DATASTAMP: DOMAIN-SPECIFIC DATA SYNTHESIS VIA STRUCTURAL PROJECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) achieve strong performance on general-purpose tasks through instruction tuning, yet high-quality instruction-response pairs remain scarce in specialized domains such as humanities and medicine, where typically only raw, unlabeled documents are available. The central challenge is synthesizing pairs that are both *grounded* in target documents and *diverse* in reasoning patterns. Existing approaches either bootstrap from seed examples via LLM generation, which defaults to shallow and repetitive questions, or rely on manually engineered templates, which offer limited diversity and fail to transfer across domains. We observe that high-quality QA data spanning diverse domains has already been produced by prior research, and that each pair implicitly encodes a domain-free reasoning pattern. Inspired by meta-learning, we propose DATASTAMP, which treats these patterns as transferable prototypes and adapts them to new target domains. Each document is abstracted into a *logic graph* that retains only its logical skeleton; from existing cross-domain QA data, DATASTAMP extracts *transfer graphs*, typed-graph prototypes encoding how a logical structure should be transformed into an instruction-response pair. Given a target document, the most structurally compatible transfer graphs are retrieved via a relational GCN and projected onto the target content, producing grounded and diverse pairs without domain-specific engineering. Evaluated on two domains, DATASTAMP outperforms most open instruction datasets and produces data that is substantially harder and more diverse than existing baselines. Code and data are available at https://anonymous.4open.science/r/DataStamp-03A4.

1 INTRODUCTION

Instruction tuning has become central to modern LLM capabilities, with large-scale data synthesis yielding substantial improvements in general-purpose domains such as mathematics and code generation (Wang et al., 2023; Ouyang et al., 2022; Köpf et al., 2023; Toshniwal et al., 2025). Adapting LLMs to specialized fields such as medicine, law, and the humanities, however, remains difficult: domain practitioners typically possess raw document collections but lack labeled instruction–response pairs, making automated synthesis from such documents a practical necessity. A law firm, for instance, may hold thousands of legal memoranda and case briefs yet have no curated instruction data with which to fine-tune an LLM and few public legal instruction datasets exist to fill that gap. Because the quality and diversity of synthesized data directly govern downstream performance (Liu et al., 2026; Xu et al., 2024; Honovich et al., 2023), the central challenge is to generate instruction data for data-scarce domains that is both grounded in the target documents and diverse in the reasoning patterns it elicits.

Early work addresses this challenge through *seed expansion*: a small set of human-curated or LLM-generated instructions is iteratively rewritten and augmented to grow a larger pool (Wang et al., 2023; Xu et al., 2024; Luo et al., 2024; 2025; Zeng et al., 2024; Yu et al., 2025; Jung et al., 2025; Havrilla et al., 2025). Because seed-expansion methods bootstrap from limited examples alone, they are not grounded in any target document collection and thus cannot absorb domain-specific knowledge directly from text. More recent pipelines therefore condition synthesis on source documents: a teacher LLM reads a raw document and produces instruction–response pairs, optionally steered by predefined templates or rubrics (Yue et al., 2024; Nayak et al., 2024; Jiang et al., 2025; Zhu et al., 2025; Liu et al., 2026; Fan et al., 2026; Ziegler et al., 2025; Cao et al., 2025; Lee et al., 2025; Wei et al., 2026). Yet this

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

document-conditioned paradigm faces two fundamental barriers. **(i) Domain-entangled guidance.** The templates, rubrics, or knowledge taxonomies that steer generation are typically hand-crafted by domain experts or extracted from existing QA datasets in a specific field. Such guidance inherits the vocabulary, framing, and reasoning conventions of its source domain; when applied to a new field, it transfers poorly, forcing practitioners to repeat the costly design process for every target domain. **(ii) Structural–semantic conflation.** Matching guidance to a target document relies on surface-level similarity (e.g., text embeddings), which captures lexical overlap but not structural compatibility. A legal reasoning pattern and a clinical reasoning pattern may share identical logical structure, but appear distant in embedding space because their vocabularies differ. Conversely, documents that are lexically close may require entirely different reasoning patterns. Both barriers share a common root: existing methods never decouple the *structure* of a reasoning pattern from its *domain content*, preventing guidance from generalizing across fields. These observations motivate the central question of this work:



Given raw documents in a new domain, how can we leverage reasoning patterns already encoded in existing QA data to synthesize high-quality, diverse instruction–response pairs without any domain-specific engineering?

This decoupling insight finds a natural parallel in metric-based meta-learning (Finn et al., 2017; Snell et al., 2017): transferable knowledge can be extracted from diverse source tasks and reused for unseen targets via a shared embedding space. If both a document's logical skeleton and the reasoning pattern that transforms it into a question are represented in a domain-free space, the two barriers above dissolve simultaneously. Therefore, we propose **Data** **S**yn**T**hesis **A**dapted to a target Do**M**ain via Structural **P**rojection (DATASTAMP), which factors every document-to-QA transformation into two domain-free graph objects: a *logic graph* $\mathcal{G}^{\mathrm{L}}$ capturing the logical skeleton of a document, and a *transfer graph* $\mathcal{G}^{\mathrm{X}}$ encoding how that structure is turned into an instruction-response pair. Both are stripped of domain content and live in a shared typed-graph space: transfer graphs are domain-free by construction, resolving the transferability constraint; retrieval compares logic graphs in structural space, guaranteeing coverage regardless of domain distance. Because the two objects are fully decoupled, a single document admits one pair per compatible transfer graph, yielding combinatorial diversity that single-template methods cannot provide. The main contributions of this paper are as follows:

- ❶ We propose DATASTAMP, which transfers domain-free reasoning patterns from existing QA data to raw target documents via typed-graph matching, without labeled target-domain data.
- ❷ We build a reusable library of 189K reasoning prototypes and synthesize diverse instruction datasets for medical & health and humanities & social sciences.
- ❸ Across twelve benchmark evaluations, DATASTAMP achieves the highest average scores in both domains. Ablations support the benefits of structural retrieval and explicit graph guidance.

2 RELATED WORK

Instruction Data Synthesis. Instruction tuning (Ouyang et al., 2022) increasingly uses synthetic data to reduce annotation needs (Wang et al., 2023; Nadăș et al., 2025). Prior work expands seed instructions (Xu et al., 2024; Luo et al., 2024; 2025) or explores task spaces through domain trees and seedless prompting (Wan et al., 2023; Gupta et al., 2024; Xu et al., 2026a). Document-based approaches instead use raw text as a content source, generating instructions for existing human-written passages (Li et al., 2024; Köksal et al., 2024) or constructing new instruction-response pairs from documents (Jiang et al., 2025; Yue et al., 2024; Nayak et al., 2024; Zhu et al., 2025). Retrieval and augmentation can target particular tasks (Ziegler et al., 2025), while quality and faithfulness filtering help retain reliable examples (Yehudai et al., 2024). Document grounding supplies the content for synthesis, but does not by itself specify which relationships to combine or what reasoning to require. This motivates a complementary question: how can reasoning patterns from existing QA data guide the construction of new tasks from target documents?

Structural Guidance and Transfer. Prior synthesis methods use design logics, optimized rubrics, and knowledge hierarchies to guide generation (Liu et al., 2026; Fan et al., 2026; Cao et al., 2025). Graphs also organize concepts and evidence for synthesizing QA pairs or explanations (Wang et al.,

2025; Chen et al., 2025b; Xu et al., 2026b; Chen et al., 2025a), and represent reasoning dependencies or tool interactions for generating instructions and trajectories (Bu et al., 2025; Yin et al., 2025). Reuse is also an established direction: prior work transfers design logics across source texts (Liu et al., 2026) and projects general-purpose data into target domains (Lee et al., 2025). The distinction in DATASTAMP is therefore not the use of graphs or transfer alone, but *what is transferred and how it is matched to new content*. We extract document-to-QA transformation patterns from cross-domain examples and pair each pattern with an abstraction of its source document. Logic graphs represent document structure, while transfer graphs encode the transformation; both omit domain vocabulary. Matching the target document's logic graph against the library selects transfer graphs to instantiate on its content. This paired representation makes document structure the retrieval criterion and transformation logic the reusable object. A pattern extracted in one field can therefore guide synthesis using content from another, without labeled target-domain examples.

3 PRELIMINARIES

Problem formulation. Let $\mathcal{D} = \{d_1, \ldots, d_M\}$ be a corpus of raw, unlabeled documents from a target domain. Given no labeled examples, our goal is to synthesize an instruction-tuning dataset $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$ of instruction–response pairs satisfying two properties: (1) *groundedness*, where every pair is faithfully derived from content in $\mathcal{D}$; and (2) *diversity*, where pairs span a broad range of reasoning patterns and question types. The dataset is used to fine-tune a language model p_θ by minimizing the standard SFT objective:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{i=1}^{N}\sum_{l=1}^{L_i} \log\, p_\theta\big(y_{i,l} \mid x_i,\, y_{i,<l}\big), \tag{1}$$

where $y_{i,l}$ is the l-th token of response y_i, $y_{i,<l}$ denotes all preceding tokens, and L_i is the total response length.

From meta-learning to data synthesis. In meta-learning, tasks are drawn from a distribution $p(\mathcal{T})$ and the goal is to extract transferable knowledge from source tasks to solve unseen targets with minimal task-specific training (Finn et al., 2017; Ren et al., 2018). Metric-based approaches (Vinyals et al., 2016; Snell et al., 2017) learn an embedding ϕ mapping tasks into a shared space where proximity reflects structural similarity, enabling nearest-neighbor inference without gradient updates. We exploit the observation that documents across domains often share underlying logical structure: a causal chain in pharmacology and one in economics have the same abstract form. We abstract each document into a domain-free logic graph that retains only logical relations. Documents with identical structure thus map to the same point in a shared space, yielding exactly the shared representation that metric-based meta-learning requires. Complementing each logic graph, a transfer graph encodes how a document's logical structure should be transferred into an instruction-response pair. Transfer graphs are extracted from previously generated high-quality instruction data spanning diverse domains during "*meta-training*" and serve as *transferable prototypes*: analogous to class prototypes in Prototypical Networks (Snell et al., 2017), each transfer graph generalizes beyond its source document to any structurally compatible logic graph in any target domain.

4 METHOD

In this section, we describe the DATASTAMP pipeline for generating high-quality and diverse instruction data from raw domain documents. As illustrated in Figure 1, DATASTAMP factors every document-to-QA transformation into two domain-free graph objects: a logic graph $\mathcal{G}^{\text{L}}$ capturing the logical skeleton of a document, and a transfer graph $\mathcal{G}^{\text{X}}$ specifying how that structure is transformed into an instruction-response pair. The pipeline proceeds in three phases: ❶ **Structural Abstraction and Library Construction** (§4.1) extracts transfer graphs from cross-domain QA data to build a library of transferable prototypes; ❷ **Structural Retrieval via Graph Embedding** (§4.2) encodes logic graphs with a relational GCN and retrieves compatible transfer graphs through nearest-neighbor search; ❸ **Structural Projection and Verification** (§4.3) applies each retrieved transfer graph to the target document and filters the resulting pairs by factual consistency.

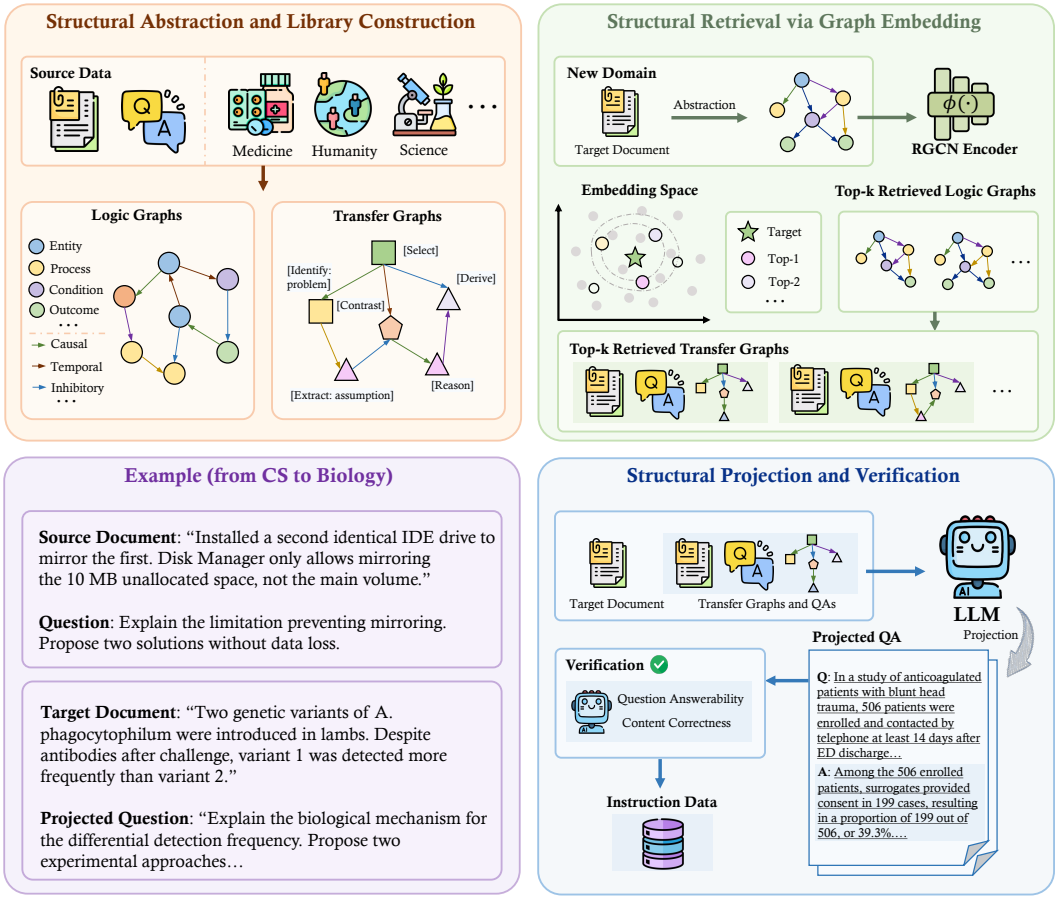


Figure 1: Overview of DATASTAMP.

4.1 STRUCTURAL ABSTRACTION AND LIBRARY CONSTRUCTION

Meta-training in DATASTAMP mirrors prototype extraction in Prototypical Networks: given existing cross-domain QA data as source tasks, we distill each task's transformation pattern into a transfer graph and organize these prototypes into a retrievable library. We begin by formally defining the two core graph objects.

Definition 1 (Logic Graph)**.** A **logic graph** $\mathcal{G}^{\mathrm{L}} = (V, E, \tau_V, \tau_E)$ is a typed directed graph that abstracts a document into its logical skeleton over two fixed type vocabularies:

- **Node types** $\mathcal{T}_V$ (6): ENTITY, PROCESS, PROPERTY, OUTCOME, CONDITION, QUANTITY.
- **Edge types** $\mathcal{T}_E$ (7): CAUSAL, COMPARATIVE, HIERARCHICAL, TEMPORAL, CONDITIONAL, INHIBITORY, COMPOSITIONAL.

Each node carries only its abstract type label with no domain-specific terms. Documents from different domains that share the same logical structure yield isomorphic logic graphs, placing them at the same point in the shared structural space. Beyond representing documents, a key challenge is capturing *how* a document should be converted into a high-quality instruction. Inspired by previous work (Liu et al., 2026), we define the transfer graph to encode this conversion logic.

Definition 2 (Transfer Graph)**.** A **transfer graph** $\mathcal{G}^{\mathrm{X}}$ is a directed graph that encodes the step-by-step reasoning logic by which a document's content is transformed into an instruction-response pair. Each node describes a generic structural operation (e.g., selecting components, reorganizing relationships, imposing a reasoning demand, deriving an answer).

Let $\mathcal{P} = \{(d_j^s, q_j, a_j)\}_{j=1}^{K}$ be a collection of publicly available (document, question, answer) triplets spanning diverse disciplines. For each triplet, we prompt an LLM to extract a paired logic graph

$\mathcal{G}_j^{\mathrm{L}}$ and transfer graph $\mathcal{G}_j^{\mathrm{X}}$, both stripped of domain vocabulary (prompt details in Appendix D). The resulting library,

$$\mathcal{L} = \left\{(d_j^s,\ \mathcal{G}_j^{\mathrm{L}},\ \mathcal{G}_j^{\mathrm{X}},\ q_j,\ a_j)\right\}_{j=1}^{K}, \tag{2}$$

stores each transfer graph alongside the source document d_j^s, the logic graph from which it was derived, and an exemplar QA pair (q_j, a_j) demonstrating how it was instantiated. We illustrate both objects on a real library entry below; the same entry will reappear in §4.3 to demonstrate cross-domain projection.

Example: CS Domain

Document (condensed): An identical IDE drive was added for mirroring, but Disk Manager allows mirroring only the 10 MB unallocated space, not the main volume.
QA: Explain this limitation and propose two solutions without data loss.
Logic graph $\mathcal{G}^{\mathrm{L}}$ (logical skeleton):
$\mathrm{E}_1 \xrightarrow{\text{Caus.}} \mathrm{P}_1 \xrightarrow{\text{Caus.}} \mathrm{O}_1$; $\mathrm{E}_1 \xrightarrow{\text{Cond.}} \mathrm{P}_2 \xrightarrow{\text{Caus.}} \mathrm{O}_2$; $\mathrm{E}_2 \xrightarrow{\text{Comp.}} \mathrm{E}_1$
$\mathrm{P}_2 \xrightarrow{\text{Inh.}} \mathrm{P}_3 \xrightarrow{\text{Caus.}} \mathrm{O}_3$; $\mathrm{O}_1 \xrightarrow{\text{Temp.}} \mathrm{O}_3$
Transfer graph $\mathcal{G}^{\mathrm{X}}$ (condensed node descriptions):
A–G: Select matched units → identify constraints → extract requirements → compare capacity → frame pairing failure → trace rules → derive the matching-capacity requirement.
G → H → J: Propose an intermediate configuration → justify a compliant path.
G → I → K: Propose an external workaround → justify bypassing internal limits.

4.2 Structural Retrieval via Graph Embedding

Given a target document d_t, we construct its logic graph $\mathcal{G}_t^{\mathrm{L}}$ using the same LLM prompting procedure as during library construction, and retrieve the most structurally compatible transfer graphs from $\mathcal{L}^*$. Existing data synthesis methods typically retrieve in-context examples via semantic similarity over raw text embeddings. This is effective when source and target belong to similar domains, but breaks down in two common scenarios: (i) the target domain is entirely new and shares no lexical overlap with the library, and (ii) documents from distant domains happen to share the same underlying reasoning structure despite using completely different terminology. In both cases, semantic retrieval either returns irrelevant matches or misses structurally compatible ones. We address this by retrieving over logic graph embeddings, where proximity reflects logical compatibility.

Encoder. We encode each logic graph with a relational graph convolutional network (R-GCN) (Schlichtkrull et al., 2018), where distinct edge types propagate information through separate weight matrices:

$$\mathbf{h}_v^{(t+1)} = \sigma\Big(\sum_{r\in\mathcal{T}_E}\sum_{u\in\mathcal{N}_v^r}\tfrac{1}{|\mathcal{N}_v^r|}\,\mathbf{W}_r\,\mathbf{h}_u^{(t)} + \mathbf{W}_0\,\mathbf{h}_v^{(t)}\Big). \tag{3}$$

Initial node features are type embeddings from $\mathcal{T}_V$; after T message-passing rounds, the node representations are mean-pooled, projected, and L_2-normalized to produce a graph descriptor $\phi(G) \in \mathbb{R}^{d_o}$. The encoder weights are initialized once and frozen: since retrieval serves as a candidate generator and the downstream verification step (§4.3) prunes false positives, recall matters more than precision at this stage.

Retrieval. We then retrieve the top-k_m library entries whose logic graph embeddings are most similar to the target:

$$\mathcal{M}_t = \underset{j\in\mathcal{L}^*}{\text{top-}k_m}\ \mathrm{sim}\big(\phi(\mathcal{G}_t^{\mathrm{L}}),\ \phi(\mathcal{G}_j^{\mathrm{L}})\big). \tag{4}$$

For each retrieved entry, we look up its associated transfer graph $\mathcal{G}_j^{\mathrm{X}}$, source document d_j^s, and exemplar QA pair (q_j, a_j) from $\mathcal{L}^*$, which together serve as the input to the projection step (§4.3).

4.3 Structural Projection and Verification

The final step completes the meta-learning loop: we instantiate each retrieved transfer graph on the target document to produce candidate instruction-response pairs, verify their factual consistency, and select a diverse subset for the final dataset.

Structural projection. The operator Π applies each retrieved transfer graph $\mathcal{G}_j^{\mathrm{X}}$ to the target document via a node correspondence f_j,

$$(\hat{x}_{t,j},\ \hat{y}_{t,j}) = \Pi\big(d_t,\ \mathcal{G}_j^{\mathrm{X}},\ f_j,\ (q_j, a_j)\big). \tag{5}$$

In practice, Π is an LLM call prompted with the target document d_t, the retrieved transfer graph as a transformation recipe, the node map f_j translating abstract roles to target entities, and the exemplar (q_j, a_j) demonstrating how the transfer graph was previously instantiated on different content. The node correspondence $f_j : V_t \to V_j$ is constructed by greedy type alignment, pairing nodes of the same type in stable order. We revisit the running example from §4.1 to illustrate how a transfer graph originally extracted from a computer science document transfers to a biomedical target.

Projection of Running Example (CS → Biology)

Target (PubMed, condensed): Lambs received two variants of *A. phagocytophilum*; despite antibodies after challenge, variant 1 was detected more often than variant 2.

Projected question (condensed): Explain the difference. Propose and justify two experiments distinguishing immune-mediated clearance from an intrinsic replication advantage.

Transferred pattern: Matched units with asymmetric behavior → diagnose constraints → propose and justify two solutions. Disks become variants; storage fixes become experiments.

Verification. Each projected pair is verified against the target document by an LLM judge $\mathcal{V}(\hat{x}_{t,j},\ \hat{y}_{t,j},\ d_t) \in \{0, 1\}$, which checks two criteria: (1) the question is answerable from the document, and (2) all claims in the answer are supported by it. Pairs failing either criterion are discarded. This step is essential because structural compatibility at the logic graph level does not guarantee that a particular transfer graph will produce a factually grounded pair on a given target; the verification judge serves as the final quality gate.

Diversity selection. After verification, a single target document may yield multiple valid pairs from different retrieved transfer graphs. Retaining all of them would introduce redundancy, as structurally similar transfer graphs can produce near-duplicate questions. To maximize the instructional value of the final dataset, we select a diverse subset from the verified pool: we embed all verified pairs using a sentence encoder, compute pairwise cosine distances, and select pairs that maximize the minimum distance to the already-selected set. This ensures the final dataset contains complementary reasoning patterns rather than repetitive variations. The resulting dataset is:

$$\mathcal{S} = \bigcup_{d_t \in \mathcal{D}} \text{div-select}\Big(\big\{(\hat{x}_{t,j},\ \hat{y}_{t,j}) \ \big|\ \mathcal{V}(\hat{x}_{t,j}, \hat{y}_{t,j}, d_t) = 1,\ (\mathcal{G}_j^{\mathrm{X}}, f_j) \in \mathcal{M}_t\big\}\Big), \tag{6}$$

and is passed directly to supervised fine-tuning via Eq. 1.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate DataStamp on two domains, *Humanities & Social Sciences (HSS)* and *Medical & Health*. For HSS, source documents are drawn from 79 humanities and social science books (full list in Appendix A); for Medical & Health, source documents are PubMed abstracts (Jin et al., 2019). PubMedQA test abstracts are excluded from the medical data-construction pool. The logic graph and transfer graph library is constructed from DLR-Web (Liu et al., 2026), a large-scale multidisciplinary QA collection. All models are fine-tuned from Qwen3-8B-Base (Yang et al., 2025) with identical hyperparameters (full details in Appendix C).

Benchmarks. Following the previous work (Fan et al., 2026), we evaluate models on 12 benchmarks in two domains. For HSS we use MMLU-Pro History (Wang et al., 2024), SuperGPQA History (Du et al., 2025), HLE (HSS) (Phan et al., 2025), BIG-Bench Hard (Suzgun et al., 2023), HellaSwag (Zellers et al., 2019), and DROP (Dua et al., 2019). For Medical & Health we use MMLU-Pro Health (Wang et al., 2024), SuperGPQA Health (Du et al., 2025), HLE (Biology/Medicine) (Phan et al., 2025), HealthBench (Arora et al., 2025), PubMedQA (Jin et al., 2019), and MedQA (Jin et al., 2021). Evaluation follows standard protocols: 5-shot for MMLU-Pro, 1-shot for PubMedQA, and 4-shot for MedQA, and GPT-4o as judge for HLE and HealthBench. Benchmark scores are averaged over five evaluations. Scores, reported averages, and differences are rounded independently to one decimal place for display.

Table 1: Results across two domains. All metrics are accuracy (%, higher is better). **Best** result per column; 2nd best per column, based on the values before display rounding. **Avg**: mean accuracy across six benchmarks; ↑/↓ denotes change vs. Qwen3-8B-Base.

	MMLU-Pro	SuperGPQA	HLE	BBH	HellaSwag	DROP	Avg
Humanities & Social Sciences (HSS)							
Qwen3-8B-Base	32.7	29.2	9.1	43.5	44.3	60.9	36.6
Qwen3-8B-Instruct	50.3 ↑17.6	35.7 ↑6.4	11.4 ↑2.2	**75.3** ↑31.8	**79.5** ↑35.2	68.2 ↑7.3	53.4 ↑16.8
DLR-HSS	36.8 ↑4.1	34.2 ↑4.9	12.5 ↑3.4	65.0 ↑21.5	70.2 ↑25.9	64.4 ↑3.5	47.2 ↑10.5
Condor	46.8 ↑14.1	32.6 ↑3.3	13.2 ↑4.1	73.0 ↑29.5	72.6 ↑28.2	57.8 ↓3.1	49.3 ↑12.7
LongWriter	34.0 ↑1.2	19.5 ↓9.8	8.5 ↓0.7	48.7 ↑5.3	66.3 ↑22.0	41.2 ↓19.7	36.4 ↓0.3
OpenHermes	48.4 ↑15.7	29.1 ↓0.2	10.5 ↑1.4	63.9 ↑20.4	79.5 ↑35.2	61.3 ↑0.4	48.8 ↑12.1
WebRPro	49.7 ↑17.0	29.9 ↑0.7	12.9 ↑3.8	59.6 ↑16.1	75.4 ↑31.1	64.4 ↑3.5	48.7 ↑12.0
WildChat	50.8 ↑18.1	27.1 ↓2.2	14.6 ↑5.5	66.1 ↑22.6	76.2 ↑31.9	62.5 ↑1.6	49.5 ↑12.9
DATASTAMP-HSS	**52.4** ↑19.7	**40.3** ↑11.1	**15.1** ↑5.9	67.6 ↑24.2	77.8 ↑33.4	**69.0** ↑8.1	**53.7** ↑17.1
	MMLU-Pro	**SuperGPQA**	**HLE**	**HealthBench**	**PubMed**	**MedQA**	**Avg**
Medical & Health							
Qwen3-8B-Base	51.0	26.1	13.6	63.0	56.4	57.1	44.5
Qwen3-8B-Instruct	**61.5** ↑10.5	39.4 ↑13.3	14.1 ↑0.5	71.6 ↑8.6	55.7 ↓0.7	59.3 ↑2.2	50.3 ↑5.7
DLR-Med	53.9 ↑2.9	36.1 ↑9.9	10.7 ↓2.9	64.9 ↑1.9	71.7 ↑15.3	28.7 ↓28.4	44.3 ↓0.2
ChatDoctor	48.8 ↓2.2	27.2 ↑1.1	7.3 ↓6.3	31.3 ↓31.7	71.8 ↑15.4	60.0 ↑2.9	41.1 ↓3.5
MedQA	33.7 ↓17.4	37.2 ↑11.1	9.4 ↓4.2	42.9 ↓20.2	70.5 ↑14.1	59.6 ↑2.5	42.2 ↓2.3
Medical-O1	54.9 ↑3.9	32.6 ↑6.4	14.3 ↑0.7	71.4 ↑8.4	72.6 ↑16.2	62.1 ↑5.0	51.3 ↑6.8
Medical-R1-Distill	57.8 ↑6.8	39.8 ↑13.7	11.1 ↓2.4	71.2 ↑8.1	76.2 ↑19.8	61.3 ↑4.2	52.9 ↑8.4
ReasonMed	60.0 ↑9.0	29.5 ↑3.3	15.6 ↑2.1	52.9 ↓10.2	**78.0** ↑21.7	34.5 ↓22.6	45.1 ↑0.5
DATASTAMP-Med	61.1 ↑10.1	**41.4** ↑15.3	**19.6** ↑6.1	**71.7** ↑8.7	76.6 ↑20.2	**64.4** ↑7.3	**55.8** ↑11.3

Baselines and Implementation Details. We compare against Qwen3-8B-Base and Qwen3-8B-Instruct as reference points, plus diverse open instruction corpora fine-tuned on the same base model under identical training settings. HSS baselines: DLR-HSS (Liu et al., 2026), Condor (Cao et al., 2025), LongWriter (Bai et al., 2025), OpenHermes (Teknium, 2023), WebRPro (Jiang et al., 2025), and WildChat (Zhao et al., 2024). Medical baselines: DLR-Med (Liu et al., 2026), ChatDoctor (Li et al., 2023), MedQA-SFT (Jin et al., 2021), Medical-O1 (Chen et al., 2024), Medical-R1-Distill (FreedomIntelligence, 2025), and ReasonMed (Sun et al., 2025). Detailed statistics and descriptions of each baseline are provided in Appendix C.

5.2 Main Results

Table 1 compares the models in both domains. In HSS, DATASTAMP reaches an average score of 53.7, narrowly above Qwen3-8B-Instruct at 53.4 and ahead of WildChat, the strongest open-data baseline, at 49.5. It leads on MMLU-Pro, SuperGPQA, HLE, and DROP, but trails Qwen3-8B-Instruct on BBH and HellaSwag. Relative to the base model, the largest improvement is on HellaSwag, where the score rises from 44.3 to 77.8. The gains therefore extend beyond the knowledge-intensive benchmarks, although they do not yield the best score on every task.

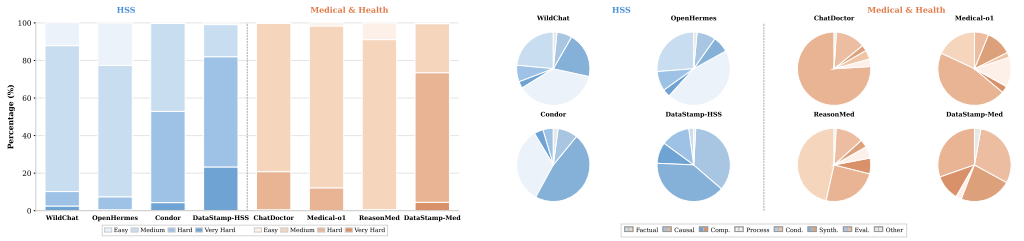
In Medical & Health, DATASTAMP averages 55.8, compared with 52.9 for Medical-R1-Distill and 50.3 for Qwen3-8B-Instruct. It ranks first on SuperGPQA, HLE, HealthBench, and MedQA, and second on the remaining two benchmarks. ReasonMed achieves the highest PubMedQA score, but falls below the base model on HealthBench and MedQA. By comparison, DATASTAMP improves over the base model on all six medical benchmarks, including an increase from 13.6 to 19.6 on HLE. Across both domains, fine-tuning on DATASTAMP data produces higher average scores than the evaluated open-data baselines and remains competitive with Qwen3-8B-Instruct.

5.3 Data Analysis

Diversity. Table 2 measures diversity in two ways: embedding metrics capture variation in semantic content, while lexical metrics capture variation in wording. DATASTAMP-Med ranks first on all five

Table 2: Embedding and lexical diversity of instruction datasets (higher = more diverse on all metrics). Datasets are grouped into HSS and Medical domains. **Best** result per column per domain; 2nd best per column per domain.

Dataset	Embedding Diversity					Lexical Diversity			
	Cos Dist	L2 Dist	1-NN	Inertia	Radius	Uni.	Bi.	Tri.	MTLD
Humanities & Social Sciences (HSS)									
WildChat	0.8048	1.2228	**0.3404**	0.5432	**0.9079**	0.0596	0.2511	0.3994	103.22
OpenHermes	0.8135	1.2446	0.2681	0.4854	0.8897	0.0607	0.2971	0.5324	83.23
Condor	0.7933	1.2446	0.2858	0.5415	0.8803	**0.0870**	0.3804	0.6886	133.57
DATASTAMP-HSS	**0.8179**	**1.2753**	0.2810	**0.5584**	0.9035	0.0676	**0.3845**	**0.7010**	**141.25**
Medical & Health									
ChatDoctor	0.7982	1.2597	0.3258	0.5726	0.8911	0.0311	0.2083	0.5269	122.37
Medical-O1	0.7695	1.2362	0.2881	0.5789	0.8741	**0.0503**	**0.3135**	0.5968	**147.39**
ReasonMed	0.5966	1.0896	0.2364	0.4658	0.7708	0.0339	0.2148	0.4498	70.65
DATASTAMP-Med	**0.8434**	**1.2956**	**0.3503**	**0.6333**	**0.9173**	0.0399	0.2885	**0.6137**	139.30



(a) Difficulty distribution.

(b) Question type distribution.

Figure 2: Difficulty and question type distributions of instruction datasets, grouped into HSS and Medical & Health domains. Higher difficulty entropy and more balanced question types indicate richer instructional coverage.

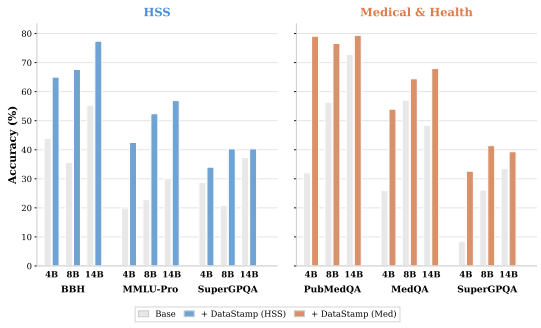
embedding metrics, and DATASTAMP-HSS leads on three of the five; WildChat leads on nearest-neighbor distance and radius. For lexical diversity, DATASTAMP-HSS achieves the highest bigram TTR, trigram TTR, and MTLD. The medical results are more mixed: DATASTAMP-Med has the highest trigram TTR, but Medical-O1 leads on MTLD. Overall, DATASTAMP produces semantically diverse data in both domains, with a more consistent advantage in lexical diversity for HSS.

Difficulty. Following DESIGNER (Liu et al., 2026), we classify questions as Easy, Medium, Hard, or Very Hard based on the complexity and length of reasoning required to answer them, using the few-shot prompt in Appendix D.3. Figure 2a shows a larger proportion of Hard and Very Hard questions in DATASTAMP across both domains, while comparison datasets concentrate more on Medium or Easy questions. These labels reflect estimated reasoning demands, not empirical model error rates.

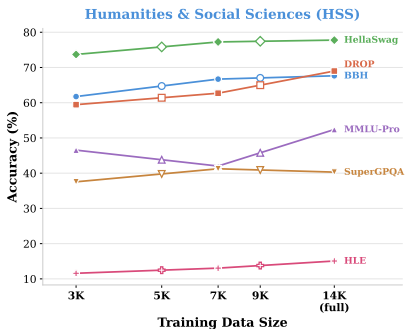
Question type. Figure 2b shows the types of reasoning questions in each dataset. In HSS, DATASTAMP includes substantial proportions of questions that require combining information (Synthesis) or assessing claims (Evaluation). In Medical, many questions involve explaining cause-and-effect relationships (Causal) or making evaluations. By comparison, ChatDoctor focuses mainly on Causal questions, while OpenHermes emphasizes Process questions. These patterns highlight differences in the reasoning tasks emphasized by each dataset.

5.4 ABLATION STUDY AND FURTHER ANALYSIS

Model scaling. Figure 3a compares 4B, 8B, and 14B Qwen3 models. DATASTAMP improves over the corresponding base model at every plotted task–size combination. In HSS, fine-tuned BBH and MMLU-Pro scores increase with model size, whereas SuperGPQA changes little between 8B and 14B. In Medical, MedQA also improves with size, but PubMedQA remains at a similar level across the three fine-tuned models. Gains over the base model need not grow with capacity: on PubMedQA,



(a) Model scaling across 4B, 8B, and 14B.



(b) Data scaling on HSS.

Figure 3: Scaling analysis. (a) Per-benchmark accuracy of DATASTAMP vs. base models at three parameter scales. (b) HSS benchmark accuracy as training data grows from 3K to 14K (full).

the stronger 14B base leaves a smaller gap than at 4B or 8B. These results show benefits across model sizes, but task-dependent returns to additional capacity.

Data scaling. Figure 3b varies the HSS training set from 3K to 14K examples, with intermediate sizes of 5K, 7K, and 9K. BBH and HLE improve throughout this range. HellaSwag gains mainly at the smaller sizes and changes little after 7K, whereas DROP continues to improve through 14K, including a clear increase over 9K. MMLU-Pro declines from 3K to 7K before recovering to its highest score at 14K; SuperGPQA peaks at 7K and then decreases slightly. Overall, 14K yields the best score on five benchmarks and exceeds 3K on all six. Additional data is useful over the tested range, but its marginal benefit varies by benchmark and training-set size.

Ablation study. Table 3 compares four alternatives to the full pipeline on HSS. The full model performs best on all three benchmarks. *Random QA exemplars*, which supplies randomly sampled raw QA pairs without transfer graphs, yields the lowest scores: BBH falls from 67.6 to 58.0, with similar declines on MMLU-Pro and SuperGPQA. Under random selection, supplying *transfer graphs* raises BBH to 62.2 and improves the other two scores, suggesting that explicit transformation structure provides useful guidance beyond raw examples. However, random graphs still trail the full model across all three tasks, supporting the importance of matching the retrieved structure to the target document. *w/o logic graph* replaces structural retrieval with text-embedding similarity. This reduces MMLU-Pro from 52.4 to 46.6 and SuperGPQA from 40.3 to 34.2, while also lowering BBH; semantic similarity does not recover the benefit of structural matching in this comparison. Finally, *w/o verification* removes factual-consistency filtering and produces the smallest decline on each benchmark, retaining a BBH score of 65.4. The consistent but smaller losses suggest an additional benefit from filtering alongside the larger differences associated with graph input and retrieval. Appendix F.2 presents a more comprehensive eight-configuration study across both domains, including controlled selection–input comparisons, adversarial retrieval, and on-the-fly graph generation.

Table 3: Ablation on HSS.

	BBH	MMLU	SGPQA
Full model	**67.6**	**52.4**	**40.3**
Random QA exemplars	58.0	45.1	33.9
w/o logic graph	60.3	46.6	34.2
Random transfer graphs	62.2	47.9	35.6
w/o verification	65.4	50.8	38.7

6 CONCLUSION

We introduced DATASTAMP, which synthesizes instruction data from raw target documents without labeled target-domain data or manually designed reasoning templates. By separating document structure from document-to-QA transformation through logic graphs and transfer graphs, it retrieves and adapts reasoning patterns mined from existing QA data. Across twelve benchmark evaluations in medical & health and humanities & social sciences, DATASTAMP achieves the highest average scores among the evaluated methods. Studies across model sizes and data volumes further examine its scalability. These results support reusing reasoning patterns as prototypes for cross-domain data synthesis, following the retrieve-and-adapt perspective of metric-based meta-learning.

Reproducibility Statement

We describe the DATASTAMP pipeline in Section 4 and provide additional implementation details in the appendix, including dataset sources, graph construction and retrieval settings, synthesis prompts, fine-tuning hyperparameters, and evaluation protocols. Appendix D.1 presents the prompts used for data synthesis, and Appendix D.2 provides the evaluation prompts.

AI Use Statement

We used LLMs to assist with language editing, including improving clarity, grammar, and concision. The authors reviewed all LLM-assisted content and take full responsibility for the research design, implementation, experimental results, analyses, and final manuscript.

References

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longwriter: Unleashing 10,000+ word generation from long context LLMs. In *ICLR*, 2025.

Tianpeng Bu, Minying Zhang, Hongtao Duan, Shurui Li, Lulu Hu, and Yu Li. Enhanced data synthesis for LLM through reasoning structures generated by hierarchical GFlowNet. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15931–15958. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.821. URL https://aclanthology.org/2025.findings-acl.821/.

Maosong Cao, Taolin Zhang, Mo Li, Chuyu Zhang, Yunxin Liu, Haodong Duan, Songyang Zhang, and Kai Chen. Condor: Enhance LLM alignment with knowledge-driven data synthesis and refinement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22392–22412. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1091. URL https://aclanthology.org/2025.acl-long.1091/.

Hanzhu Chen, Xu Shen, Jie Wang, Zehao Wang, Qitan Lv, Junjie He, Rong Wu, Feng Wu, and Jieping Ye. Knowledge graph finetuning enhances knowledge manipulation in large language models. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e44337573fcac83f219c8effa4ebf90d-Abstract-Conference.html.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. HuatuoGPT-o1, towards medical complex reasoning with LLMs. *arXiv preprint arXiv:2412.18925*, 2024. URL https://arxiv.org/abs/2412.18925.

Zihong Chen, Wanli Jiang, Jinzhe Li, Zhonghang Yuan, Huanjun Kong, Wanli Ouyang, and Nanqing Dong. GraphGen: Enhancing supervised fine-tuning for LLMs with knowledge-driven synthetic data generation. *arXiv preprint arXiv:2505.20416*, 2025b. doi: 10.48550/arXiv.2505.20416. URL https://arxiv.org/abs/2505.20416.

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL*, 2019.

Zhiting Fan, Ruizhe Chen, Tianxiang Hu, Ru Peng, Zenan Huang, Haokai Xu, Yixin Chen, Jian Wu, Junbo Zhao, and Zuozhu Liu. Optimsyn: Influence-guided rubrics optimization for synthetic data generation. In *ICLR*, 2026.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

FreedomIntelligence. Medical-R1-Distill-Data. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/FreedomIntelligence/Medical-R1-Distill-Data. Released February 22, 2025; accessed September 23, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783v3.

Himanshu Gupta, Kevin Scaria, Ujjwala Anantheswaran, Shreyas Verma, Mihir Parmar, Saurabh Arjun Sawant, Chitta Baral, and Swaroop Mishra. TarGEN: Targeted data generation with large language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=gpgMRWgv9Q.

Alex Havrilla, Edward Hughes, Mikayel Samvelyan, and Jacob Abernethy. Sparq: Synthetic problem generation for reasoning via quality-diversity algorithms. *arXiv preprint arXiv:2506.06499*, 2025.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In *ACL*, 2023.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Yuxin Jiang, Yufei Wang, Chuhan Wu, Xinyi Dai, Yan Xu, Weinan Gan, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, and Wei Wang. Instruction-tuning data synthesis from scratch via web reconstruction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6603–6618. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.343. URL https://aclanthology.org/2025.findings-acl.343/.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 2021.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP*, 2019.

Jaehun Jung, Seungju Han, Ximing Lu, Skyler Hallinan, David Acuna, Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. Prismatic synthesis: Gradient-based data diversification boosts generalization in LLM reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3032. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/82a34882f560db982d1257b5af461605-Abstract-Conference.html.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. LongForm: Effective instruction tuning with reverse instructions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7056–7078. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.414. URL https://aclanthology.org/2024.findings-emnlp.414/.

Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. In *NeurIPS*, 2023.

Seungyoo Lee, Giung Nam, Moonseok Choi, Hyungi Lee, and Juho Lee. PANGEA: Projection-based augmentation with non-relevant general data for enhanced domain adaptation in LLMs. In *NeurIPS*, 2025.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason E Weston, and Mike Lewis. Self-alignment with instruction backtranslation. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/0f8e3534eb8dee7478d4dc0e9d9a0b1a-Abstract-Conference.html.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 2023.

Weize Liu, Yongchi Zhao, Yijia Luo, Mingyu Xu, Jiaheng Liu, Yanan Li, Xiguo Hu, Zhiqi Bai, Yuchi Xu, Wenbo Su, and Bo Zheng. DESIGNER: Design-logic-guided multidisciplinary data synthesis for LLM reasoning. In *ICLR*, 2026.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *ICLR*, 2025.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. In *ICLR*, 2024.

Mihai Nadăș, Laura Dioșan, and Andreea Tomescu. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*, 2025.

Nihal Nayak, Yiyang Nan, Avi Trost, and Stephen Bach. Learning to generate instruction tuning datasets for zero-shot task adaptation. In *ACL Findings*, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018. URL https://arxiv.org/abs/1803.00676.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pp. 593–607. Springer, 2018.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

Yu Sun, Xingyu Qian, Weiwen Xu, Hao Zhang, Chenghao Xiao, Long Li, Deli Zhao, Wenbing Huang, Tingyang Xu, Qifeng Bai, and Yu Rong. ReasonMed: A 370K multi-agent generated dataset for advancing medical reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 26446–26467. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1344. URL https://aclanthology.org/2025.emnlp-main.1344/.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants, 2023. URL https://huggingface.co/datasets/teknium/OpenHermes-2.5.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating AI for math with massive open-source instruction data. In *ICLR*, 2025.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016.

Fanqi Wan, Xinting Huang, Tao Yang, Xiaojun Quan, Wei Bi, and Shuming Shi. Explore-instruct: Enhancing domain-specific instruction coverage through active exploration. In *EMNLP*, 2023.

Jiankang Wang, Jianjun Xu, Xiaorui Wang, Yuxin Wang, Mengting Xing, Shancheng Fang, and Hongtao Xie. GRIP: A graph-based reasoning instruction producer. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2577. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/6f1989abe9562c5cd306e070725fe0a3-Abstract-Conference.html.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *ACL*, 2023.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *NeurIPS*, 2024.

Yifan Wei, Li Du, Xiaoyan Yu, Yang Feng, and Angsheng Li. Towards compositional generalization of llms via skill taxonomy guided data synthesis. *arXiv preprint arXiv:2601.03676*, 2026.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. In *ICLR*, 2024.

Ruiyao Xu, Noelle I. Samia, and Han Liu. DS2-instruct: Domain-specific data synthesis for large language models instruction tuning. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3368–3384, Rabat, Morocco, March 2026a. Association for Computational Linguistics. ISBN 979-8-89176-386-9. doi: 10.18653/v1/2026.findings-eacl.176. URL https://aclanthology.org/2026.findings-eacl.176/.

Xiaoliang Xu, Huang Yuan, Junmei Wang, and Can Xu. GCIG: GraphRAG-based cross-document instruction generation for boosting LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17593–17609. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.findings-acl.871. URL https://aclanthology.org/2026.findings-acl.871/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. Genie: Achieving human parity in content-grounded datasets generation. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/a774503daed55eb53c634847ae071ec7-Paper-Conference.pdf.

Fan Yin, Zifeng Wang, I-Hung Hsu, Jun Yan, Ke Jiang, Yanfei Chen, Jindong Gu, Long T. Le, Kai-Wei Chang, Chen-Yu Lee, Hamid Palangi, and Tomas Pfister. Magnet: Multi-turn tool-use data synthesis and distillation via graph translation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32600–32616. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1566. URL https://aclanthology.org/2025.acl-long.1566/.

Ping Yu, Jack Lanchantin, Tianlu Wang, Weizhe Yuan, Olga Golovneva, Ilia Kulikov, Sainbayar Sukhbaatar, Jason Weston, and Jing Xu. Cot-self-instruct: Building high-quality synthetic prompts for reasoning and non-reasoning tasks. *arXiv preprint arXiv:2507.23751*, 2025.

Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhu Chen. Mammoth2: Scaling instructions from the web. In *NeurIPS*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *ACL*, 2019.

Weihao Zeng, Can Xu, Yingxiu Zhao, Jian-Guang Lou, and Weizhu Chen. Automatic instruction evolving for large language models. In *EMNLP*, 2024.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatGPT interaction logs in the wild. In *ICLR*, 2024.

Chiwei Zhu, Benfeng Xu, Xiaorui Wang, and Zhendong Mao. From real to synthetic: Synthesizing millions of diversified and complicated user instructions with attributed grounding. In *ACL*, 2025.

Ingo Ziegler, Abdullatif Köksal, Desmond Elliott, and Hinrich Schütze. CRAFT your dataset: Task-specific synthetic dataset generation through corpus retrieval and augmentation. In *TACL*, 2025.

Appendix Overview

- Appendix A: HSS Domain Document Corpus
- Appendix B: Algorithm
- Appendix C: Datasets and Implementation Details
- Appendix D: Prompts
- Appendix E: Statistical Analysis
- Appendix F: More Experiments
- Appendix G: Case Study
- Appendix H: Limitations
- Appendix I: Broader Impact

A HSS Domain Document Corpus

The HSS domain corpus consists of 79 publicly available books spanning seven sub-disciplines: history, political science, economics, psychology, philosophy, sociology, and STEM. Books were selected to cover a broad range of topics within each sub-discipline, from canonical texts (e.g., Kant's *Critique of Pure Reason*, Weber's *The Protestant Ethic and the Spirit of Capitalism*) to modern introductions and reference works. Each book was segmented into document chunks of approximately 500–1,000 tokens, yielding 38,572 target documents for logic graph extraction and downstream synthesis.

HSS Domain Book List

A Black Jurist in a Slave Society, A Confession, A History of Modern Africa 1800 to the Present, A History of Science (parts 1–4), A People's History of the United States, A Study of the Bible, African History: A Very Short Introduction, Algorithms: 3 Books in 1, American Government 101, Bayesian Methods in Finance, Can Science Explain Religion, Collective Action and Exchange, Computer Architecture, Computer Architecture: A Quantitative Approach, Constitutional Law: Principles and Policies, Evolutionary Psychology: The New Science of the Mind, Factfulness: Discussion Prompts, Feeding the World, Feminist Teaching in Theory and Practice, Fiasco: The American Military Adventure in Iraq, Global Sales and Contract Law, History of China: A Captivating Guide, History of Eastern Europe: A Captivating Guide, Humanity, In Command of History, Influence: The Psychology of Persuasion, Institutions and Economic Development, Introduction to Data Structures and Algorithms for Beginners, Introduction to Econometrics, Justice for All, Key Concepts in Historical Geography, Leading from the Front Line, Lean In, Learn Cognitive Psychology, Leviathan and the Air-Pump, Liberty, Linear Algebra Done Right, Logic: A Very Short Introduction, Macroeconomics, Making Our Democracy Work, Manias Panics and Crashes, Marshall Marshallians and Industrial Economics, Mental Models, Modern Historiography, Modern History: A Captivating Guide, Mostly Harmless Econometrics, National Insecurity, Oil Palm: A Global History, Personality and Intelligence at Work, Political Philosophy: A Complete Introduction, Presidential Lessons in Leadership, Seeing Like a State, Southeast Asia: Political Economy of Structural Change, Sustainable Growth and Resource Productivity, The Basic Political Writings, The Blank Slate, The Cambridge Companion to Utopian Literature, The Courage to Create, The Critique of Pure Reason, The Joy of Game Theory, The Laws of Human Nature, The Lucifer Effect, The Making of the Middle Sea, The Mystery of Capital, The Nature of Rent, The Political Economy of Expertise, The Protestant Ethic and the Spirit of Capitalism, The Researcher's Bible, The Slave Ship: A Human History, The United Nations and Changing World Politics, The Wiley Encyclopedia of Personality and Individual Differences, West from Appomattox, What is Property, Why Nations Fail, Why We Disagree About Climate Change, Wind Energy: Renewable Energy and the Environment, World of Patterns: A Global History of Knowledge, and Wrongful Convictions and Miscarriages of Justice.

B Algorithm

Algorithm 1 describes the one-time transfer graph library construction, and Algorithm 2 describes the synthesis pipeline applied to new target documents.

C Datasets and Implementation Details

C.1 Implementation Details

Medical source documents and test separation. The PubMed abstracts used to construct DataSTAMP-Med are disjoint from the documents in the PubMedQA test split used for evaluation. This exclusion is at the source-document level, rather than only at the level of existing QA pairs.

Library construction. We construct the transfer graph library from DLR-Web, the cross-disciplinary QA corpus released by the DESIGNER pipeline (Liu et al., 2026), which spans 75 disciplines including mathematics, computer science, biology, physics, philosophy, law, and clinical medicine.

Algorithm 1 Transfer graph Library Construction (Meta-Training)

Require: Cross-domain QA collection $\mathcal{P} = \{(d_j^s, q_j, a_j)\}_{j=1}^{K}$; LLM $\mathcal{M}$; deduplication threshold τ
Ensure: Deduplicated transfer graph library $\mathcal{L}^*$

for each $(d_j^s, q_j, a_j) \in \mathcal{P}$ **do**
  Extract logic graph $\mathcal{G}_j^{\mathrm{L}}$ from d_j^s using $\mathcal{M}$ with type vocabularies $\mathcal{T}_V, \mathcal{T}_E$ ▷ Def. 1
  Extract transfer graph $\mathcal{G}_j^{\mathrm{X}}$ from (d_j^s, q_j, a_j) using $\mathcal{M}$, stripping domain surface form ▷ Def. 2
  Store $(d_j^s, \mathcal{G}_j^{\mathrm{L}}, \mathcal{G}_j^{\mathrm{X}}, q_j, a_j)$ in library $\mathcal{L}$ ▷ Eq. 2
end for
Embed each $\mathcal{G}_j^{\mathrm{X}}$ with a sentence encoder
Build similarity graph with edges for pairs above threshold τ
Find connected components via union-find
For each component, retain the element maximizing intra-cluster similarity
return Deduplicated library $\mathcal{L}^*$

Algorithm 2 DATASTAMP Synthesis Pipeline (Meta-Test)

Require: Target documents $\mathcal{D} = \{d_1, \ldots, d_M\}$; transfer graph library $\mathcal{L}^*$; R-GCN encoder ϕ; number of transfer graphs k; LLM $\mathcal{M}$; verifier $\mathcal{V}$; sentence encoder ψ
Ensure: Synthesized dataset $\mathcal{S}$

$\mathcal{S} \leftarrow \emptyset$
Pre-compute library embeddings: $\mathbf{E}_{\mathcal{L}} \leftarrow [\phi(\mathcal{G}_j^{\mathrm{L}})]_{j \in \mathcal{L}^*}$ ▷ Eq. 3
for each $d_t \in \mathcal{D}$ **do**
  Extract logic graph $\mathcal{G}_t^{\mathrm{L}}$ from d_t using $\mathcal{M}$
  Compute embedding $\mathbf{e}_t \leftarrow \phi(\mathcal{G}_t^{\mathrm{L}})$
  Retrieve top-k entries: $\mathcal{M}_t \leftarrow \text{top-k}(\mathbf{e}_t^{\top} \mathbf{E}_{\mathcal{L}})$ ▷ Eq. 4
  $\mathcal{C}_t \leftarrow \emptyset$ ▷ Candidate pool for d_t
  for each $(\mathcal{G}_j^{\mathrm{X}}, d_j^s, q_j, a_j) \in \mathcal{M}_t$ **do**
    Construct node correspondence f_j via greedy type alignment
    Project: $(\hat{x}_{t,j}, \hat{y}_{t,j}) \leftarrow \Pi(d_t, \mathcal{G}_j^{\mathrm{X}}, f_j, (q_j, a_j))$ using $\mathcal{M}$ ▷ Eq. 5
    if $\mathcal{V}(\hat{x}_{t,j}, \hat{y}_{t,j}, d_t) = 1$ **then**
      $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \{(\hat{x}_{t,j}, \hat{y}_{t,j})\}$
    end if
  end for
  $\mathcal{S} \leftarrow \mathcal{S} \cup \text{div-select}(\mathcal{C}_t, \psi)$ ▷ Eq. 6
end for
return $\mathcal{S}$

Because the raw corpus is highly skewed across disciplines, we first draw a stratified 200,000-entry subsample (seed 42): each discipline receives $\max(50, \lfloor 200{,}000 \cdot p_d \rfloor)$ entries, where p_d is the discipline's share of the full corpus, and within each discipline we further stratify by difficulty so that all four levels (Easy, Medium, Hard, Very Hard) are represented in proportion to their natural frequency. Logic graph and transfer graph extraction (prompts P1 and P2 in App. D.1) is performed by Qwen3-Next-80B-A3B-Instruct with sampling temperature 0.3 and a 2,048-token output cap. The six node types $\mathcal{T}_V$ and seven edge types $\mathcal{T}_E$ defined in §4 are specified in the extraction prompts, and the model's adherence to these closed type vocabularies is empirically near-total (99.93% canonical node tokens, 99.95% canonical edge tokens; see App. E.1). The raw extraction yields 199,975 entries, which are deduplicated by encoding each transfer graph Mermaid string with `all-roberta-large-v1`, building an undirected $\tau \geq 0.85$ similarity graph, and retaining one representative per connected component. The deduplicated library contains 189,664 entries.

R-GCN encoder. The graph encoder is a 5-layer R-GCN with hidden dimension $d_h = 32$, output dimension $d_o = 128$, and ReLU activations. Graph-level representations are obtained via mean pooling followed by a linear projection and L_2 normalization. The encoder embeds the full 189,664-entry library in approximately 18 seconds on a single CPU core.

SFT training. All SFT experiments use identical hyperparameters for a controlled comparison. We fine-tune Qwen3-8B-Base (Yang et al., 2025) in full-parameter mode with AdamW, learning rate 1×10^{-5} with cosine decay and 5% warmup ratio, and 3 training epochs. The effective batch size is 64. Training is conducted on $4\times$ NVIDIA A100 (80 GB) GPUs with DeepSpeed ZeRO Stage 3.

Evaluation protocol. All twelve benchmarks are evaluated with the protocol native to each: zero-shot for HLE, SuperGPQA, BBH, DROP, HellaSwag, and HealthBench; 5-shot for MMLU-Pro; one-shot for PubMedQA; and 4-shot for MedQA. System prompts are reproduced in App. D.2. For free-form benchmarks (DROP, HLE, HealthBench) we use GPT-4o (Hurst et al., 2024) as the LLM judge; for multiple-choice and yes/no/maybe benchmarks we extract the chosen label with deterministic regex matching and fall back to an LLM judge call only when extraction fails.

C.2 Datasets

Table 4: Statistics of instruction datasets used in our experiments.

Domain	Dataset	Tokens (M)	# Pairs
	Humanities & Social Sciences (HSS)		
	DLR-HSS (Liu et al., 2026)	56.1	14K
	Condor (Cao et al., 2025)	8.6	20K
	LongWriter (Bai et al., 2025)	10.9	6K
	OpenHermes (Teknium, 2023)	236.5	1.0M
	WebRPro (Jiang et al., 2025)	99.6	100K
	WildChat (Zhao et al., 2024)	153.3	529K
	DataStamp-HSS	**9.7**	**13.9K**
	Medical & Health		
	DLR-Med (Liu et al., 2026)	251.8	67K
	ChatDoctor (Li et al., 2023)	246.3	100K
	MedQA-SFT (Jin et al., 2021)	15.7	12.7K
	Medical-O1 (Chen et al., 2024)	557.9	19.7K
	Medical-R1-Distill	579.4	22K
	ReasonMed (Sun et al., 2025)	645.4	370K
	DataStamp-Med	**50.6**	**67.1K**

Medical reasoning dataset provenance. Medical-O1 refers to the GPT-4o-generated `medical-o1-reasoning-SFT` dataset associated with HuatuoGPT-o1 (Chen et al., 2024). Medical-R1-Distill instead refers to FreedomIntelligence's separate DeepSeek-R1-distilled release (FreedomIntelligence, 2025), based on the verifiable medical problems from HuatuoGPT-o1. These are distinct instruction datasets, not two names for the same release.

C.3 Retrieval Complexity and Cost Accounting

Offline encoding versus library construction. Appendix C reports approximately 18 seconds to encode the 189,664 library entries with the R-GCN on one CPU core. This is an embedding computation, not the total cost of library construction: the preceding LLM-based extraction and deduplication must be accounted for separately. With a fixed encoder, existing embeddings can be cached and only new entries need to be encoded.

Online retrieval. For N cached library vectors of dimension d, exhaustive similarity scoring requires Nd multiply-accumulate operations (MACs), followed by top-k selection. At $N = 189{,}664$ and $d = 128$, this is approximately 24.3 million MACs, or 48.6 million FLOPs when a multiplication and an addition are counted separately. The float32 embedding matrix occupies approximately 97.1 MB, excluding graph storage and other overheads. These quantities describe cached-vector scoring and exclude target-graph extraction, target encoding, and QA generation or verification.

Scaling estimates. At $N = 10^7$, cached-vector scoring requires approximately 1.28 billion MACs and 5.12 GB of float32 embedding storage. A linear extrapolation of the reported encoding time

gives roughly 16 minutes, assuming unchanged throughput. This is an analytical extrapolation, not a measured large-library runtime.

D PROMPTS

This appendix collects the full text of every LLM prompt used in our pipeline and evaluation suite. Prompts are reproduced verbatim. Placeholders in braces such as `{document}` or `{question}` are runtime substitutions performed in code.

D.1 DATA SYNTHESIS PIPELINE PROMPTS

The four prompts in this subsection drive the dual-logic decomposition, library construction, cross-domain projection, and grounding verification stages described in §4.

P1. Logic graph Extraction

You are an expert at analyzing the structural organization of knowledge in documents. Given a document segment, your task is to extract its **knowledge structure** as an abstract graph in Mermaid format.

Critical Rule: ZERO Domain Content. You MUST use ONLY numbered abstract labels. Node labels must be EXACTLY in the format `Type_N` (e.g., `Entity_1`, `Process_2`). Do NOT add any descriptive text, domain words, or qualifiers after the number.

WRONG: `Entity_1: Election Outcome`, `Process_2: centripetal acceleration`, `Outcome_1: sustained trajectory`
RIGHT: `Entity_1`, `Process_2`, `Outcome_1`

The ONLY place where brief abstract qualifiers are allowed is on edges, using ONLY these relationship types — no domain words.

Node Types (use ONLY these, numbered):
`Entity_N` (a concrete or abstract thing), `Process_N` (an action, mechanism, or transformation), `Property_N` (an attribute), `Outcome_N` (a result), `Condition_N` (a prerequisite or threshold), `Quantity_N` (a measurable value).

Edge Types (use ONLY these exact labels):
`causal`, `comparative`, `hierarchical`, `temporal`, `conditional`, `inhibitory`, `compositional`.

Document Segment: `{document}`

Output: Return ONLY a Mermaid graph (" ' `mermaid ...` " ') capturing the knowledge structure. Keep it concise (typically 3–10 nodes). Focus on the dominant structural pattern. Use ONLY numbered abstract labels — any domain-specific word in a node label is a failure.

P2. Transfer graph Extraction

You are an expert at analyzing how document knowledge is transformed into exam questions. Given a document segment and a question–answer pair generated from it, your task is to extract the **conversion logic** — the step-by-step transformation process from document to QA — as an abstract graph in Mermaid format.

Critical Rule: ZERO Domain Content. Every node label must describe a generic structural operation — NO domain-specific words whatsoever. If someone reads only your graph, they must have NO idea what field (law, physics, biology, etc.) the original document is from.

WRONG: `Select: legal authority clause`, `Extract: protonation states`, `Contrast: state vs. federal authority`
RIGHT: `Select: two conflicting authority sources`, `Extract: parameter values at boundary`, `Contrast: local vs. global constraints`

What to Capture. Represent the document-to-QA transformation as a flowchart covering:
(1) **Selection**: which structural components were selected?
(2) **Reorganization**: how were components restructured?
(3) **Reasoning Demand**: what cognitive operation is required?
(4) **Derivation Path**: how to get from document to answer?

Document Segment: `{document}`
Question: `{question}`
Answer: `{answer}`

Output: Return ONLY a Mermaid graph capturing the conversion logic. Every node must be domain-free — any domain-specific word is a failure. Focus on the reusable structural pattern.

P3. Cross-Domain Projection

You are an expert at creating domain-specific questions from documents in the humanities and social sciences (e.g., history, political science, philosophy, economics, sociology, psychology, law, geography, cultural studies). You are given:
(1) a **target document segment** containing domain knowledge,
(2) a **conversion logic** (in Mermaid format) describing how to transform document knowledge into a question,
(3) a **node mapping** showing which parts of the target document correspond to which structural roles, and
(4) an **exemplar QA pair** from a potentially different domain, showing how this conversion logic was previously applied. Use the exemplar only to understand the *reasoning structure* — do NOT imitate its domain, terminology, or style.

Task. Apply the conversion logic to the target document to generate a new question–answer pair grounded entirely in the target document's domain.

Rules. Follow the conversion logic steps strictly; ground ALL content in the target document — do not hallucinate facts; the question must be self-contained and answerable without the source document; the answer must trace the reasoning path specified by the conversion logic; frame questions and answers using the analytical vocabulary of the target domain (argue, critique, compare perspectives, evaluate evidence, analyze causes/consequences, interpret significance).

Target Document Segment: `{document}`
Conversion Logic: `{conversion_logic}`
Node Mapping: `{node_mapping}`
Exemplar Question: `{exemplar_question}`
Exemplar Answer: `{exemplar_answer}`

Output (JSON only):
`{ "question": "<your generated question>", "answer": "<your generated answer>" }`

P4. Grounding Verification

You are a quality checker for synthesized QA pairs. Verify whether the question and answer are factually correct based on the source document.

Source Document: `{document}`
Generated Question: `{question}`
Generated Answer: `{answer}`

Evaluate.
(1) **Question Valid**: Is the question answerable from the document? Does it ask about information that is actually present?
(2) **Answer Correct**: Is the answer factually accurate according to the document? Are all claims, numbers, and conclusions supported?

Output (JSON):
`{ "question_valid": {"pass": true/false, "reason": "..."}, "answer_correct": {"pass": true/false, "reason": "..."} }`

D.2 Evaluation Prompts

System prompts and input templates used to elicit answers from the model under evaluation. Each benchmark uses a distinct prompt tailored to its answer format. Placeholders in braces are runtime substitutions; few-shot exemplars are separated by dashes.

Evaluation Prompts

E1. HLE Bio/Med (zero-shot, free-form)
System: You are a biomedical expert. Answer the following question as accurately and thoroughly as possible.
User: `{question}`

E2. HLE HSS (zero-shot, free-form)
System: You are a humanities and social sciences expert. Answer the following question as accurately and thoroughly as possible.
User: `{question}`

E3. PubMedQA (1-shot, yes/no/maybe)
System: You are a biomedical expert. Answer the following question based on the context. Think through the evidence step by step in a few sentences, then provide your final answer on a new line in the format: `Answer: yes/no/maybe`
User: `{1-shot exemplar}` \n---\n Context: `{context}` \n Question: `{question}`

E4. MedQA (4-shot, 4 options A–D)
System: You are a medical expert taking the USMLE exam. Think through the problem step by step in a few sentences, then provide your final answer on a new line in the format: `Answer: X` where X is the letter of the correct option (A, B, C, or D).
User: `{4-shot exemplars}` \n---\n Question: `{question}` \n `{options}`

E5. MMLU-Pro Health (5-shot, up to 10 options A–J)
System: You are a medical expert. Answer the following multiple-choice question about health. Think through the problem step by step in a few sentences, then provide your final answer on a new line in the format: `Answer: X` where X is the letter of the correct option.
User: `{5-shot exemplars}` \n---\n Question: `{question}` \n `{options}`

E6. MMLU-Pro History (5-shot, up to 10 options A–J)
System: You are a humanities and social sciences expert. Answer the following multiple-choice question about history. Think through the problem step by step in a few sentences, then provide your final answer on a new line in the format: `Answer: X` where X is the letter of the correct option.
User: `{5-shot exemplars}` \n---\n Question: `{question}` \n `{options}`

E7. SuperGPQA Health (zero-shot, multiple-choice)
System: You are a medical expert. Answer the following multiple-choice question. Think through the problem step by step in a few sentences, then provide your final answer on a new line in the format: `Answer: X` where X is the letter of the correct option.
User: Question: `{question}` \n `{options}`

E8. SuperGPQA History (zero-shot, multiple-choice)
System: You are a humanities and social sciences expert. Answer the following multiple-choice question. Think through the problem step by step in a few sentences, then provide your final answer on a new line in the format: `Answer: X` where X is the letter of the correct option.
User: Question: `{question}` \n `{options}`

E9. BBH (zero-shot, free-form)
System: You are an expert problem solver. Answer the following question step by step. After your reasoning, provide your final answer on a new line in the format: `Answer: <your answer>`
User: `{question}`

E10. DROP (zero-shot, extractive span)
System: You are an expert at reading comprehension and discrete reasoning. Read the passage and answer the question. Your answer should be a short, precise span (a number, name, date, or phrase) extracted from or derived from the passage. After your reasoning, provide your final answer on a new line in the format: `Answer: <your answer>`
User: Passage: \n `{passage}` \n\n Question: `{question}`

E11. HellaSwag (zero-shot, 4 continuations A–D)
System: You are given a context and four possible continuations. Choose the most plausible continuation. Think through each option briefly, then provide your final answer on a new line in the format: `Answer: X` where X is the letter of the correct option (A, B, C, or D).
User: Context: `{context}` \n `{options}`

E12. HealthBench (zero-shot, conversational; GPT-4o rubric judge)
System: You are a helpful and knowledgeable medical assistant.
User: `{conversation_turns}`

D.3 Difficulty Classification Prompt

Following DESIGNER (Liu et al., 2026), the difficulty analysis in Figure 2a uses the few-shot prompt below. It assigns one of four labels based on the complexity and length of the reasoning needed to answer a question. The three examples and their labels are reproduced from DESIGNER; `{text}` is replaced with the question being classified.

Difficulty Classification

You are an expert in education and examination, specializing in classifying the difficulty levels of multidisciplinary questions. For the given question, please evaluate its difficulty based on the complexity and length of the reasoning required to answer it. Label it as one of the following: **Easy**, **Medium**, **Hard**, or **Very Hard**. Please directly output "Difficulty: (Your chosen label)".

 # Example 1
Input: "Consider a photon traveling at the speed of light. How does the photon experience space, and what are the implications of relativistic beaming on its perception of spatial dimensions? Provide a detailed explanation, including any relevant mathematical derivations and physical principles."
Output: "Difficulty: Very Hard"

 # Example 2
Input: "A heavy pole, of mass M and length L, is freely hinged to a wall at the point O. A rope connects the other end of the pole, B, to a fixed point A on the wall above O. The system is in equilibrium, with the pole making an angle of θ with the horizontal, and the rope making an angle of α with the horizontal. Explore how the system's parameters (M, L, θ, α) affect its equilibrium and stability."
Output: "Difficulty: Hard"

 # Example 3
Input: "If John rented a car for $150 and had to buy 8 gallons of gas at $3.50 per gallon to fill it up, and the final expense is $0.50 per mile, how much did it cost him to drive 320 miles?"
Output: "Difficulty: Easy"

 # Given Question
Input: "`{text}`"
Output:

E STATISTICAL ANALYSIS

This appendix provides additional statistical evidence supporting the claims in the main text.

E.1 LIBRARY TOPOLOGY

We characterize the structural distribution of the 189,664-entry transfer graph library along three axes: (i) per-graph size, (ii) node- and edge-type composition, and (iii) topology-signature concentration.

Graph sizes. Both the logic graph $\mathcal{G}^{\mathrm{L}}$ and the transfer graph $\mathcal{G}^{\mathrm{X}}$ are concise: median 10 nodes / 9 edges and 9 nodes / 8 edges respectively. The distributions are tightly concentrated; a long thin tail (max 266 nodes for $\mathcal{G}^{\mathrm{L}}$) is dominated by a small number of very long extracted documents.

Table 5: Library size summary statistics. Each row is a per-graph quantity (nodes or edges) summarized over all 189,664 entries.

	mean	std	p25	median	p75	p95	max
$\mathcal{G}^{\mathrm{L}}$ nodes	10.27	5.39	8	10	12	15	266
$\mathcal{G}^{\mathrm{L}}$ edges	9.36	3.56	8	9	10	12	153
$\mathcal{G}^{\mathrm{X}}$ nodes	9.99	3.66	7	9	11	18	48
$\mathcal{G}^{\mathrm{X}}$ edges	8.93	3.49	6	8	10	16	41

Node and edge type composition. The extraction prompt instructs the LLM to use exactly six node types (ENTITY, PROCESS, PROPERTY, OUTCOME, CONDITION, QUANTITY) and seven edge types (CAUSAL, COMPARATIVE, HIERARCHICAL, TEMPORAL, CONDITIONAL, INHIBITORY, COMPOSITIONAL). Empirically, adherence is near-total: 99.93% of node tokens and 99.95% of edge tokens are canonical (Tables 6 and 7).

Table 6: Node-type histogram across the full library (1.95M node tokens).

Type	Count	% of tokens
OUTCOME	644,221	33.07
PROCESS	623,893	32.03
ENTITY	285,968	14.68
CONDITION	167,806	8.61
QUANTITY	134,629	6.91
PROPERTY	90,024	4.62
Other (373 LLM-introduced types)	1,455	0.075

Table 7: Edge-type histogram across the full library (1.78M edge tokens).

Type	Count	% of tokens
CAUSAL	766,832	43.17
CONDITIONAL	345,738	19.47
COMPARATIVE	253,115	14.25
INHIBITORY	144,239	8.12
TEMPORAL	112,623	6.34
HIERARCHICAL	98,043	5.52
COMPOSITIONAL	54,760	3.08
Other (107 LLM-introduced types)	831	0.047

Topology diversity. To assess whether the library collapses to a small number of dominant topologies, we measure concentration two ways. First, we compute the multiset signature of canonical node types per graph (e.g., $\{\text{ENTITY}, \text{PROCESS}, \text{PROCESS}, \text{OUTCOME}\}$); the library contains $6{,}683$ distinct signatures, with the top-20 covering only 33.81% of entries and the single most frequent

signature accounting for 5.50%. Second, we apply Weisfeiler-Lehman color refinement to collapse isomorphism classes of typed directed graphs, yielding 117,313 distinct WL classes (uniqueness ratio 0.619), with the top-20 covering only 3.66%. By either measure, the library is highly diverse.

E.2 Source-Discipline Composition and Cross-Domain Transfer

A central question is whether the medical and HSS gains reported in §5 come from in-domain matches (i.e., the library entry was itself a medical/HSS exemplar) or from genuinely cross-domain structural transfer.

Library composition. The 189,664 library entries span 75 source disciplines (entropy 3.32 nats). The largest sources are general-purpose technical fields: Computer Science (25,495), Mathematics (18,904), Philosophy (14,788), Biology (12,887), Physics (11,502), Psychology (11,158), and Clinical Medicine (8,142). Medical disciplines collectively total ∼5.7% of the library; HSS disciplines total ∼30%.

Source-target cross-tabulation. Among the 75,015 PubMedQA targets that received a match, only 7.69% of the retrieved transfer graphs originate from medical disciplines; 92.31% come from non-medical sources, led by Computer Science (10.0%), Mathematics (9.2%), and Biology (8.6%). For Books (HSS) targets, 54.19% of matches come from non-HSS sources. Table 8 details the full cross-tabulation. This confirms that the gains attributed to DataStamp arise from genuinely cross-domain transfer of reasoning structure rather than in-domain template re-use.

Table 8: Source-discipline composition of the library and percentage of retrieved transfer graphs per target domain. Top 12 of 75 disciplines, ranked by total retrieval volume.

Source discipline	Library count	DataStamp-Med %	DataStamp-HSS %
Computer Science and Technology	25,495	10.01	9.28
Mathematics	18,904	9.16	7.99
Biology	12,887	8.63	5.32
Philosophy	14,788	6.06	11.81
Law	7,374	5.87	10.43
Physics	11,502	7.14	3.63
Clinical Medicine	8,142	4.85	3.32
Political Science	6,166	3.71	5.19
Psychology	11,158	3.87	4.35
Chemistry	6,775	3.88	2.19
Mechanical Engineering	6,093	3.64	2.25
Business Administration	5,287	3.05	2.02
In-domain total	n/a	**7.69**	**45.81**
Cross-domain total	n/a	**92.31**	**54.19**

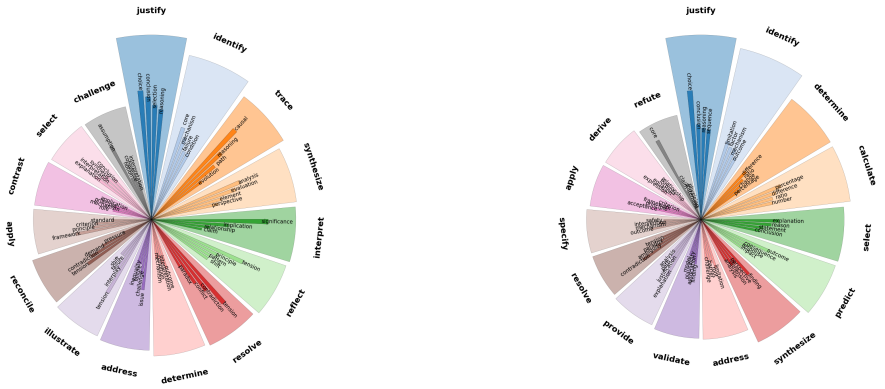
E.3 Lexical Diversity of Synthesized Instructions

To complement the structural-diversity analysis above, we visualize the surface-level diversity of DataStamp-synthesized instructions in both target domains. Figure 4 shows word clouds over the instruction text for DataStamp-HSS and DataStamp-Med after removing stop words and benchmark-specific boilerplate; the clouds span a broad vocabulary spread across many topical clusters rather than concentrating on a few high-frequency tokens. Figure 5 reports the top verb–noun pairs extracted from the instructions via spaCy dependency parsing, following the protocol of Self-Instruct (Wang et al., 2023). The diversity of leading verb–noun pairs in both domains indicates that the synthesized instructions exercise heterogeneous reasoning patterns rather than a small set of templated forms.



(a) DATASTAMP-HSS (b) DATASTAMP-Med

Figure 4: Word clouds over instructions in the two synthesized datasets.



(a) DATASTAMP-HSS (b) DATASTAMP-Med

Figure 5: Top verb–noun pairs in the synthesized instructions.

F MORE EXPERIMENTS

This section brings together experiments on model architecture generalization, structure selection, graph faithfulness, library size, and transfer to agronomy. The expanded ablation is reported separately from the original component ablation in Table 3; it does not replace that table. Performance results, reported averages, and differences are independently rounded to one decimal place for display; averages and differences are not recomputed from the rounded table entries.

F.1 MODEL ARCHITECTURE GENERALIZATION

To examine performance with another model family, we fine-tune LLaMA-3-8B-Base (Grattafiori et al., 2024) on the same datasets generated for Qwen3-8B. Table 9 reports the results. In the medical domain, LLaMA-3-8B + DATASTAMP improves average accuracy from 29.6 to 43.8 (+14.2), with PubMedQA showing the largest single gain (+47.0). In HSS, the average rises from 28.4 to 33.2 (+4.8), with gains on HLE (+5.9), BBH (+6.5), and HellaSwag (+7.8). LLaMA-3-8B + DATASTAMP remains below Qwen3-8B + DATASTAMP on the domain averages, but exceeds it on PubMedQA (78.1 vs. 76.6). The same synthesized datasets therefore improve both tested model families, although these results do not isolate the reason for their different performance levels.

Table 9: Cross-architecture generalization. All metrics are accuracy (%, higher is better). **Avg**: mean accuracy across six benchmarks.

	MMLU-Pro	SuperGPQA	HLE	BBH	HellaSwag	DROP	Avg
Humanities & Social Sciences (HSS)							
LLaMA-3-8B-Base	21.5	23.7	5.5	34.6	39.5	45.8	28.4
LLaMA-3-8B + DATASTAMP	24.2	26.8	11.4	41.1	47.3	48.7	33.2
	MMLU-Pro	**SuperGPQA**	**HLE**	**HealthBench**	**PubMedQA**	**MedQA**	**Avg**
Medical & Health							
LLaMA-3-8B-Base	16.5	28.7	9.3	54.9	31.1	36.8	29.6
LLaMA-3-8B + DATASTAMP	34.2	29.5	18.6	63.0	78.1	39.3	43.8

F.2 EXPANDED STRUCTURE-SELECTION AND PROJECTION ABLATIONS

We compare eight synthesis conditions to distinguish the choice of retrieved structure from the information supplied to the projection model. Table 10 defines the conditions. The study uses common experimental settings as in Appendix C. For each condition, downstream scores are averaged over five evaluations; comparisons of means alone do not establish statistical significance. Within the library-based graph-plus-exemplar conditions, the generation budget and synthesized pair count are held fixed.

Table 10: Conditions in the expanded ablation. Structural retrieval uses R-GCN representations; semantic retrieval uses text embeddings.

ID	Condition	Selection	Projection input
R1	Base model	Not applicable	No SFT
C1	Random + raw QA	Uniform library sampling	Exemplar QA only
C2	Semantic + raw QA	Text-embedding retrieval	Exemplar QA only
C3	Structural + raw QA	R-GCN retrieval	Exemplar QA only
C4	Random + graph	Uniform library sampling	Transfer graph + exemplar
C5	Semantic + graph	Text-embedding retrieval	Transfer graph + exemplar
C6	Adversarial + graph	Lowest structural similarity	Transfer graph + exemplar
C7	On-the-fly graph	LLM-generated; no library	Self-generated graph
C8	Full DATASTAMP	R-GCN retrieval	Transfer graph + exemplar

Comparisons enabled by this design. C1–C3 vary selection while supplying only a raw QA exemplar. C4, C5, C6, and C8 vary selection while retaining a graph-plus-exemplar prompt. C1 versus C4, C2 versus C5, and C3 versus C8 compare the addition of an explicit graph under the same selection strategy. C7 tests synthesis without the pre-built library, but also omits the retrieved exemplar; therefore, C7 versus C8 is a comparison of complete configurations rather than an isolated intervention on library access.

Relationship to the original ablation. The *Random QA exemplars* baseline in Table 3 uses random selection and raw QA inputs, corresponding to C1 rather than the structurally retrieved QA exemplars in C3. Its comparison with the full model changes both selection and graph input, whereas C1 versus C4 and C3 versus C8 hold selection fixed. The original table also includes removal of verification, which is not varied in C1–C8. Results from the original and expanded studies are reported separately and are not pooled. The expanded study covers all six benchmarks in each of the two main domains.

F.2.1 RESULTS ACROSS THE TWO MAIN DOMAINS

Selection and explicit structure. Among graph-plus-exemplar conditions, full structural retrieval exceeds random selection by 5.2 percentage points on HSS and 3.7 on Medical. Adding a graph to structural retrieval (C3 versus C8) improves the averages by 4.3 and 3.1 points, respectively. Adding a graph to random selection (C1 versus C4) yields gains of 3.4 and 2.2 points. These comparisons support contributions from both selection and explicit graph guidance.

Table 11: Expanded ablation on HSS. All values are percentages; Avg is the reported six-benchmark mean. C8 is best in every column.

ID	MMLU-Pro	SuperGPQA	HLE	BBH	HellaSwag	DROP	Avg
Humanities & Social Sciences (HSS)							
R1	32.7	29.2	9.1	43.5	44.3	60.9	36.6
C1	44.8	34.3	12.4	57.7	58.1	63.0	45.1
C2	45.3	36.4	11.8	60.9	63.3	61.7	46.6
C3	48.9	35.6	14.3	63.2	68.6	65.7	49.4
C4	47.6	36.1	13.0	61.8	67.3	64.9	48.5
C5	46.1	34.8	10.9	60.9	63.1	66.1	47.0
C6	40.2	33.5	10.4	56.7	65.0	62.9	44.8
C7	43.2	36.2	10.3	62.3	62.5	64.0	46.4
C8	**52.4**	**40.3**	**15.1**	**67.6**	**77.8**	**69.0**	**53.7**

Table 12: Expanded ablation on Medical & Health. All values are percentages, displayed to one decimal place; Avg is the reported six-benchmark mean. C8 is best in every column.

ID	MMLU-Pro	SuperGPQA	HLE	HealthBench	PubMedQA	MedQA	Avg
Medical & Health							
R1	51.0	26.1	13.6	63.0	56.4	57.1	44.5
C1	57.0	31.9	17.5	68.7	64.1	60.3	49.9
C2	57.9	35.8	16.3	69.9	67.0	58.8	51.0
C3	59.6	34.6	18.9	70.4	71.2	61.5	52.7
C4	58.9	35.4	17.3	70.0	70.4	60.5	52.1
C5	58.2	33.4	15.1	69.5	68.0	62.0	51.0
C6	55.0	32.0	15.8	68.1	69.8	58.8	49.9
C7	56.6	37.4	15.9	70.3	67.4	60.2	51.3
C8	**61.1**	**41.4**	**19.6**	**71.7**	**76.6**	**64.4**	**55.8**

Mismatched and on-the-fly structures. Adversarial selection gives averages close to random raw-QA prompting (44.8 versus 45.1 on HSS; 49.9 versus 49.9 on Medical), indicating that a graph's presence alone is insufficient. On-the-fly generation trails full DATASTAMP by 7.3 and 4.5 points, and random library graphs by 2.0 and 0.8 points. These are observed differences under the reported configurations, without multi-seed uncertainty estimates.

F.3 FAITHFULNESS OF EXTRACTED GRAPHS AND PROJECTED PAIRS

Adherence to the closed node and edge vocabularies does not establish semantic faithfulness. An additional evaluation checks three distinct properties: whether a logic graph faithfully represents its source document, whether a transfer graph faithfully represents its source transformation, and whether a projected QA pair both instantiates the supplied graph and remains grounded in the target document.

Table 13: Faithfulness pass rates (%). Automated judges evaluate the corresponding evaluation collection; human evaluation uses 100 samples and two annotators. The human column reports aggregate pass rates, not inter-annotator agreement.

Object evaluated	Gemini 3.5 Flash	Claude Opus 4.8	Human (100)
Logic graph faithful to its document	94.2	89.2	94.0
Transfer graph faithful to its source transformation	91.1	92.9	89.0
Projected QA instantiates its graph and is grounded	92.6	91.6	93.0

Interpretation and scope. The three checks address different failure modes, and the non-perfect pass rates indicate residual errors at each stage. The human results provide a small-sample check

alongside the automated assessments. Similar aggregate pass rates do not establish per-example agreement between judges or with human annotations; the table should not be interpreted as an inter-rater reliability measurement.

F.4 Sensitivity to Structural Library Size

We rebuild the library using a 50% subsample of DLR-Web and re-run the synthesis pipeline. Table 14 compares the resulting downstream performance with the full-library configuration. The base model uses no synthesized training data and serves as a reference, rather than a library-size condition.

Table 14: Library-size sensitivity across both domains. Entries and reported averages are displayed to one decimal place.

Library	MMLU-Pro	SuperGPQA	HLE	BBH	HellaSwag	DROP	Avg
Humanities & Social Sciences							
Base (no SFT)	32.7	29.2	9.1	43.5	44.3	60.9	36.6
DLR-Web 50%	50.1	37.4	13.6	65.7	75.6	67.6	51.7
DLR-Web 100%	52.4	40.3	15.1	67.6	77.8	69.0	53.7
Library	**MMLU-Pro**	**SuperGPQA**	**HLE**	**HealthBench**	**PubMedQA**	**MedQA**	**Avg**
Medical & Health							
Base (no SFT)	51.0	26.1	13.6	63.0	56.4	57.1	44.5
DLR-Web 50%	56.7	35.6	14.9	68.0	71.3	60.5	51.2
DLR-Web 100%	61.1	41.4	19.6	71.7	76.6	64.4	55.8

The half-library configuration remains above the base model in both domains, but falls below the full library by 2.0 points on HSS and 4.6 on Medical. This indicates a larger sensitivity to library coverage in the medical setting. Subsampling a single seed source does not test robustness to a different seed dataset, and these results do not establish that domain coverage is irrelevant.

F.5 Additional Domain: Agronomy

An additional experiment uses 500 target documents from agronomy journals and reuses the existing 189,664-entry structural library. The library is reported to contain no entries assigned to agronomy. Table 15 reports performance on the Agronomy subset of SuperGPQA.

Table 15: Additional agronomy evaluation. The same structural library is reused without adding agronomy entries. Scores are reported accuracy (%).

Model	SuperGPQA Agronomy
Qwen3-8B-Base	28.6
Qwen3-8B-Instruct	34.5
Qwen3-8B-Base + DataStamp	**37.5**

DataStamp improves accuracy by 8.9 percentage points over the base model and 3.0 over the instruct reference. This extends the evaluation beyond the two main domain families. Absence of an agronomy category in the library does not establish absence of related content in other disciplines or in the model's pretraining data. The experiment therefore supports transfer to an additional target domain, but does not by itself establish adaptation to an unseen-pretraining domain or isolate a purely structural causal mechanism.

G Case Study

This appendix examines a cross-domain projection from statistics to political economy. The example reuses a two-chain critique followed by a combined verdict; shared task structure does not establish causal equivalence between the two fields.

G.1 STATISTICS → POLITICAL ECONOMY

A library entry on *annualizing beta and CAPM-derived beta* is projected onto an HSS target from *The Mystery of Capital* on *U.S. property-system reform in the early republic*. Source idx 81360 (*Business Administration, Problem-solving, Hard*).

Logic graph. Figure 6 shows the logic graph $\mathcal{G}^{\mathrm{L}}$: eight typed nodes with conditional fan-out at H (out-degree 2) and a multi-cause merger at G (in-degree 2).

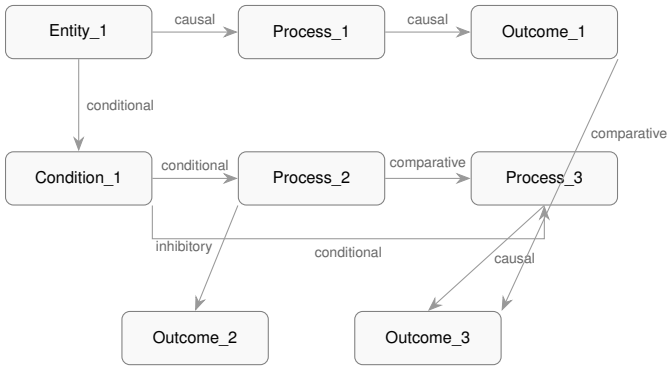


Figure 6: Logic graph $\mathcal{G}^{\mathrm{L}}$ for the statistics source (library idx 81360). H fans out conditionally; G has in-degree 2.

Transfer graph. Figure 7 shows the transfer graph $\mathcal{G}^{\mathrm{X}}$: two parallel critique sub-chains merge at synthesis node K (13 nodes, 12 edges, diamond topology with branching at D and merging at K).

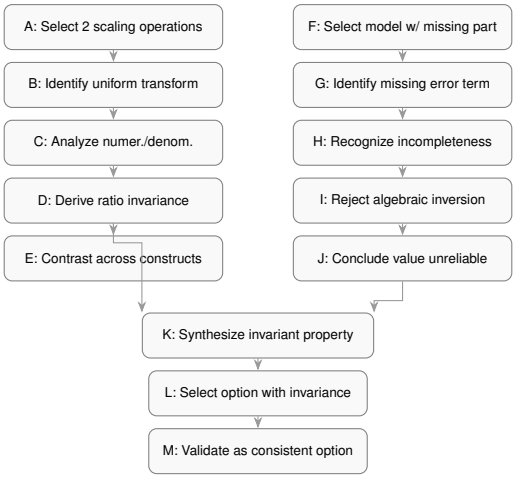


Figure 7: Transfer graph $\mathcal{G}^{\mathrm{X}}$ for the statistics-to-political-economy projection. Two critique chains converge at K; D has out-degree 2.

Source document (Statistics, idx 81360, abridged)

Annualizing Beta: can we annualize beta? We simply multiply each monthly return by 12. But what happens? We get the exact same result. Since we are uniformly altering both portfolio and benchmark returns, the covariance and variance are adjusted uniformly, resulting in the same answer. Furthermore, "Beta is the slope of the regression equation; it makes absolutely no sense to annualize it."
Deriving beta from CAPM: Rather than use the standard covariance/variance formula, why not derive beta directly from CAPM? Unfortunately, this doesn't work: Fama/French showed that expected return isn't derived just from beta (3-factor model), and the formula is missing an "error term." The "=" sign should be replaced by "≠", negating the algebraic approach.

Source question (Statistics)

A portfolio manager claims: "Since we annualize volatility by multiplying monthly volatility by $\sqrt{12}$, we should similarly annualize beta by multiplying monthly beta by 12. Additionally, beta can be derived algebraically from the CAPM equation: $E(R_p) = R_f + \beta(E(R_m) - R_f)$." Critically evaluate this claim. Beta is defined as $\mathrm{Cov}(R_p, R_m)/\mathrm{Var}(R_m)$. Which statement is *irrefutably correct*? A) Annualizing beta by scaling by 12 is valid. B) Beta remains unchanged under uniform scaling because covariance and variance scale identically. C) Beta can be derived from CAPM via algebra. D) Annualizing beta requires geometric compounding. E) CAPM-derived beta is reliable because the equation holds exactly.

Source answer (Statistics, abridged)

Two critical errors. First, scaling returns by 12 scales both covariance and variance by 144, leaving beta unchanged — beta is a regression slope requiring no temporal adjustment. Second, deriving beta from CAPM algebraically is flawed: CAPM is an incomplete expectational model with no error term, and real returns deviate due to omitted factors (Fama-French). Only B is correct: uniform scaling preserves beta due to proportional covariance/variance scaling.

The following projected pair was produced by applying the same transfer graph to the HSS target via P3.

Target document (HSS, *The Mystery of Capital*, abridged)

Between 1785 and 1890, the United States Congress passed more than 500 laws to reform the property system, based on the Jeffersonian ideal of putting property into private hands. The complicated procedures often hampered this goal. Individual states developed their own rules that largely protected their propertied elite. Attempts to reform only heightened difficulties while making migrants wary about losing their title. One contemporary emphasized that "many of the inhabitants derive the security of their estates from this confusion. . . [and] many dare not assert their rights, from a fear of being obliged to pay considerable indemnifications."

Projected question (Political economy)

Critically evaluate the claim that U.S. federal and state efforts to reform the property system in the early republic effectively secured land titles for migrants. Analyze how uniform legal reforms applied across states with differing local rules affected property rights stability, and explain why the resulting confusion undermined the goal of securing private ownership. Support your argument with evidence regarding reform processes, elite interests, and the psychological impact on landholders.

Projected answer (Political economy, abridged)

The claim is undermined by the evidence. Two parallel processes were intended to converge: (i) over 500 federal laws applying a uniform Jeffersonian transformation, and (ii) state-specific rules applied in parallel. Their combined effect was fragmentation rather than convergence: state implementations primarily protected their own elite, so the same transformation was applied inconsistently. The psychological consequence was that many landholders refrained from asserting their rights for fear of indemnification. The synthesis is that uniform legal procedures do not guarantee invariant outcomes across heterogeneous contexts; the reforms, despite their scale, entrenched uncertainty rather than resolving it.

Structural correspondence and scope. The projection reuses a two-chain critique followed by a combined verdict. In the target passage, federal reforms and state-level rules provide two strands of evidence for evaluating the claim. This is an analogy in task organization, not an equivalence between statistical invariance and property rights: uniform legislation is not itself an invariance law. The case illustrates reuse of a question scaffold rather than transfer of a causal theory between the two fields.

H LIMITATIONS

Both logic graph and transfer graph extraction rely on LLM prompting, so extraction errors on ambiguous passages propagate through the pipeline. The fixed type vocabularies balance expressiveness with matching coverage but may miss finer-grained structural distinctions. We evaluate on two domain families (HSS and medical). Finally, the transfer graph library is constructed once and frozen; iterative refinement based on downstream feedback is left to future work.

I BROADER IMPACT

DATASTAMP aims to democratize access to high-quality instruction data for specialized domains, potentially benefiting practitioners in resource-limited settings such as healthcare in low-resource regions or humanities research at smaller institutions. The interpretability of logic graphs and transfer graphs supports auditability of the synthesis process. However, as with any data synthesis method, the framework could produce misleading training data if applied without verification, and it inherits biases present in the source documents and extraction LLM. We recommend that practitioners always

apply the verification step, manually inspect a random sample of synthesized pairs, and evaluate fine-tuned models on held-out benchmarks with attention to fairness and safety before deployment.