Appendix

1 Implementation Details

Table 1. Pipeline hyperparameters, thresholds, and model configurations.

Component	Parameter	Value	Notes
§3.1: Knowledge integration			
Master lexicon retrieval	Embedding model	Sentence-BERT[a]	Vector retrieval
	Top-K candidates	5	Per query skill phrase
	Similarity threshold	≥ 0.65	Accept canonical match
	LLM (temperature)	GPT-4o ($T = 0.02$)	Normalisation / tie-break
Domain ontology grounding	Embedding model	Sentence-BERT[a]	GCMD/SWEET concept retrieval
	Top-K candidates	10	Per query skill phrase
	Similarity threshold	≥ 0.65	Accept concept grounding
	Max concepts retained	3	Per skill phrase
	LLM (temperature)	GPT-4o ($T = 0.0$)	Concept selection/ranking
§3.2: Skill extraction			
LLM extraction	LLM (temperature)	GPT-4o ($T = 0.2$)	Extract $S_{\text{pre}}, S_{\text{out}}$
	Prompting	3-shot	Fixed exemplars
	Evidence format	Character offsets	Extractive spans in source text
§3.3: Graph construction			
Prerequisite inference	Candidate pool Top-K	15	Skill-overlap retrieval
	Edge threshold	≥ 0.2	Retain inferred links
	Temporal constraint	Eq.1	Year/term eligibility
LLM audit	LLM (temperature)	GPT-4o ($T = 0.2$)	Keep/drop/unsure
Taxonomy assignment	L1 categories	5	Number of top-level classes
	Similarity threshold	≥ 0.85	Assign taxonomy label
§3.4: Pathway planning			
Plan generation	LLM (temperature)	GPT-4o ($T = 1.0$)	Explanation / ranking
	Credit budget	60 ECTS	Per academic year
	Planning horizon	2 terms	Default lookahead
	Interface	Gradio	Prototype UI

[a] Sentence-BERT used for embedding-based retrieval (cosine similarity).

2 Skill extraction with evidence spans: Explicit versus implicit

2.1 Explicit skill inference example

Example: Linear Algebra module

Module specification text.

```
MATH301: Linear Algebra

Description: This module introduces matrix operations, eigenvalue
decomposition, and vector spaces. Students will learn to solve
systems of linear equations using Gaussian elimination and compute
determinants. Prerequisites include introductory calculus and
basic algebraic manipulation.

Learning outcomes: Upon completion, students will be able to
perform eigenvalue calculations, apply singular value decomposition,
and interpret geometric transformations.
```

Extracted skills with evidence spans. The extraction pipeline identifies the following skills with character-level evidence:

```
Prerequisite skills:
{
  "skill": "introductory calculus",
  "evidence": {
    "start": 278,
    "end": 300,
    "text": "introductory calculus",
    "source_doc": "MATH301_spec.txt"
  }
}

{
  "skill": "basic algebraic manipulation",
  "evidence": {
    "start": 305,
    "end": 333,
    "text": "basic algebraic manipulation",
    "source_doc": "MATH301_spec.txt"
  }
}

Outcome skills:
{
```

```
  "skill": "matrix operations",
  "evidence": {
    "start": 45,
    "end": 62,
    "text": "matrix operations",
    "source_doc": "MATH301_spec.txt"
  }
}

{
  "skill": "eigenvalue decomposition",
  "evidence": {
    "start": 64,
    "end": 87,
    "text": "eigenvalue decomposition",
    "source_doc": "MATH301_spec.txt"
  }
}

{
  "skill": "Gaussian elimination",
  "evidence": {
    "start": 174,
    "end": 194,
    "text": "Gaussian elimination",
    "source_doc": "MATH301_spec.txt"
  }
}
```


2.2 Implicit skill inference example

Implicit prerequisites also extracted. For the same MATH301 module, our pipeline also extracts the following implicit skills without direct textual evidence:


```
{
  "skill": "mathematical notation and symbols",
  "evidence": null,
  "type": "implicit",
  "reasoning": "Required for matrix operations and eigenvalue
  decomposition, but assumed rather than stated."
}

{
  "skill": "abstract mathematical reasoning",
  "evidence": null,
  "type": "implicit",
```

```
  "reasoning": "Needed for vector spaces but not explicitly
  mentioned as prerequisite."
}

{
  "skill": "function composition",
  "evidence": null,
  "type": "implicit",
  "reasoning": "Essential for geometric transformations but
  not stated in prerequisites."
}

{
  "skill": "basic arithmetic operations",
  "evidence": null,
  "type": "implicit",
  "reasoning": "Required for determinants and linear equations
  but assumed as universal background."
}
```


2.3 Design rationale and implicit inference mechanism.

Extracting both explicit skills (with evidence) and implicit skills (without evidence) provides comprehensive prerequisite coverage while maintaining transparency about extraction confidence through type labels. Implicit skills are inferred by the LLM based on contextual analysis of module content and learning outcomes. For example, when a module teaches eigenvalue decomposition and matrix operations, the LLM can reasonably infer that mathematical notation is a prerequisite even if not explicitly stated, as these topics cannot be understood without notation proficiency. Similarly, abstract mathematical reasoning is inferred as necessary for vector spaces because this topic inherently requires conceptual thinking beyond concrete computation.

This two-tiered extraction approach addresses a common limitation in curriculum specifications: educators often assume foundational knowledge without listing it explicitly, creating gaps in prerequisite documentation. By extracting both explicit and implicit skills, our pipeline captures the complete prerequisite landscape while preserving evidence traceability for explicit skills. The type label allows downstream applications to handle the two categories differently. For instance, pathway planning tools can prioritize explicit prerequisites while flagging implicit ones for student self-assessment, or curriculum designers can identify commonly assumed skills that might benefit from explicit instruction.

During graph construction, implicit prerequisites are cross-validated against module outcomes across the curriculum. If an implicit prerequisite extracted for Module A appears as an explicit learning outcome in an earlier Module B, this provides indirect confirmation of the inference. This cross-module consistency

Table 2. Overall knowledge graph construction statistics. Grounding coverage is computed over unique canonical skills ($n = 1{,}593$). Evidence preservation is computed over extracted module–skill relations ($n = 1{,}735$). Stage 2 subrows (SWEET/GCMD) are not mutually exclusive; the Stage 2 union is reported in the first Stage 2 row.

Metric	Count	%
Module-level statistics		
Total modules processed	108	–
Skill extraction statistics ($n = 1{,}735$ edges)		
Total module–skill relations	1,735	–
Prerequisite skill relations	849	48.9
Outcome skill relations	886	51.1
Unique canonical skills identified	1,593	–
Grounding coverage ($n = 1{,}593$ unique skills)		
Grounded via Stage 1 (master lexicon)	263	16.5
Grounded via Stage 2 (GCMD/SWEET; union)	1,218	76.5
SWEET ontology	884	55.5
GCMD keywords	346	21.7
SWEET ∩ GCMD	12	0.8
Overall grounded (Stage 1 ∪ Stage 2)	1,481	93.0
Ungrounded (candidates for expansion)	112	7.0
Evidence preservation ($n = 1{,}735$ edges)		
Relations with character-level evidence spans	1,547	89.2
Average evidence span length (chars)	127.3	–

check helps reduce hallucination risk inherent in implicit skill inference while maintaining comprehensive prerequisite coverage.

3 Overall KG extraction and grounding statistics

4 Inter-Annotator Agreement Study Details

To rigorously assess extraction quality, we employed two domain experts with expertise in Earth Science curriculum design to independently annotate 265 randomly sampled module-skill relations. Each annotator evaluated:

1. **Extraction correctness (Q1):** Is the skill mentioned or reasonably implied?
2. **Edge type accuracy (Q2):** Is the prerequisite/outcome label correct?
3. **Evidence quality (Q3):** Is the evidence span sufficient?
4. **Master lexicon grounding (Q4):** For grounded skills, is the master lexicon grounding accurate?
5. **Concept ontology grounding (Q5):** For grounded skills, is the concept ontology grounding accurate?

Two domain experts with experience in Earth Science curriculum design independently evaluated a random sample of 265 extracted module–skill relations using the rubric. Annotator 1 completed Q1–Q3 for all 265 relations and serves as the primary annotator reported in the main paper. Annotator 2 additionally evaluated Stage 2 concept grounding (Q5) on the concept-grounded subset within the same sample ($n = 193$), enabling cross-annotator comparison of grounding judgements; metrics for Q5 exclude cases marked *Unclear*.

5 Baseline Comparison

Methods

Baseline 1: Keyword Matching (KW). We treat the *master lexicon* as a closed candidate set of canonical skills. For each module specification, KW detects skills by direct string matching between lexicon labels and the module text. To reduce spurious matches from single tokens, we compute Jaccard similarity over token sets and retain matches with $\text{sim}_{\text{KW}} > 0.1$. Module–module prerequisite edges are then inferred if two modules share at least one matched canonical skill and satisfy the temporal constraints (Eq. 1). KW does not generate relation-level extractive rationales beyond token overlap.

Baseline 2: Zero-shot LLM (ZS-LLM). We prompt GPT-4o to (i) extract prerequisite and outcome skills from each module text, and (ii) infer module–module prerequisite relations directly. ZS-LLM outputs skills as free-form strings and does *not* provide evidence spans, grounding to canonical IDs, or constraint/DAG validation at inference time. For evaluation of grounding-related metrics only, we apply a post-hoc mapping step that aligns ZS-LLM free-form skill strings to the master lexicon using the same string/embedding matching rule as in our pipeline and then submit the mapped results to expert validation.

Experimental Setup Test set: 50 randomly sampled modules (Years 1–4: 12/14/13/11). All methods process identical inputs and are evaluated under the same expert validation protocols (Sec. 4.1–4.2).

Metrics. *Extraction* precision/recall/F1 are computed as micro-averaged scores over module–skill edges against expert-validated *canonical skill IDs* (a prediction is correct iff it matches the gold canonical ID for that module). *Grounding coverage* is the fraction of extracted edges that can be assigned a canonical ID; *grounding accuracy* is the fraction of grounded edges judged correct by experts. *Inference precision* is computed over inferred module–module prerequisite edges. *Temporal violation rate* is the fraction of inferred prerequisite edges that violate year/term constraints (Eq. 1). *Evidence coverage* is the fraction of extracted relations accompanied by a character-level supporting span in the source text.

Table 3. Baseline comparison on the 50-module test set. Bold indicates best. †Edges violating year/term constraints (Eq. 1). ‡Fraction of extracted relations accompanied by character-level supporting spans. For KW, token overlap does not constitute relation-level supporting evidence under this definition.

Method	Extraction	Grounding	Inference	Temporal	Evidence‡
	P / R / F1	Cov. / Acc.	Precision	Viol.†	Coverage
KW	0.42 / 0.81 / 0.55	– / –	0.38	0.0%	0%
ZS-LLM	0.68 / 0.72 / 0.70	43% / 62%	0.51	18.2%	0%
Ours	**0.86 / 0.84 / 0.85**	**93% / 87%**	**0.71**	**0.0%**	**89%**

Results Table 3 shows that our method improves extraction quality (F1: 0.85 vs. 0.70), grounding coverage (93% vs. 43%), and inference precision (0.71 vs. 0.51), while providing relation-level evidence traceability (89% character-level span coverage).

Key Findings. KW achieves high recall (0.81) but low precision (0.42) due to lexical overlap noise, yielding low prerequisite precision (0.38). ZS-LLM improves extraction (F1: 0.70) and inference (precision: 0.51) but often conflates prerequisite and outcome skills and produces 18.2% temporal violations; its free-form outputs also reduce post-hoc mapping performance (43% grounding coverage; 62% accuracy). The higher KW recall is expected: under a low similarity threshold, keyword matching over-generates candidates, increasing recall at the expense of precision, whereas ZS-LLM is more selective and further loses matches during post-hoc mapping to canonical IDs. In contrast, our structured prompting and candidate shortlisting improve extraction (F1: 0.85), two-tier grounding (ESCO/QAA; GCMD/SWEET) increases coverage/accuracy to 93%/87%, and constraint validation eliminates temporal violations (0%) while raising inference precision to 0.71.

6 Generalization and extensibility

To assess extensibility beyond Earth Science, we applied our pipeline to a small Physics sample (16 modules) from the same institution. We retained the Stage 1 master lexicon (ESCO/QAA/A-level/MSC) and replaced Stage 2 domain resources with Physics-oriented vocabularies: PhySH (Physics Subject Headings) and QUDT (quantities/units), with SSN/SOSA optionally covering measurement and observation concepts. We used the same retrieval–and–constrained-selection grounding procedure, enabling discipline-specific alignment without GCMD/SWEET.

Extraction quality was comparable across disciplines (Physics: 87.4% vs. Earth Science: 86.2%). Stage 1 grounding covered 34.6% of Physics skills, indicating substantial normalisation to transferable vocabularies. Stage 2 grounding (PhySH/QUDT) captured Physics-specific concepts (e.g., subfields and for-

malisms), improving overall grounding coverage and reducing the ungrounded tail (84% overall grounded coverage). These results suggest the two-tier grounding strategy transfers when a discipline-appropriate subject vocabulary is available for Stage 2 alignment.

These results demonstrate that our two-tier architecture is extensible to other STEM disciplines with minimal modification: the master lexicon provides broad coverage of transferable skills, while domain-specific ontologies can be swapped to capture disciplinary competencies. Domains with mature ontologies can achieve high grounding coverage, while emerging or interdisciplinary fields may require custom ontology curation or rely primarily on Stage 1 grounding.