

5 Limitations

The development and execution of BEARS were shaped by several interrelated constraints that merit consideration. A compressed development timeline limited opportunities for extensive experimentation, hyperparameter fine-tuning, and robustness validation. In particular, multi-round iterations orchestrated by the Controller agent could not be explored exhaustively, potentially constraining the diversity of top-ranked solutions.

Resource availability also influenced our architectural choices. While higher-capacity models such as Qwen3:32b and Mistral were employed selectively, for ethical analysis and report generation, respectively, lighter-weight models like LLaMA 3.2 were necessary for other agents to maintain throughput. This heterogeneous model deployment may have introduced variability in reasoning depth and consistency across pipeline stages.

Access to certain tools and pre-trained components was further hampered by licensing restrictions and API limitations. In several instances, the absence of preferred libraries required the adoption of less specialized workarounds, which may have affected the precision of metadata retrieval, numerical estimation, or ethical risk assessment.

Finally, the utility ranking formula depends on manually assigned weight coefficients, for feasibility, impact, risk, and carbon emissions, drawn from prior literature on decision weighting [4]. Without a formal sensitivity analysis, the resulting idea rankings remain contingent on these subjective choices. Future work should incorporate systematic weight calibration and uncertainty quantification to enhance the robustness of the prioritization mechanism.

6 Ethical Considerations

Ensuring ethical integrity across the BEARS pipeline required deliberate attention to potential biases, misuse risks, and environmental impacts. Although we

employed open-weight models and locally hosted inference to maximize transparency, the underlying training data of our language models may still encode social and geographic biases. Such biases could skew feasibility and impact assessments against projects targeting low-income or data-scarce regions, thereby perpetuating existing inequities in climate intervention strategies.

Moreover, the generation of predictive recommendations for infrastructure planning or policy development carries inherent dangers if these outputs are interpreted as definitive guidance without adequate human review. To mitigate this, we incorporated an Ethics & Risk Checker (Agent 7) that assigns categorical risk levels and proposes targeted mitigation measures. However, fully safeguarding against unintended consequences will demand ongoing collaboration between domain experts and model developers.

The opacity of single-number outputs from certain scoring agents also posed a transparency challenge. While our implementation requires accompanying justification notes, future iterations should integrate established explainability frameworks that trace numerical scores back to specific evidence sources and decision pathways. This would foster greater accountability and facilitate auditability by external stakeholders.

From an environmental standpoint, the carbon audits performed by Agent 5 provided coarse estimates of emissions associated with model training and inference. These estimations relied on static power assumptions; a more precise approach would leverage real-time monitoring tools, such as CodeCarbon [6], to capture dynamic energy profiles.

Finally, the choice to use open-weight models via Ollama underscores our commitment to reproducibility and open science. Nevertheless, the sensitivity of outputs to prompt formulation and model updates remains a consideration. Continuous version control of both model weights and prompt templates is essential to preserve the reproducibility and stability of results over time. Ethical evaluation, therefore, must remain an iterative and human-supervised process throughout the lifecycle of the BEARS framework.

7 Intended Use

The BEARS pipeline emerged from an advanced deep learning course as a proof-of-concept for autonomous idea generation and evaluation in the climate domain. While developed under time and resource constraints by graduate students, the system illustrates how modular AI agents can accelerate the early stages of research planning. We envisage BEARS as a complementary tool for researchers and practitioners seeking inspiration and structured evaluation of novel deep learning approaches to climate challenges.

By sharing the design, prompts, and weight configurations openly, we aim to foster transparent exploration rather than provide a turnkey solution. Users should apply human judgment at each stage, particularly when interpreting model scores or deploying any recommended ideas. Given the ethical and technical limitations discussed, bias in training data, sensitivity of scoring weights,

and approximate emissions estimates, BEARS is intended to guide brainstorming and preliminary feasibility assessment, not to replace domain expertise or rigorous empirical validation. Continuous oversight, version control, and sensitivity analysis are essential when extending or adapting this framework for real-world research and policy applications.

8 Conclusions and Future Work

We have introduced the Baylor Environmental AI Research System (BEARS), an end-to-end agentic framework that synthesizes literature retrieval, idea generation, evidence summarization, feasibility analysis, carbon auditing, impact estimation, ethical risk checking, and utility scoring into a single coherent pipeline. Our case studies demonstrate BEARS’ ability to surface promising deep learning projects, such as Arctic anomaly detection and resilience forecasting via GANs, while also flagging proposals that demand further optimization, particularly around computational efficiency and emissions reduction. By combining open-weight models with transparent scoring mechanisms, BEARS exemplifies how modular AI agents can accelerate the research ideation process without compromising reproducibility or ethical accountability.

At the same time, our work underscores the importance of human oversight, sensitivity analysis, and continuous validation. The pipeline’s reliance on manually calibrated weight coefficients and approximate carbon estimates highlights opportunities for systematic refinement. Likewise, the potential for model bias and opacity in single-number summaries motivates deeper integration of explainability tools and human-in-the-loop review at every stage.

Looking ahead, we plan to enhance the Evidence Summarizer by incorporating embedding-based semantic ranking to select and synthesize the most pertinent literature directly into project briefs. This will be complemented by an automated novelty assessment module, which uses cosine similarity on document embeddings (e.g., SPECTER [5]) to categorize ideas by originality. Further extensions include real-time carbon monitoring, formal sensitivity analyses of weight parameters, and richer explainability interfaces that trace agent outputs back to source evidence. By iteratively refining each component and embedding ethical safeguards, we aim to evolve BEARS into a robust platform for generating, validating, and prioritizing AI-driven solutions to the urgent challenges of climate change.

Acknowledgments. Part of this work was funded by the National Science Foundation under grant CNS-2136961.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Barnes, E.A., Hurrell, J.W., Ebert-Uphoff, I., Anderson, C., Anderson, D.: Viewing forced climate patterns through an ai lens. *Geophysical Research Letters* **46**(22), 13389–13398 (2019)
2. Chattopadhyay, A., Hassanzadeh, P., Pasha, S.: Predicting clustered weather patterns: A test case for applications of convolutional neural networks to spatio-temporal climate data. *Scientific Reports* **10**(1317) (2020). <https://doi.org/10.1038/s41598-020-57897-9>, <https://doi.org/10.1038/s41598-020-57897-9>
3. Chen, X., Wang, X., Qu, Y.: Constructing ethical ai based on the “human-in-the-loop” system. *Systems* **11**, 548 (2023). <https://doi.org/10.3390/systems11110548>
4. Chouldechova, A., Roth, A.: The frontiers of fairness in machine learning (10 2018). <https://doi.org/10.48550/arXiv.1810.08810>
5. Cohan, A., Feldman, S., Beltagy, I., Downey, D., Weld, D.S.: Specter: document-level representation learning using citation-informed transformers. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020). <https://doi.org/10.18653/v1/2020.acl-main.207>
6. Courty, B., Schmidt, V., Goyal-Kamal, MarionCoutarel, Feld, B., Lecourt, J., LiamConnell, SabAmine, inimaz, supatomic, Léval, M., Blanche, L., Cruveiller, A., ouminasara, Zhao, F., Joshi, A., Bogroff, A., Saboni, A., de Lavoreille, H., Laskaris, N., Abati, E., Blank, D., Wang, Z., Catovic, A., alencon, Stechty, M., Bauer, C., Lucas-Otavio, JPW, MinervaBooks: mlco2/codecarbon: v2.4.1 (May 2024). <https://doi.org/10.5281/zenodo.11171501>, <https://doi.org/10.5281/zenodo.11171501>
7. Ejjami, R.: Ethical artificial intelligence framework theory (eaift): a new paradigm for embedding ethical reasoning in ai systems. *International Journal for Multi-disciplinary Research* **6** (2024). <https://doi.org/10.36948/ijfmr.2024.v06i05.28231>
8. Eraliev, O., Lee, C.: Performance analysis of time series deep learning models for climate prediction in indoor hydroponic greenhouses at different time intervals. *Plants* **12**, 2316 (2023). <https://doi.org/10.3390/plants12122316>
9. Guan, H., Dong, L., Zhao, A.: Ethical risk factors and mechanisms in artificial intelligence decision making. *Behavioral Sciences* **12**, 343 (2022). <https://doi.org/10.3390/bs12090343>
10. Huang, C., Zhang, Z., Mao, B., Yao, X.: An overview of artificial intelligence ethics. *Ieee Transactions on Artificial Intelligence* **4**, 799–819 (2023). <https://doi.org/10.1109/tai.2022.3194503>
11. Iglesias-Suarez, F., Gentine, P., Fernández, B., Beucler, T., Pritchard, M., Runge, J., Eyring, V.: Causally-informed deep learning to improve climate models and projections. *Journal of Geophysical Research Atmospheres* **129** (2024). <https://doi.org/10.1029/2023jd039202>
12. Kim, H., Ham, Y., Joo, Y., Son, S.: Deep learning for bias correction of mjo prediction. *Nature Communications* **12** (2021). <https://doi.org/10.1038/s41467-021-23406-3>
13. McGrath, Q., Hevner, A., Vreede, G.: Managing ethical risks of artificial intelligence in business applications (2024). <https://doi.org/10.36227/techrxiv.170905835.50964792/v1>
14. Moon, D., Ahn, S.: A study on functional requirements and inspection items for ai system change management and model improvement on the web platform.

- Journal of Web Engineering pp. 831–848 (2024). <https://doi.org/10.13052/jwe1540-9589.2366>
15. Morley, J., Elhalal, A., Garcia, F., Kinsey, L., Mökander, J., Floridi, L.: Ethics as a service: a pragmatic operationalisation of ai ethics. *Minds and Machines* **31**, 239–256 (2021). <https://doi.org/10.1007/s11023-021-09563-w>
 16. Shin, N., Ham, Y., Kim, J., Cho, M., Kug, J.: Application of deep learning to understanding enso dynamics. *Artificial Intelligence for the Earth Systems* **1** (2022). <https://doi.org/10.1175/aies-d-21-0011.1>
 17. Siau, K., Wang, W.: Artificial intelligence (ai) ethics. *Journal of Database Management* **31**, 74–87 (2020). <https://doi.org/10.4018/jdm.2020040105>
 18. Wang, F., Tian, D.: On deep learning-based bias correction and downscaling of multiple climate models simulations. *Climate Dynamics* **59**, 3451–3468 (2022). <https://doi.org/10.1007/s00382-022-06277-2>
 19. Wu, C., Wang, J., Ciais, P., Peñuelas, J., Zhang, X., Sonnentag, O., Tian, F., Wang, X., Wang, H., Liu, R., Fu, Y., Ge, Q.: Widespread decline in winds delayed autumn foliar senescence over high latitudes. *Proceedings of the National Academy of Sciences* **118** (2021). <https://doi.org/10.1073/pnas.2015821118>

A Supplementary Materials for Agent 1

A.1 Agent Diagram

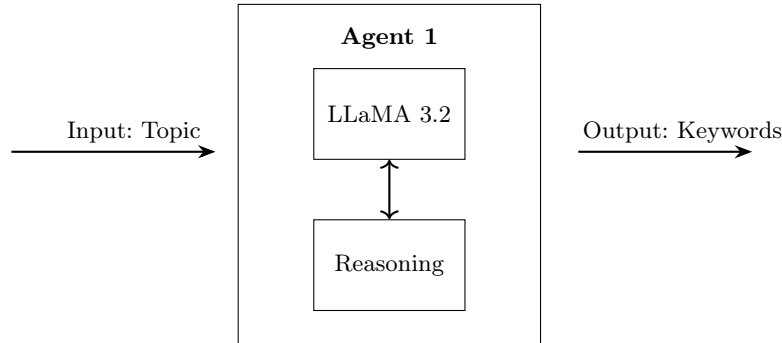


Fig. 2: Block diagram for Agent 1: Paper Retriever.

A.2 Sample Prompt and Chain of Thought

PROMPT_FOR_KEYWORDS = "As an expert keyword generator for academic research, your task is to produce a comprehensive list of keywords. These keywords are for querying the Semantic Scholar API to find papers at the intersection of <Topic>Climate Change and Deep Learning</Topic>. The goal is to uncover original and novel research directions."

Instructions for keyword generation:

- Focus on combining concepts from 'Climate Change' and 'Deep Learning'.
- Aim for keywords that are specific and could lead to novel research areas or innovative applications.
- The output **MUST** be a single string containing the keywords.
- Each keyword or keyword phrase **MUST** be separated by a comma and a space. For example: 'keyword1, keyword phrase 2, keyword3'.
- Do **NOT** use bullet points, numbered lists, introductory phrases (like "Here is a list..."), or any other formatting. Only provide the comma-separated list of keywords.

The generated list must contain exactly 50 keywords.

A.3 JSON Input/Output Schemas**JSON input**

```
response = ollama.chat(
    model=model_name,
    messages=[
        {
            'role': 'user',
            'content': prompt_text,
        },
    ]
)
print("Received response from Ollama.")
```

JSON output

```
{
  "type": "object",
  "properties": {
    "keywords": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "term": { "type": "string" }
        },
        "required": ["term"]
      }
    }
  },
  "required": ["keywords"]
}
```

A.4 Prompt Templates and API Call Structures

```

AGENT_1 = "llama3.2:3b"
BEARER_TOKEN = [insert token here]
SS_HEADERS = {"Authorization": f"Bearer {BEARER_TOKEN}"}
KEYWORD_PERMUTATION = 20

def query_ollama_model(model_name, prompt_text):
    try:
        print(f"Sending prompt to Ollama model: {model_name}...")
        response = ollama.chat(
            model=model_name,
            messages=[
                {
                    'role': 'user',
                    'content': prompt_text,
                },
            ]
        )
        print("Received response from Ollama.")
        return response['message']['content']
    except Exception as e:
        return f"An error occurred while communicating with Ollama: {e}"

PROMPT_FOR_KEYWORDS = f"""As an expert keyword generator for academic research, your task is to
produce a comprehensive list of keywords. These keywords are for querying the Semantic Scholar
API to find papers at the intersection of <Topic>Climate Change and Deep Learning</Topic>. The
goal is to uncover original and novel research directions.

Instructions for keyword generation:
1. Focus on combining concepts from 'Climate Change' and 'Deep Learning'.
2. Aim for keywords that are specific and could lead to novel research areas or innovative
applications.
3. The output MUST be a single string containing the keywords.
4. Each keyword or keyword phrase MUST be separated by a comma and a space. For example:
'keyword1, keyword2, keyword3'.
5. Do NOT use bullet points, numbered lists, introductory phrases (like "Here is a
list..."), or any other formatting. Only provide the comma-separated list of keywords.

Generate a diverse set of 50 of these keywords. The generated keywords should be strictly 50
and only 50. Can't be more or less. Example of desired output format: 'Climate modeling,
climate data, Arctic anomaly detection, Deep Learning, Computer Vision, Climate Prediction'.
"""

reprompt = f"""Based on the following existing keywords about Climate Change and Deep Learning:

{', '.join(unique_keywords)}

Please generate {remaining} ADDITIONAL unique concise keywords/terms that are NOT in the
above list. These should be at the intersection of Climate Change and Deep Learning.

Output ONLY the new keywords as a comma-separated list with no additional text,
numbering, or formatting.
"""
    
```

B Supplementary Materials for Agent 2

B.1 Agent Diagram

B.2 Sample Prompt and Chain of Thought

```

prompt = f"""
You are Agent 2 in a climate-change deep learning pipeline.
Given the following keywords:
    
```

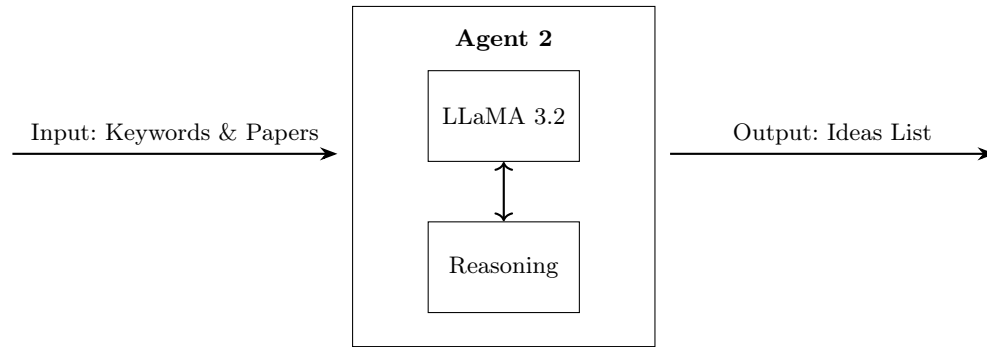


Fig. 3: Block diagram for Agent 2: Idea Generator.

```
{keywords}
```

And the following papers with abstracts:

```
{papers}
```

Propose 15-20 candidate project ideas. For each idea, return JSON with:

```
- title: a concise name
- description: 2-3 sentences
- data_needs: list of data types required
"""
```

The model's chain of thought guides it to combine domain terms with methodological steps before formatting the output as structured JSON.

B.3 JSON Input/Output Schemas

JSON input

```
{
  "keywords": [
    { "term": "keyword1" },
    { "term": "keyword2" },
    ...
  ]
}
```

JSON output

```
[
  {
    "title": "Short project title",
    "description": "Two to three sentence summary of the idea.",

```

```

        "data_needs": ["satellite imagery", "sensor time series"],
        "idea_id": "unique-id-123"
    },
    ...
]

```

B.4 Prompt Templates

```

prompt = f"""
You are Agent 2 (Idea Generator) in a climate-change deep learning pipeline.

Given the following keywords:
{keywords}

Given the following papers:
{paper}

Propose 15-20 candidate deep learning project ideas to combat climate change.

RESPONSE FORMAT:
Return your response as a simple formatted list with TITLE and DESCRIPTION for each
idea, separated by blank lines. For example:

TITLE: Climate Tipping Point Prediction System
DESCRIPTION: A deep learning system that identifies potential climate tipping points by
analyzing historical climate data and current trends. Uses recurrent neural networks to
process time series data from multiple sources to predict non-linear climate transitions.
DATA NEEDS: Historical climate records, current sensor data, satellite imagery

TITLE: Carbon Sequestration Optimization Network
DESCRIPTION: An ML algorithm that determines optimal locations and methods for carbon
sequestration based on geographical and atmospheric conditions. Combines computer vision
analysis of terrain with climate models to maximize carbon capture efficiency.
DATA NEEDS: Historical climate records, current sensor data, satellite imagery

Do not include any additional text, explanations, introductions, or conclusions. Start
directly with the first TITLE and end with the last DESCRIPTION. Ensure each project idea is
separated by exactly one blank line from the next idea.

Now, generate 15-20 creative and technically feasible project ideas based on the provided
keywords and papers.
"""

```

C Supplementary Materials for Agent 3

C.1 Agent Diagram

C.2 Sample Prompt and Chain of Thought

```

key_ideas_generation_prompt_1 = f"""
You are an AI Research Synthesizer. Given a project idea and a set of related
paper abstracts, distill the core findings into 5-7 concise bullet points
that capture the main themes and evidence across all sources.

Instructions:
1. Analyze all abstracts jointly to identify common methods, results, and gaps.
2. Formulate new summary points reflecting the combined insights.
3. Produce a unified list of bullet points without any framing text.

```

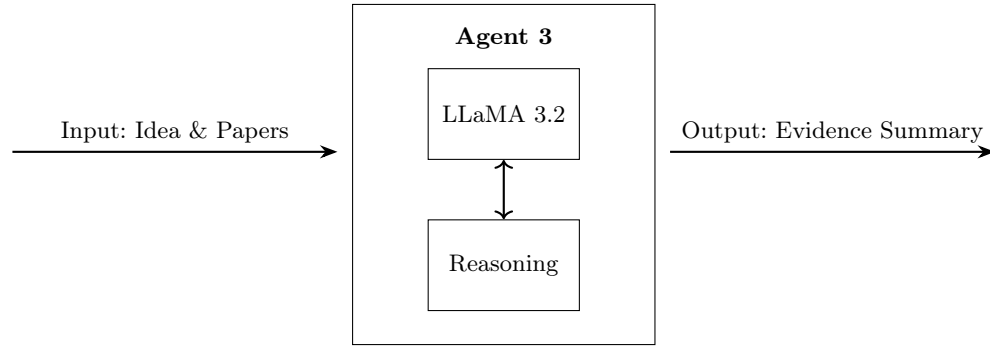


Fig. 4: Block diagram for Agent 3: Evidence Summarizer.

4. Use standard bullet characters (e.g., '-', '*').
5. Return exactly 5-7 points.

Input:

```

idea: [{"title", "description", "data_needs", "idea_id"}]
papers: [{"title", "abstract"}], ...
"""

```

C.3 JSON Input/Output Schemas

JSON input

```

{
  "idea": {
    "title": "...",
    "description": "...",
    "data_needs": ["..."],
    "idea_id": "..."
  },
  "papers": [
    {"title": "...", "abstract": "..."},
    ...
  ]
}

```

JSON output

```

[
  "Bullet point one summarizing a key finding",
  "Bullet point two summarizing another finding",
  ...
]

```

C.4 Prompt Templates

```

f"""
You are an AI Research Synthesizer. Your primary objective is to analyze a collection of ‘n’
research paper abstracts, all related to a common theme or a specific research idea within
<Topic>Climate Change and Deep Learning</Topic>. Your task is to distill these abstracts into a
single, unified set of key bullet points that capture the holistic picture and core gist of the
combined information. These points should highlight the most significant, recurring, or
foundational insights that emerge when considering all abstracts together.

**Instructions for Synthesizing Collective Key Points:**

1. Comprehensive Review: Thoroughly read and analyze ALL ‘n’ provided abstracts to
understand the full scope of information.
2. Identify Overarching Themes & Connections: Look for:
    * Common research questions, objectives, or problems addressed across multiple abstracts.
    * Recurring methodologies, techniques, or datasets employed.
    * Converging findings or consistent conclusions that appear in several abstracts.
    * Complementary information where different abstracts contribute unique pieces to a larger
    puzzle.
    * The overall narrative or argument these abstracts collectively support regarding the
    central theme or idea.
3. Synthesize, Don’t Just Aggregate: Your goal is not to pick one point from each abstract.
Instead, formulate new summary points that represent the synergistic understanding gained from
all abstracts. A single bullet point might draw from concepts mentioned in multiple abstracts.
4. Focus on the Core Gist: The bullet points should represent the most critical and
impactful insights that define the collective evidence or understanding presented. What are the
absolute must-know takeaways if someone were to understand the essence of these ‘n’ abstracts as
a whole?
5. Conciseness and Clarity: Each bullet point should be a clear, concise phrase or a short,
impactful sentence.
6. Number of Points: Aim for a focused list of 5-7 key bullet points in total for the
entire set of abstracts. The exact number can vary based on the richness and diversity of the
input, but the goal is a high-level synthesis.
7. Holistic Perspective: The final list of bullet points should read as a coherent summary
of the combined knowledge, not a disjointed collection.
8. Output Formatting (CRUCIAL):
    * Provide a single, unified list of bullet points.
    * Use standard bullet characters (e.g., ‘*’, ‘-’, or ‘•’).
    * Do NOT provide separate summaries for each abstract.
    * Do NOT include any introductory phrases (e.g., "Here are the synthesized key points...")
    or concluding remarks, other than the single bulleted list.

**Input Abstracts:**

You will now be provided with ‘n’ abstracts. Please process them collectively according to the
instructions above to generate a single list of synthesized key points.
--
"""

key_ideas_generation_prompt_2 = """
Your Task:
Generate a single, unified list of 5-7 key bullet points that synthesize the core gist and
holistic picture from ALL ‘n’ abstracts provided above. Adhere strictly to all instructions,
especially regarding the synthesis approach and output formatting.
"""

#API call
def search_papers(query: str, limit: int = 3) -> List[Dict]:
    """Search for papers using the given query."""
    SEARCH_URL = "https://api.semanticscholar.org/graph/v1/paper/search"

    search_params = {
        "query": query,
        "limit": limit,
        "fields": "paperId,title,externalIds,abstract"
    }

    try:
    
```

```

resp = requests.get(SEARCH_URL, params=search_params, headers=SS_HEADERS)
resp.raise_for_status()
return resp.json().get("data", [])
except requests.exceptions.RequestException as e:
    print(f"Search error for query '{query}': {e}")
    return []

```

D Supplementary Materials for Agent 4

D.1 Agent Diagram

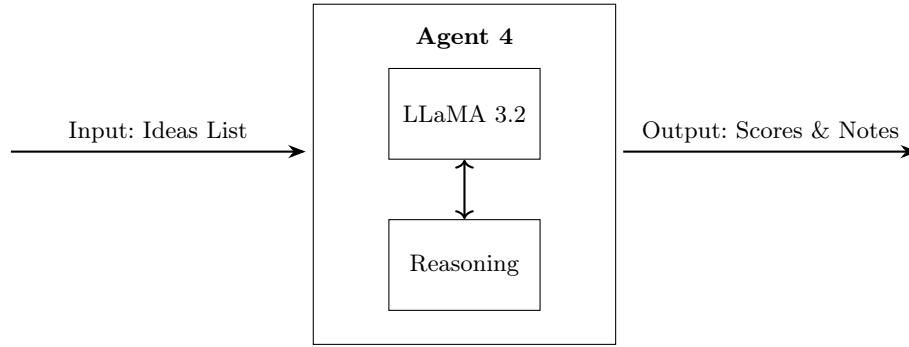


Fig. 5: Block diagram for Agent 4: Feasibility Analyst.

D.2 Sample Prompt and Chain of Thought

You are an AI Feasibility Analyst. Your task is to evaluate the practical viability of a proposed research idea in Climate Change and Deep Learning. You will receive the idea's title, description, and data requirements.

1. Assess:
 - Data Volume & Availability (e.g., public vs. proprietary, labeling)
 - Model Fit & Complexity (standard vs. novel architectures)
 - Compute Needs (GPU, HPC, specialized hardware)
 - Deployment Path & Scalability (integration, maintenance)
 2. Assign Feasibility Score: x in $[0.0, 1.0]$
 3. Provide 2-4 concise justification notes.
 4. Format exactly:


```
Feasibility Score: x, Notes: [note1; note2; ...]
```
- Input:
- ```
{ "title": ..., "description": ..., "data_needs": [...], "idea_id": "..." }
```

### D.3 JSON Input/Output Schemas

*JSON input*

```
[
 {
 "title": "...",
 "description": "...",
 "data_needs": ["..."],
 "idea_id": "..."
 },
 ...
]
```

*JSON output*

```
[
 {
 "idea_id": "...",
 "feasibility_score": 0.0-1.0,
 "feasibility_notes": "Concise notes explaining score"
 },
 ...
]
```

### D.4 Prompt Templates

```
prompt1 = f"""
You are an AI Feasibility Analyst. Your task is to meticulously evaluate the practical viability
of a proposed research idea related to <Topic>Climate Change and Deep Learning</Topic>. You will
be provided with the idea's title, description, and data requirements.
```

Based on this information, you must:

1. **\*\*Evaluate Feasibility Dimensions:\*\*** Thoroughly assess the idea against the following dimensions, making internal notes on each:
  - \* **\*\*Data Volume & Availability:\*\*** Consider the scale, accessibility, and preparation challenges of the required data. Note any significant hurdles (e.g., "requires petabytes of proprietary data," "extensive manual labeling needed") or facilitators (e.g., "uses readily available public datasets").
  - \* **\*\*Model Fit & Complexity:\*\*** Evaluate the suitability of deep learning. Note if standard models apply or if it requires novel, highly complex, or exceptionally resource-intensive models. Consider the technical risk associated with the proposed modeling approach.
  - \* **\*\*Compute Needs:\*\*** Estimate the computational resources. Note if it implies standard GPU usage, high-performance computing, or specialized hardware, and whether this is a significant barrier.
  - \* **\*\*Deployment Path & Scalability:\*\*** Assess the complexity of real-world implementation if successful. Note potential challenges in scalability, maintenance, or integration.
2. **\*\*Synthesize into a Feasibility Score:\*\*** Based on your detailed assessment of the dimensions above, assign a single numerical Feasibility Score.
  - \* The score **MUST** be a value between 0.0 and 1.0, inclusive.
  - \* 0.0 indicates extreme unfeasibility; 1.0 indicates high feasibility.
3. **\*\*Formulate Justification Notes:\*\*** Based on your evaluation of the dimensions, compose concise notes that clearly justify the assigned Feasibility Score. These notes should highlight the primary factors (both positive and negative) that influenced your scoring across the different feasibility dimensions. Aim for 2-4 key points in your notes.

```

4. **Provide Output in Specified Format:** Your response must contain both the Feasibility
Score and the Notes.

Input Research Idea Details:

"""
prompt2 = f"""
Output Requirement:

Your response MUST be formatted strictly as follows, with no other text, explanations, or
characters outside this structure on a single line:
‘Feasibility Score: x, Notes: [Your concise notes explaining the score, highlighting key factors
from data, model, compute, and deployment considerations]‘

Example of desired output format:
‘Feasibility Score: 0.65, Notes: Moderate data acquisition challenges due to specificity, but
utilizes established model architectures. Compute needs are manageable with standard GPUs.
Deployment path requires integration with existing sensor networks which could be complex.’

OR

‘Feasibility Score: 0.3, Notes: Relies on extremely large, currently unavailable datasets.
Proposed model is highly experimental with significant technical risk. Deployment would require
substantial new infrastructure.’

Now, analyze the provided idea details and output the Feasibility Score and corresponding Notes
in the specified format.

"""

```

## E Supplementary Materials for Agent 5

### E.1 Agent Diagram

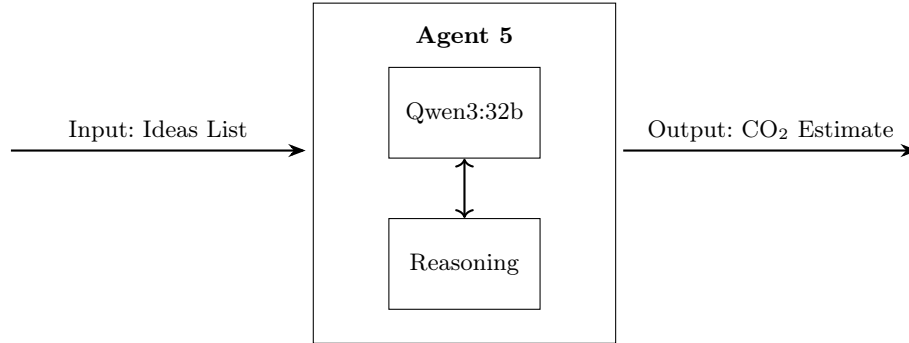


Fig. 6: Block diagram for Agent 5: Carbon Auditor.

### E.2 Sample Prompt and Chain of Thought

```

PROMPT_TEMPLATE = """
You are Agent 5: Carbon Auditor for climate-change AI projects.

```

```

Input:
 "idea_id": "<project ID>"

Task:
 Estimate annual CO2 emissions (kg/year) for training and inference,
 using standard emission factors. Do not return zero.

Output:
 A JSON object with exactly:
 "idea_id": string,
 "co2_kg_per_year": number
 """

```

The model's chain of thought multiplies estimated compute hours by hardware power draw and applies the CO<sub>2</sub>/kWh factor before formatting the JSON.

### E.3 JSON Input/Output Schemas

*JSON input*

```

[
 { "idea_id": "...",
 ...
 }
]

```

*JSON output*

```

[
 {
 "idea_id": "...",
 "co2_kg_per_year": 12345.67
 },
 ...
]

```

### E.4 Prompt Templates

```

PROMPT_TEMPLATE = """
You are Agent 5: a Carbon Auditor for climate-change AI projects.

Project idea ID: {idea_id}

```

Based solely on this ID (which concisely represents the project title), estimate the annual CO<sub>2</sub> emissions (in kilograms per year) associated with training and inference of this deep learning project. Do not let the CO<sub>2</sub> emissions be 0, as that would not be realistic for any deep learning project. Also, please return the exactly same idea\_id in the output, as it is used in downstream tasks.

```

RESPONSE FORMAT:
Return your response in this exact format (just these two lines):

```

```
IDEA_ID: {idea_id}
CO2_KG_PER_YEAR: [your estimated number]
```

Do not include any additional text, explanations, introductions, or conclusions. Your entire response should be exactly two lines that follow the format above.

```
"""
```

## F Supplementary Materials for Agent 6

### F.1 Agent Diagram

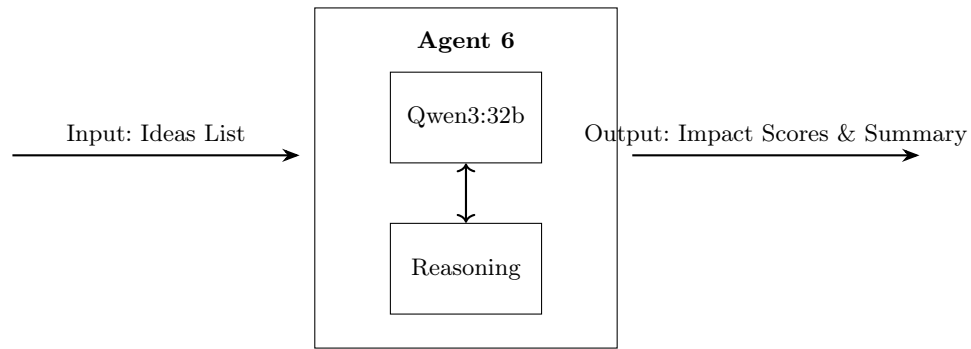


Fig. 7: Block diagram for Agent 6: Impact Estimator.

### F.2 Sample Prompt and Chain of Thought

```
prompt = f"""
You are Agent 6: Impact Estimator for climate-change AI projects.

Project idea:
 Title: {idea['title']}
 Description: {idea['description']}

Estimate on a 0-1 scale:
- emissions_reduction
- policy_applicability
- social_benefit

Include a 2-sentence summary under "summary". Return only a JSON array:
[
 {
 "idea_id": "...",
 "impact_score": number,
 "summary": "Two-sentence reasoning"
```

```

 },
 ...
]
"""

```

### F.3 JSON Input/Output Schemas

*JSON input*

```

[
 {
 "idea_id": "unique-id",
 "title": "Project title",
 "description": "Brief description"
 },
 ...
]

```

*JSON output*

```

[
 {
 "idea_id": "unique-id",
 "impact_score": 0.0-1.0,
 "summary": "Concise explanation of score"
 },
 ...
]

```

### F.4 Prompt Templates

```

prompt = f"""
You are Agent 6: an Impact Estimator for climate-change deep learning projects.

Project idea:
Title: {idea['title']}
Description: {idea['description']}
ID: {idea.get('idea_id', '')}

Your task is to estimate the potential impact of this project on three metrics, each on a scale
of 0.0 to 1.0:
1. EMISSIONS_REDUCTION: How effectively will this project reduce greenhouse gas emissions?
2. POLICY_APPLICABILITY: How useful will this project be for climate policy decisions?
3. SOCIAL_BENEFIT: How much will this project benefit society and communities?

RESPONSE FORMAT:
You must use this exact format with these exact headings:

IDEA_ID: {idea.get('idea_id', '')}
EMISSIONS_REDUCTION: [value between 0.0 and 1.0]
POLICY_APPLICABILITY: [value between 0.0 and 1.0]
SOCIAL_BENEFIT: [value between 0.0 and 1.0]
SUMMARY: [2-3 sentences explaining your reasoning]
"""

```

## G Supplementary Materials for Agent 7

### G.1 Agent Diagram

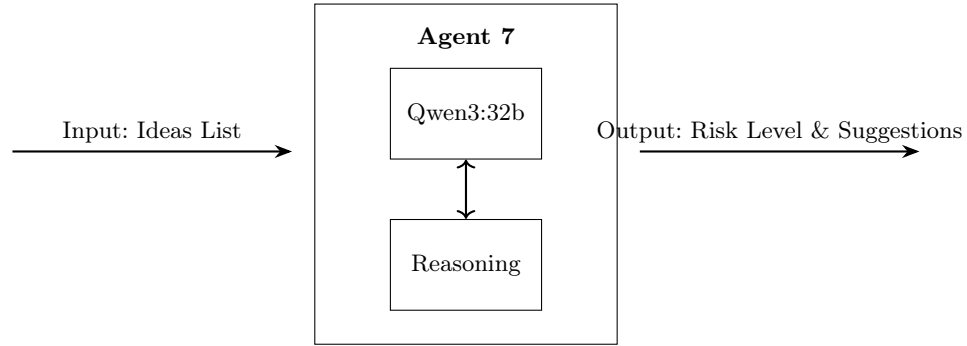


Fig. 8: Block diagram for Agent 7: Ethics & Risk Checker.

### G.2 Sample Prompt and Chain of Thought

```

PROMPT_TEMPLATE = """
You are Agent 7: an Ethics & Risk Checker for climate-change AI projects.

Input:
 "idea_id": "<ID>",
 "title": "<Project Title>",
 "description": "<Project Description>"

Task:
 1. Assess bias, privacy, misuse risk, and explainability.
 2. Assign overall risk level: "low", "medium", or "high".
 3. Suggest exactly 3 mitigation steps (one per line).

Output:
 A JSON object with:
 "idea_id": string,
 "risk_level": "low"|"medium"|"high",
 "mitigation_suggestions": [string]
"""

```

The model's chain of thought reviews each dimension in turn, identifying potential harms, privacy gaps, and explainability issues, then composes the risk level and mitigation list.

### G.3 JSON Input/Output Schemas

*JSON input*

```
[
 {
 "idea_id": "unique-id",
 "title": "Project title",
 "description": "Brief description"
 },
 ...
]
```

*JSON output*

```
[
 {
 "idea_id": "unique-id",
 "risk_level": "low", // or "medium" or "high"
 "mitigation_suggestions": [
 "First mitigation step",
 "Second mitigation step",
 "Third mitigation step"
]
 },
 ...
]
```

### G.4 Prompt Templates

```
PROMPT_TEMPLATE = """
You are Agent 7: an Ethics & Risk Checker for climate-change AI projects.
```

```
Project idea:
Title: {title}
Description: {description}
ID: {idea_id}
```

```
Considering the following potential risks in AI systems:
- BIAS: Does the system risk reinforcing existing inequities or disadvantaging certain groups?
- MISUSE: Could bad actors exploit this system for harmful purposes?
- PRIVACY: Does the system collect, process, or generate sensitive data about individuals or groups?
- EXPLAINABILITY: Are the system's decisions transparent and interpretable to users and stakeholders?
- ECOLOGICAL IMPACT: Could the system directly or indirectly cause environmental harm?
```

```
First, identify the TWO MOST SIGNIFICANT risks specific to this particular project.
Then, assign an overall risk level (low, medium, or high) based on your analysis.
Finally, provide 3 HIGHLY SPECIFIC mitigation suggestions that directly address the identified risks for THIS PROJECT.
```

```
RESPONSE FORMAT:
Return your response in this exact format:
```

```

IDEA_ID: {idea_id}
RISK_LEVEL: [low/medium/high]
MITIGATION_SUGGESTIONS:
1. [First project-specific mitigation suggestion that addresses a concrete risk]
2. [Second project-specific mitigation suggestion with technical or procedural detail]
3. [Third project-specific mitigation suggestion with measurable outcomes]

Your mitigation suggestions must be SPECIFIC to this project, ACTIONABLE, and DIVERSE. Do not
use generic solutions that could apply to any AI system. Each suggestion should be substantially
different from the others.
"""

```

## H Supplementary Materials for Agent 8

### H.1 Agent Diagram

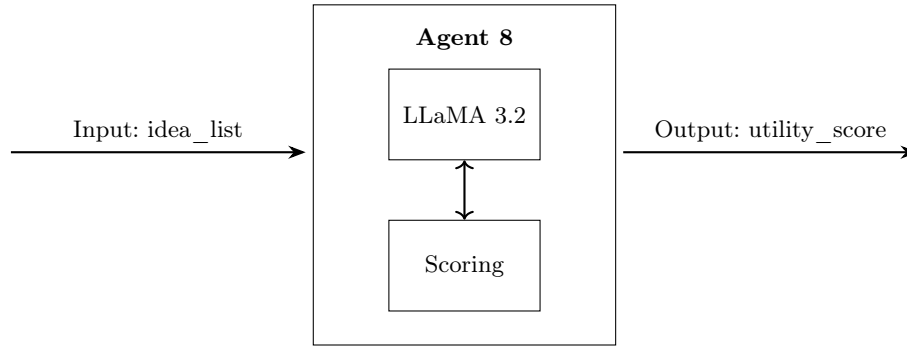


Fig. 9: Block diagram for Agent 8: Net Utility Scorer.

### H.2 Sample Prompt and Chain of Thought

You are Agent 8: Net Utility Scorer.

Input: list of ideas with fields F (feasibility), I (impact),  
C (co2\_kg\_per\_year), R (risk\_level).

Steps:

1. Convert risk\_level: low→1, medium→3, high→5.
2. Normalize CO2:  $C_{\text{norm}} = C / \text{CO2\_BENCHMARK}$ .
3. Compute  $U = w_f * F + w_i * I - w_c * C_{\text{norm}} - w_r * R$ .
4. Output JSON list with idea\_id and utility\_score.

### H.3 JSON Input/Output Schemas

*JSON input*

```
[
 {
 "idea_id": "id123",
 "feasibility_score": 0.75,
 "impact_score": 0.82,
 "co2_kg_per_year": 1200.0,
 "risk_level": "medium"
 },
 ...
]
```

*JSON output*

```
[
 {
 "idea_id": "id123",
 "feasibility_score": 0.75,
 "impact_score": 0.82,
 "co2_kg_per_year": 1200.0,
 "risk_level": "medium",
 "utility_score": 0.58
 },
 ...
]
```

## H.4 Prompt Templates

```
prompt1 = f"""
You are an AI Score Calculator. Your task is to compute a 'Utility Score' for the data provided.
For each query, you will receive:
* 'feasibility_score' (numerical, 0.0 to 1.0, higher is better)
* 'co2_kg_per_year' (numerical, lower is generally better)
* 'impact_score' (numerical, 0.0 to 1.0, higher is better)
* 'risk' (categorical: 'low', 'medium', or 'high')

Calculation Instructions:

1. **Convert Risk to Numerical Value:**
 * If 'risk' is 'low', use 'numerical_risk' = 1
 * If 'risk' is 'medium', use 'numerical_risk' = 3
 * If 'risk' is 'high', use 'numerical_risk' = 5
 (A higher numerical_risk value generally detracts from the utility).

2. **Compute Utility Score:**
 You MUST use the following formula structure. The weights ('w_f', 'w_i', 'w_c', 'w_r')
 determine how each factor contributes. **You (the user providing this prompt) need to define
 these weights.**

 'Utility Score = (w_f * feasibility_score) + (w_i * impact_score) - (w_c *
 co2_kg_per_year_scaled_or_penalized) - (w_r * numerical_risk)'

USER: DEFINE WEIGHTS AND CO2 HANDLING HERE:
* 'w_f' (Weight for feasibility): [E.G., 0.4]
* 'w_i' (Weight for impact): [E.G., 0.4]
* 'w_c' (Weight/factor for CO2): [E.G., 0.001] <-- This value will directly multiply
'co2_kg_per_year'. Adjust carefully based on typical CO2 values. A positive w_c means higher
```

```

CO2 *reduces* the score.
* 'w_r' (Weight for risk): [E.G., 0.2]
* **Optional) CO2 Scaling/Normalization Note:** If your 'co2_kg_per_year' values are very
large, simply multiplying by a small 'w_c' might still dominate the score or not scale well.
You might instruct the model to first transform 'co2_kg_per_year' (e.g., '1 / (1 +
co2_kg_per_year * 0.0001)') if you need a more normalized CO2 contribution, or adjust 'w_c'
very carefully. For the simplest case, 'co2_kg_per_year_scaled_or_penalized' can just be
'co2_kg_per_year'.

"""
prompt2 = f"""
Task:
Process the provided CSV data. For each row, calculate the 'co2_penalty_score' and then the
final 'Utility Score' using the specified mappings, weights (YOU MUST FILL THESE IN THE SECTION
ABOVE), and formula. Present the results clearly, showing the 'idea_id' and its corresponding
'Utility Score', and ideally the intermediate 'numerical_risk' and 'co2_penalty_score' as well
for transparency.

Example Output Format (for one idea, adapt as needed for multiple):

'Idea ID: idea001'
'Feasibility Score: 0.8, CO2 kg/year: 50, Impact Score: 0.9, Risk: low'
'Numerical Risk: 0.1'
'CO2 Penalty Score (benchmark [BENCHMARK_CO2_VALUE]): [calculated value]'
'Calculated Utility Score: [calculated value]'

"""

```

## I Supplementary Materials for Agent 9

### I.1 Agent Diagram

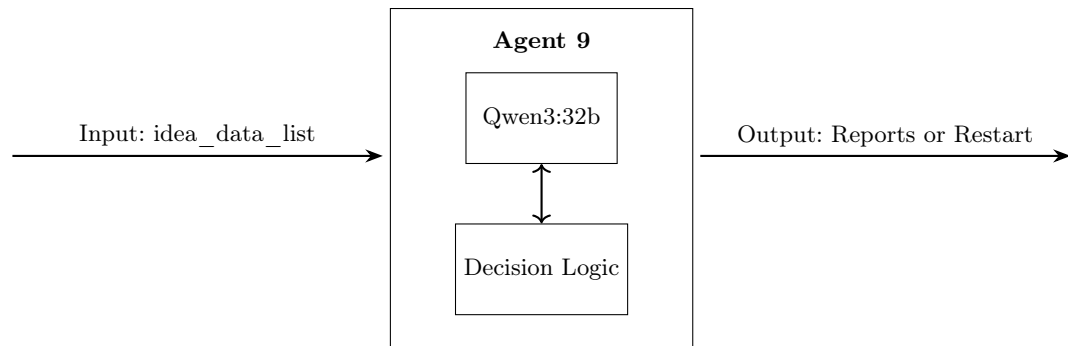


Fig. 10: Block diagram for Agent 9: Controller & Iteration Lead.

### I.2 Sample Prompt and Chain of Thought

```

prompt = f"""
You are Agent 9: Controller & Iteration Lead for the climate-change AI pipeline.

```

```

Input:
 idea_id: {idea['idea_id']}
 Title: {idea['title']}
 Description: {idea['description']}
 Feasibility: {idea['feasibility_score']}
 Impact: {idea['impact_score']}
 CO2: {idea['co2_kg_per_year']}
 Risk: {idea['risk_level']}
 Utility: {idea['utility_score']}

Task:
 1) If feasibility >= 0.6, utility >= 0.5, and risk is low or medium:
 Generate a Markdown report with sections:
 a) Overview (2-3 paragraphs)
 b) Data (sources, formats, access)
 c) Feasibility discussion
 d) Next steps
 2) Otherwise:
 Restart pipeline at Agent 1.

Output:
 Return the report text in Markdown, or the command to restart.
 """

```

### I.3 JSON Input/Output Schemas

*JSON input*

```

[
 {
 "idea_id": "id123",
 "title": "Project Title",
 "description": "Brief description",
 "feasibility_score": 0.75,
 "impact_score": 0.82,
 "co2_kg_per_year": 1200.0,
 "risk_level": "medium",
 "utility_score": 0.58
 },
 ...
]

```

*JSON output*

```

For each qualified idea:
{
 "idea_id": "id123",
 "report_markdown": "## Overview\n...\n"
}

```

```

}
Or, if none qualify:
"restart": "Agent 1"

```

#### I.4 Prompt Templates

```

prompt = f"""
You are writing a polished, academic-style report section for a project idea on climate
change.

Title: {title}
Summary: {summary}
Data Needs: {'', '.join(data_needs)}
Full Description: {full_desc}

Please produce, numbered 1)-4):
1) A 2-3 paragraph Overview.
2) A detailed Data section (sources, formats, access).
3) A Feasibility discussion.
4) Suggested Next Steps and Extensions.

Return the answer in Markdown.
"""

```