STRUCTURED KNOWLEDGE DEPENDENCY ESTIMATION FOR QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Question answering (QA) systems draw on knowledge from task context, retrieval, decomposition, and intermediate reasoning. These sources form a candidate knowledge space whose units may be redundant, complementary, noisy, or incorrect, making their influence difficult to assess independently. In particular, the effect of removing a knowledge subset can differ from the sum of its individual parts, while exhaustive subset interventions are costly. We formulate this challenge as *knowledge dependency estimation*, defining dependency by the behavioral change of a fixed QA model when subsets of candidate knowledge are removed. We propose **KNOT**, a structured estimator learned from offline interventions. KNOT maps candidates into shared latent factors and uses removal–retention coverage to model both the coverage carried by a removed subset and the coverage preserved by the remaining candidates. The same function predicts subset effects and yields unit-level dependency scores through singleton evaluation. Across five QA benchmarks, KNOT improves zero-call subset-effect prediction and knowledge ranking over amortized attribution baselines, and supports substantial context compression while preserving near-full-context answer quality.

1 INTRODUCTION

Large language models (LLMs) have substantially advanced question answering (QA), yet they can still produce fluent but factually incorrect answers (Manakul et al., 2023). Detecting such errors reveals whether an answer is unreliable, but not which available knowledge actually shaped the model's prediction. Understanding this dependency helps characterize how a model uses available knowledge and can inform knowledge selection, context compression, and targeted examination of influential units. This raises a fundamental behavioral question: *which of the available knowledge does the final answer actually depend on?*

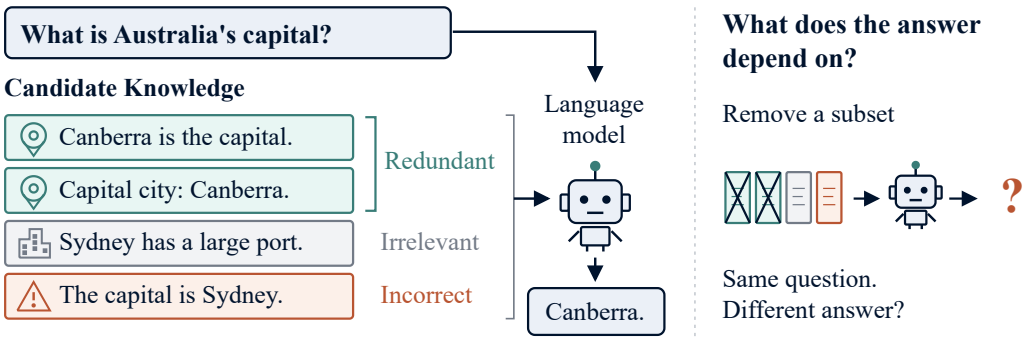

Figure 1: **Which knowledge does an answer depend on?** Candidate knowledge may be redundant, irrelevant, or incorrect. Removing a subset while keeping the question fixed probes its influence on the model's answer.

As illustrated in Fig. 1, modern QA systems may draw on heterogeneous candidate knowledge from task context, retrieval (Lewis et al., 2020), decomposition (Ammann et al., 2025), and intermediate

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.

reasoning (Wei et al., 2022). These units may be redundant, complementary, irrelevant (Shi et al., 2023), or conflict with the model's parametric knowledge (Xie et al., 2024), making their influence difficult to assess independently. Prior work has approached this problem from several directions: retrieval methods prioritize relevant passages, evidence and citation methods assess answer support, and behavioral attribution measures how model outputs change when inputs are perturbed or removed (Covert et al., 2021; Cohen-Wang et al., 2024; Liu et al., 2025). Amortized methods such as FastSHAP (Jethani et al., 2022) and Stochastic Amortization (Stoch.) (Covert et al., 2024) further learn reusable attribution estimators across examples, reducing repeated intervention cost.

The remaining challenge is that knowledge influence is inherently set-dependent. Redundant or substitutable units can mask one another, so removing one may have little effect while equivalent information remains; complementary units may instead matter only when considered jointly. Consequently, the effect of removing a knowledge subset need not equal the sum of its individual removal effects. Exhaustively measuring these interactions is also impractical because the number of possible subsets grows combinatorially with the candidate set. Effective dependency estimation must therefore capture interactions among candidates without requiring exhaustive test-time intervention.

We formulate this challenge as *knowledge dependency estimation*, defining dependency by the behavioral change of a fixed QA model when subsets of candidate knowledge are removed. We propose **KNOT**, a structured estimator learned from offline interventions. KNOT maps candidate knowledge into shared latent factors and introduces a *removal–retention coverage* function that models both the coverage carried by a removed subset and the coverage preserved by the remaining candidates. The same learned function predicts subset effects and yields unit-level dependency scores through singleton evaluation, providing a unified estimator for knowledge subsets and individual units.

We evaluate KNOT on five QA benchmarks spanning multiple-choice and generative settings. The results show that KNOT can predict subset effects and identify influential knowledge without additional test-time QA calls. Structural ablations further support the effectiveness of the removal–retention coverage design across subset prediction and knowledge selection. Context-compression experiments show that the learned dependency signal translates into practical knowledge filtering, preserving near-full-context answer quality with substantially shorter inputs. The results also reveal that dependency estimation provides a reusable scoring basis for candidate combinations, while identifying effective knowledge groups additionally depends on the downstream search strategy.

Our contributions are:

- We formulate knowledge dependency in QA through behavioral changes under candidate-knowledge removal and identify set dependence under redundancy, substitution, and complementarity as the central estimation challenge.
- We propose **KNOT**, a structured estimator that models subset effects with shared latent factors and removal–retention coverage, while supporting both subset prediction and unit-level dependency scoring through a single learned function.
- We systematically evaluate KNOT across subset prediction, knowledge ranking, structural ablations, and context compression, and analyze the relationship between dependency estimation and combinatorial knowledge selection.

2 Problem Formulation

We study knowledge dependency estimation for QA. Given a question q, a fixed black-box QA model f_θ, and a candidate knowledge space $C(q)$, we measure dependency by how the model's answer changes when candidate knowledge is removed.

2.1 Candidate Knowledge Space

For each question q, let

$$C(q) = \{k_1, k_2, \ldots, k_N\}, \tag{1}$$

where each k_i is an explicitly represented knowledge unit, such as task context, a retrieved passage, a decomposed information need, or an intermediate reasoning statement. Candidate units may be

useful, redundant, complementary, irrelevant, noisy, or incorrect. Appendix A.3 details how candidate knowledge is constructed from these sources.

The full-context answer and the answer after removing a subset $S \subseteq C(q)$ are

$$a = f_\theta(q, C(q)), \qquad a_S = f_\theta(q, C(q) \setminus S), \tag{2}$$

with the question and QA model held fixed.

2.2 Knowledge Dependency

We define the dependency of a knowledge subset S as

$$\mathcal{S}(q, S; C(q)) = \Delta(a, a_S; q), \tag{3}$$

where $\Delta \in [0, 1]$ measures the change between the two answers; its concrete construction is given in Section 3.1. This subset-removal effect measures how making S unavailable changes the model's answer. Because candidate units may be redundant, substitutable, or complementary, subset-removal effects need not decompose into independent unit-removal effects. We therefore learn

$$\hat{\mathcal{S}}_\Theta(q, S; C(q)) \approx \mathcal{S}(q, S; C(q)). \tag{4}$$

Unit-level dependency is the singleton case:

$$\mathcal{S}(q, k_i; C(q)) = \mathcal{S}(q, \{k_i\}; C(q)), \qquad \hat{\mathcal{S}}_\Theta(q, k_i; C(q)) = \hat{\mathcal{S}}_\Theta(q, \{k_i\}; C(q)). \tag{5}$$

Thus, the same estimator supports both subset- and unit-level dependency estimation. Learning from multi-unit removals allows the estimator to model knowledge combinations while producing unit-level scores without additional per-candidate QA-model calls at inference time.

3 Methodology

Fig. 2 summarizes KNOT. We describe its supervision, candidate representation, structured coverage estimator, and unified scoring and training below.

3.1 Intervention-based Supervision

For each training question q with candidate knowledge $C(q)$, we construct non-empty removal subsets $S \subseteq C(q)$ containing either single or multiple units. The fixed QA model is evaluated with the full candidate set and with S removed:

$$a = f_\theta(q, C(q)), \qquad a_S = f_\theta(q, C(q) \setminus S). \tag{6}$$

We define the observed dependency target by combining correctness degradation and semantic answer change:

$$\mathcal{S}(q, S; C(q)) = \alpha[c(a; q) - c(a_S; q)]_+ + (1 - \alpha)\delta(a, a_S), \tag{7}$$

where $c(\cdot; q) \in \{0, 1\}$ denotes correctness, $[x]_+ = \max(x, 0)$, and $\delta \in [0, 1]$ measures semantic change. We use $\alpha = 0.4$. The first term captures degradation in answer correctness, while the second captures changes in answer content even when correctness is unchanged.

Correctness is determined by normalized answer or option matching, with ambiguous cases judged by majority voting among three language models. Let $e(a, b)$ denote the entailment probability from answer a to b, estimated through natural language inference (NLI); semantic change is

$$\delta(a, b) = 1 - \min\{e(a, b), e(b, a)\}, \tag{8}$$

with identical answers assigned zero change. Appendix A provides subset construction, prompting, judging, semantic scoring, and supervision-quality analyses.

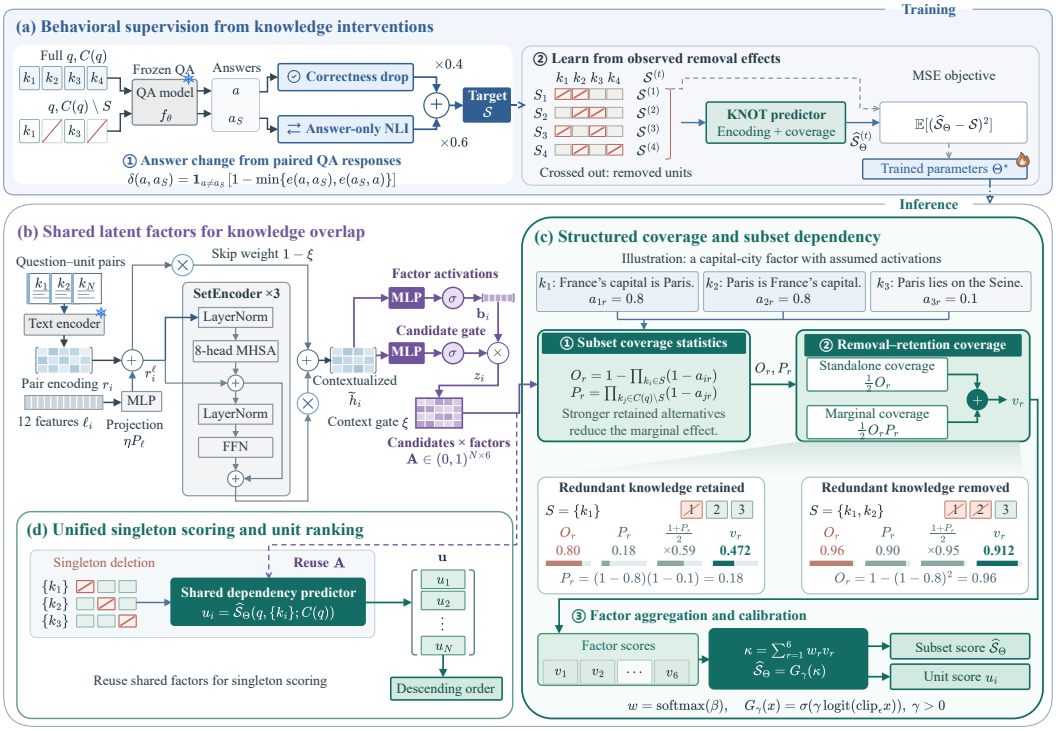

Figure 2: KNOT: from observed answer changes to knowledge-dependency estimates. (a) Offline removal interventions provide supervision from paired QA responses. (b) A shared set encoder maps candidate knowledge into latent-factor activations. (c) Removal–retention coverage combines standalone and marginal coverage to predict subset-removal effects. (d) The same predictor evaluates singleton removals for unit-level ranking.

3.2 Candidate Representation

Each candidate k_i is encoded together with the question using a frozen Qwen2.5-1.5B-Instruct encoder (Yang et al., 2024). The input contains the question, original answer options when applicable, and candidate text, truncated to 2,048 tokens. Mean pooling gives

$$r_i = \mathrm{MeanPool}(\mathrm{Enc}(q, k_i)), \qquad r_i \in \mathbb{R}^d. \tag{9}$$

We augment r_i with a 12-dimensional lexical feature vector ℓ_i encoding candidate length, question–candidate overlap, candidate-set size, and similarity to other candidates:

$$r_i^\ell = r_i + \eta P_\ell(\ell_i), \tag{10}$$

where P_ℓ is learned and η controls the fusion strength. Full feature definitions are given in Appendix B.2.

To model relations among candidates, we use a three-layer Transformer (Vaswani et al., 2017) set encoder with eight-head self-attention and no positional encoding:

$$h_{1:N} = \mathrm{SetEnc}(r_{1:N}^\ell). \tag{11}$$

A learned residual gate combines the original and set-contextualized representations:

$$\tilde{h}_i = r_i^\ell + \xi(h_i - r_i^\ell), \qquad \xi \in (0, 1). \tag{12}$$

Thus, $\tilde{h}_i$ retains candidate-specific information while incorporating interactions across the candidate set.

3.3 Removal–Retention Coverage

KNOT maps each $\tilde{h}_i$ into a shared R-factor space:

$$z_i = \sigma(g_z(\tilde{h}_i)), \qquad b_i = \sigma(g_b(\tilde{h}_i)), \qquad a_i = z_i b_i, \tag{13}$$

where $g_z : \mathbb{R}^d \to \mathbb{R}$ and $g_b : \mathbb{R}^d \to \mathbb{R}^R$. Thus $a_i \in (0,1)^R$, and a_{ir} denotes the activation of candidate k_i on factor r. Stacking all candidates gives $A = [a_{ir}] \in (0,1)^{N \times R}$, where $N = |C(q)|$. The main configuration uses $R = 6$, selected by the capacity ablation.

For a removal subset S, define the coverage carried by the removed subset and the residual non-coverage among retained candidates for factor r as

$$O_r(S) = 1 - \prod_{k_i \in S} (1 - a_{ir}), \qquad P_r(C(q) \setminus S) = \prod_{k_j \in C(q) \setminus S} (1 - a_{jr}). \tag{14}$$

The noisy-OR aggregation (Pearl, 1988) in $O_r(S)$ saturates overlapping activations, while larger $P_r(C(q) \setminus S)$ indicates that the factor is less represented by retained candidates. We define

$$v_r(S) = \frac{1}{2} O_r(S) \left[1 + P_r(C(q) \setminus S)\right]. \tag{15}$$

This averages the standalone coverage $O_r(S)$ and the marginal coverage $O_r(S)P_r(C(q) \setminus S)$. Under noisy-OR,

$$O_r(C(q)) - O_r(C(q) \setminus S) = O_r(S) P_r(C(q) \setminus S), \tag{16}$$

so the marginal term is exactly the factor coverage lost after removing S. Appendix B.1 provides the derivation of this retained-context adjustment.

Let $v(S) = (v_1(S), \ldots, v_R(S))^\top$. Factor-wise coverage is aggregated as

$$\kappa(S) = w^\top v(S), \qquad w = \mathrm{softmax}(\beta), \tag{17}$$

where $\beta \in \mathbb{R}^R$ and $\kappa(S) \in [0,1]$. A monotonic calibration produces the dependency prediction:

$$\widehat{\mathcal{S}}_\Theta(q, S; C(q)) = G_\gamma(\kappa(S)), \qquad G_\gamma(x) = \sigma(\gamma \,\mathrm{logit}(\mathrm{clip}_\epsilon(x))), \tag{18}$$

where $\gamma > 0$ is learned and $\mathrm{clip}_\epsilon(x) = \min(1 - \epsilon, \max(\epsilon, x))$. We define $\widehat{\mathcal{S}}_\Theta(q, \varnothing; C(q)) = 0$.

3.4 Unified Scoring and Training

Unit-level dependency is obtained from the same estimator by setting $S = \{k_i\}$:

$$v_r(\{k_i\}) = \frac{a_{ir}}{2} \left[1 + \prod_{\substack{k_j \in C(q) \\ j \neq i}} (1 - a_{jr})\right], \tag{19}$$

and

$$u_i = \widehat{\mathcal{S}}_\Theta(q, \{k_i\}; C(q)) = G_\gamma\left(\sum_{r=1}^{R} w_r v_r(\{k_i\})\right). \tag{20}$$

Sorting u_i gives the unit-level knowledge ranking.

Let $\mathcal{D}$ denote the intervention-based training set. KNOT minimizes subset-level mean squared error (MSE):

$$\mathcal{L}_{\mathrm{dep}} = \frac{1}{|\mathcal{D}|} \sum_{(q,C,S) \in \mathcal{D}} \left[\widehat{\mathcal{S}}_\Theta(q, S; C) - \mathcal{S}(q, S; C)\right]^2. \tag{21}$$

The lexical projection, set encoder, candidate gate, factor mapping, factor weights, and calibration are trained jointly, while the Qwen2.5-1.5B-Instruct encoder remains frozen. Unit-level scores use no separate supervision; checkpoint selection uses question-macro development MSE.

4 EXPERIMENTS

We evaluate KNOT from four perspectives: subset-effect prediction (RQ1), dependency-guided knowledge selection (RQ2), coverage structure (RQ3), and downstream utility (RQ4).

4.1 EXPERIMENTAL SETUP

Datasets and evaluation conditions. We construct KNOT datasets from five QA benchmarks: MMLU (Hendrycks et al., 2021), MedQA (Jin et al., 2021), TruthfulQA (Lin et al., 2022), SQuAD (Rajpurkar et al., 2016), and HotpotQA (Yang et al., 2018). The train/dev/full-test splits contain 5,000/327/1,666 questions. We additionally construct *Test500* with 100 questions per benchmark. *Ordinary* allows supplied and parametric knowledge, *Constrained* requires answers to rely on supplied knowledge, and *Retrieval* replaces candidate knowledge with task context and retrieved passages. The full test follows the constrained-QA setting used for training and is used for additional RQ1 evaluation; RQ2 uses all three Test500 settings, while RQ3 and RQ4 use Ordinary Test500. Data construction and evaluation populations are detailed in Appendix A.

Baselines, metrics, and implementation. We compare KNOT with feature-based baselines (Size-only Ridge, HGB, and Lexical Relevance), amortized attribution methods (FastSHAP (Jethani et al., 2022) and Stochastic Amortization (Covert et al., 2024)), and methods using additional target-model observations (LLM-Rank, LOO, MC-S4/MC-S16, and ContextCite (Cohen-Wang et al., 2024)). KNOT+5 reranks KNOT's top five units with LOO, while KNOT-Set directly applies the learned subset predictor to greedy group selection. Subset prediction uses MAE, RMSE, Pearson, and Spearman (Sp.); selection uses NDCG and joint Drop/Suff interventions; compression uses answer accuracy and input-token reduction. The target QA model is Qwen3.5-35B-A3B[1]; KNOT uses a frozen Qwen2.5-1.5B-Instruct encoder, three set-encoder layers, and six latent factors. Complete baseline comparisons, metric protocols, and optimization settings are provided in Appendices C and B.2.

4.2 RQ1: SUBSET-EFFECT PREDICTION

RQ1 asks whether KNOT can predict unseen subset-removal effects without additional target-model calls. Table 1 reports the main Test500 results.

KNOT attains the lowest zero-call MAE on both Ordinary (.220) and Constrained (.250), and leads the remaining correlation metrics except Ordinary Spearman (.427 vs. .513 for Size-only Ridge). Methods using additional target-model observations can achieve lower error at substantially higher inference cost. Monte Carlo methods directly perturb candidate knowledge and observe QA behavior at test time, so more sampled interventions provide denser empirical evidence over knowledge combinations, helping explain the strong performance of MC-S16. Overall, KNOT provides the strongest and most consistent zero-call subset-effect prediction across the evaluated Test500 settings.

On the larger constrained full test, KNOT also achieves the best zero-call performance across all four metrics (Appendix Table 11). Complete results, protocol details, and evaluations on two additional target QA models are provided in Appendices D.1, C.2, and D.2.

4.3 RQ2: DEPENDENCY-GUIDED KNOWLEDGE SELECTION

RQ2 asks whether dependency estimates can guide knowledge selection. Fig. 3 summarizes unit ranking and joint interventions on Ordinary and Constrained Test500. KNOT ranks by singleton dependency, KNOT+5 reranks the top five with LOO, and KNOT-Set directly uses the subset predictor for group selection. Drop measures answer change after jointly removing the selected 25% or 50%, while Suff measures answer preservation when only that fraction is retained.

Among zero-call ranking methods, KNOT achieves the best results on all five metrics in both Ordinary and Constrained Test500. Under Retrieval, KNOT retains the strongest zero-call NDCG@3, while lexical and intervention-based methods achieve larger Drop effects, suggesting that ranking transfers more consistently than joint-intervention effects. KNOT+5 substantially improves ranking

[1] https://huggingface.co/Qwen/Qwen3.5-35B-A3B

Table 1: Subset-effect prediction on Ordinary and Constrained Test500. Bold indicates the best result within each observation group.

		Ordinary Test500			Constrained Test500		
Method	Calls/Q	MAE↓	RMSE↓	Sp.↑	MAE↓	RMSE↓	Sp.↑
No additional target-model observations							
Lexical Relevance	0	.650	.750	.256	.603	.735	.315
Size-only Ridge	0	.254	.304	**.513**	.324	.381	.339
Text+Lex HGB	0	.245	.316	.421	.299	.387	.334
FastSHAP	0	.247	.317	.366	.277	.362	.407
Stoch.	0	.249	.308	.414	.296	.371	.380
KNOT	0	**.220**	**.297**	.427	**.250**	**.348**	**.456**
Additional target-model observations							
LLM-Rank	1†	.654 (−.434)	.762	.199	.607 (−.357)	.748	.298
LOO	13	.185 (+.0027)	.339	**.746**	**.167** (+.0064)	.368	.624
MC-S4	49	.140 (+.0016)	.249	.624	.184 (+.0013)	.327	.626
ContextCite	65	.254 (−.0005)	.403	.525	.378 (−.0020)	.560	.589
MC-S16	193	**.133** (+.0005)	**.235**	.688	.178 (+.0004)	**.302**	**.660**

Calls/Q reports additional QA-model observations under the intended $N = 12$ candidate construction; preprocessing may leave fewer candidates for some questions. Parentheses after MAE show the MAE gain per additional observation over zero-call KNOT. Negative values indicate worse MAE than KNOT. †LLM-Rank required retries after invalid first outputs on 111/500 (22.2%) Ordinary Test500 questions; retries are not counted. Actual average observation counts and complete results are given in Appendix C.

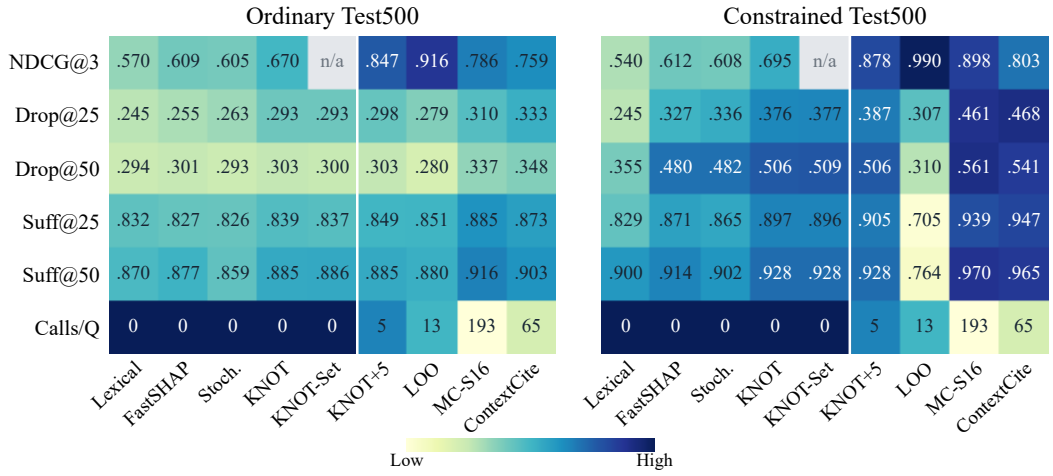

Figure 3: Knowledge ranking and joint-intervention effects on Ordinary and Constrained Test500. KNOT-Set selects subsets separately for Drop and Suff, so NDCG@3 is not applicable.

but changes Drop/Suff only slightly. KNOT-Set, which directly applies the learned subset predictor to group selection, yields small and mixed changes relative to singleton-based selection.

Overall, RQ2 shows that learned dependency provides an effective signal for knowledge selection: KNOT achieves strong zero-call ranking and joint-intervention results across the main settings. The results further show that better unit ranking or direct subset scoring does not necessarily produce consistent gains in joint selection. Selecting the best knowledge subset therefore depends not only on dependency estimation, but also on how candidate combinations are searched. Complete comparisons and intervention analyses are provided in Appendices D.3, D.4, and D.5.

Table 2: Matched structural controls on Ordinary Test500. Best results are bold.

	Subset prediction				Ranking	Drop ↑		Suff ↑	
Variant	MAE ↓	RMSE ↓	Pearson ↑	Sp. ↑	NDCG@3 ↑	25%	50%	25%	50%
KNOT	**.190**	.293	**.551**	.421	**.641**	**.280**	**.297**	**.824**	.877
Noisy-OR only	.207	.289	**.551**	.410	.616	.268	.292	.811	.869
MLP decoder	.236	**.286**	.549	**.473**	.585	.254	.261	.811	.829
No lexical fusion	.214	.308	.482	.364	.622	.267	.285	.809	**.878**

4.4 RQ3: Coverage Structure

RQ3 examines whether KNOT's structured coverage improves over generic set modeling. Table 2 compares the full model with three controlled variants: *Noisy-OR only*, which removes the retained-context term; a *multilayer perceptron (MLP) decoder*, which replaces structured coverage with a generic nonlinear set decoder; and *No lexical fusion*. All variants use the same training data, encoder, subset-MSE objective, and development protocol, isolating the effect of the coverage structure and lexical features. The 1,000-question training subset contains 12,552 removal observations.

KNOT achieves the lowest MAE and highest NDCG@3, while also giving the strongest Drop@25/50 and Suff@25. The MLP decoder performs better on RMSE and Spearman, and removing lexical fusion slightly improves Suff@50, but these gains do not carry consistently across prediction and selection. Overall, the full coverage structure provides the most balanced performance across both tasks.

A separate capacity analysis supports the compact main configuration: with three set-encoder layers, six latent factors give the best overall prediction results among the tested factor counts. Definitions of the structural variants and full capacity results are provided in Appendices B.3 and D.6.

4.5 RQ4: Downstream Utility

RQ4 asks whether dependency ranking enables effective context compression. Fig. 4 reports answer accuracy against achieved input-token reduction on Ordinary Test500. Each method ranks candidate knowledge by its own scores and retains the highest-ranked units so that the complete QA input is as close as possible to a shared target length. We evaluate target reductions from 10% to 60%, keeping the question, answer options, prompts, and generation settings fixed. Knowledge units are retained intact, with at least one kept per question. Under this rule, the 60% target is already near the effective compression limit: roughly two-thirds of questions retain only one unit, yielding an actual reduction of about 52–53%. Higher targets increasingly reproduce the same retained contexts, so the horizontal axis reports achieved rather than target reduction.

KNOT remains competitive across a broad compression range and is strongest in the medium-to-high compression regime. At the 40% target, it achieves the highest accuracy among all compared methods (.792 at 39.7% actual reduction), exceeding the full-context accuracy of .780. At the 50% target, it remains best among zero-call methods and ties ContextCite overall (.772 at 47.8% reduction). Outside this range, FastSHAP performs best around 30% reduction, while Lexical Relevance is strongest at the most aggressive setting. A development-selected KNOT stopping policy further achieves 47.0% reduction with .783 accuracy. Overall, dependency ranking provides an effective signal for substantial context compression while maintaining near-full-context answer quality.

Complete compression results and stopping-policy analyses are provided in Appendix D.7.

5 Related Work

Knowledge Attribution in QA. Prior work has studied the role of knowledge in QA through relevance, evidential support, and behavioral influence. Retrieval-augmented QA selects passages for generation (Lewis et al., 2020), while rationale evaluation and citation methods assess whether selected content supports an answer (DeYoung et al., 2020; Gao et al., 2023). SelfCite (Chuang et al., 2025) further connects citation quality with context removal and retention. Behavioral

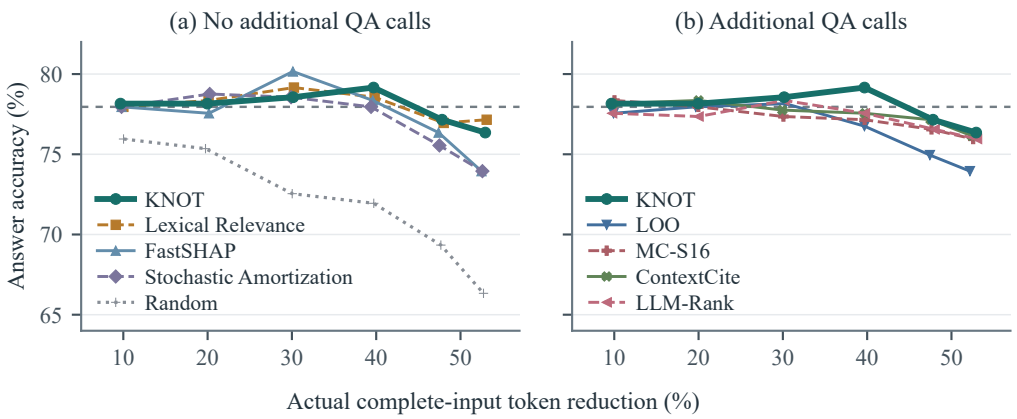

Figure 4: Context compression on Ordinary Test500. The horizontal axis shows achieved input-token reduction; dashed lines mark full-context accuracy. KNOT is shown in both panels for reference.

attribution instead measures how model outputs change under input interventions (Covert et al., 2021): ContextCite (Cohen-Wang et al., 2024) fits sparse surrogates over context ablations, AttriBoT (Liu et al., 2025) accelerates leave-one-out attribution, and Pan et al. (2026) reduce intervention cost through adaptive subset selection. Amortized methods such as FastSHAP (Jethani et al., 2022) and Stochastic Amortization (Covert et al., 2024) learn reusable attribution estimators across examples. KNOT follows this behavioral attribution perspective while adopting amortized estimation, focusing on predicting knowledge-subset effects from answer changes.

Structured Knowledge Dependency. Modeling knowledge influence becomes more challenging when candidate units overlap, are substitutable, or act jointly. Shapley–Taylor (Sundararajan et al., 2020) formalizes higher-order feature interactions, while Set Transformers (Lee et al., 2019) provide permutation-invariant architectures for modeling relations within unordered sets. In RAG, Shapley-based source attribution has been used to study redundancy and complementarity (Nematov et al., 2026), and SCARLet (Xu et al., 2025) uses interaction-aware perturbation signals to train utility-based retrievers. KNOT adopts this interaction-aware perspective by modeling dependency directly at the knowledge-subset level, while deriving unit-level dependency from the same learned function.

6 Conclusion

We proposed KNOT for knowledge dependency estimation in QA, learning reusable subset effects from offline interventions through shared latent factors and removal–retention coverage. Across five benchmarks, KNOT improves zero-call subset-effect prediction and knowledge ranking, while structural ablations support the coverage design. The learned dependency signal also transfers to downstream context compression, maintaining near-full-context answer quality under substantial input reduction.

Our results further distinguish dependency estimation from combinatorial knowledge selection. Singleton scores provide effective rankings, and the learned subset function can evaluate candidate combinations without additional QA-model observations, but finding the best knowledge subset under a downstream objective still requires an effective search strategy. This suggests a broader role for knowledge-dependency models as reusable scoring functions for knowledge analysis and selection. Promising directions include improved combinatorial search and extending the preliminary cross-model evidence to more systematic evaluation across QA models and knowledge sources.

AI Use Statement

Generative AI models were used as part of the experimental methodology described in this paper. In particular, language models were used to generate candidate subquestions and reasoning needs,

to produce QA responses under full and knowledge-removal conditions, and to support correctness judging as specified in the corresponding experimental sections. These model outputs form part of the constructed intervention data and evaluation pipeline and were processed and validated according to the procedures reported in the paper.

Generative AI tools were also used for limited editorial assistance, including language refinement, grammatical correction, and improving the clarity of presentation. They were not used to independently determine the scientific claims or conclusions of this work. All AI-assisted content and experimental procedures were reviewed and verified by the authors, who take full responsibility for the originality, accuracy, integrity, and reliability of the manuscript.

REFERENCES

Paul J. L. Ammann, Jonas Golde, and Alan Akbik. Question decomposition for retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 497–507, 2025.

Yung-Sung Chuang, Benjamin Cohen-Wang, Zejiang Shen, Zhaofeng Wu, Hu Xu, Xi Victoria Lin, James R. Glass, Shang-Wen Li, and Wen-Tau Yih. SelfCite: Self-supervised alignment for context attribution in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 10839–10858, 2025. URL https://proceedings.mlr.press/v267/chuang25a.html.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Mądry. ContextCite: Attributing model generation to context. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95762–95807, 2024. doi: 10.52202/079017-3035. URL https://papers.nips.cc/paper/2024/hash/adbea136219b64db96a9941e4249a857-Abstract-Conference.html.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

Ian Covert, Chanwoo Kim, Su-In Lee, James Zou, and Tatsunori Hashimoto. Stochastic amortization: A unified approach to accelerate feature and data attribution. In *Advances in Neural Information Processing Systems*, volume 37, pp. 4374–4423, 2024. doi: 10.52202/079017-0143. URL https://papers.nips.cc/paper_files/paper/2024/hash/084252ec5f05f13bf565843c1873686d-Abstract-Conference.html.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4443–4458, 2020.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023. URL https://aclanthology.org/2023.emnlp-main.398/.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

Neil Jethani, Mukund Sudarshan, Ian Covert, Su-In Lee, and Rajesh Ranganath. FastSHAP: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2107.07436.

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*, 11(14):6421, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, volume 97, pp. 3744–3753. PMLR, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Fengyuan Liu, Nikhil Kandpal, and Colin Raffel. AttriBoT: A bag of tricks for efficiently approximating leave-one-out context attribution. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 9004–9017, 2023.

Ikhtiyor Nematov, Tarik Kalai, Elizaveta Kuzmenko, Gabriele Fugagnoli, Dimitris Sacharidis, Katja Hose, and Tomer Sagi. Source attribution in retrieval-augmented generation. In Irena Koprinska, João Mendes-Moreira, and Paula Branco (eds.), *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, volume 2840 of *Communications in Computer and Information Science*, pp. 317–332. Springer, Cham, 2026. doi: 10.1007/978-3-032-19099-4_23. URL https://link.springer.com/chapter/10.1007/978-3-032-19099-4_23.

Deng Pan, Keerthiram Murugesan, Ting Hua, Nuno Moniz, and Nitesh V. Chawla. Context attribution with multi-armed bandit optimization. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11651–11662, 2026. doi: 10.18653/v1/2026.findings-acl.565. URL https://aclanthology.org/2026.findings-acl.565/.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988. ISBN 978-0-934613-73-6.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

Lloyd S Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, Princeton, 1953.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31210–31227. PMLR, 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The Shapley Taylor interaction index. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9259–9268, 2020. URL https://proceedings.mlr.press/v119/sundararajan20a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03762.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018. URL https://aclanthology.org/N18-1101/.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/99261adc8a6356b38bcf999bba9a26dc-Abstract-Conference.html.

Yilong Xu, Jinhua Gao, Xiaoming Yu, Yuanhai Xue, Baolong Bi, Huawei Shen, and Xueqi Cheng. Training a utility-based retriever through shared context attribution for retrieval-augmented language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 629–648, 2025. doi: 10.18653/v1/2025.emnlp-main.33. URL https://aclanthology.org/2025.emnlp-main.33/.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

A Datasets and Counterfactual Supervision

This section describes the source benchmarks (Appendix A.1), processed evaluation splits (Appendix A.2), and candidate construction (Appendix A.3). Appendices A.4 and A.5 specify removal sampling and hybrid labels; Appendices A.6 and A.7 examine learning contrast and label reliability.

A.1 Source Benchmarks

We construct the KNOT datasets from five QA benchmarks spanning general knowledge, medical reasoning, truthfulness, reading comprehension, and multi-hop QA (Table 3). Each benchmark retains its native question format, answer options when applicable, and supplied context. Reference answers are used only to determine answer correctness, and original multiple-choice options remain fixed during knowledge removal.

Table 3: Source QA benchmarks and their native task inputs. Processed KNOT split sizes are reported in Table 4.

Benchmark	Task	Native input
MMLU (Hendrycks et al., 2021)	General-knowledge QA	Question and options
MedQA (Jin et al., 2021)	Medical QA	Question and options
TruthfulQA (Lin et al., 2022)	Truthfulness QA	Question and options
SQuAD (Rajpurkar et al., 2016)	Reading comprehension	Question and passage
HotpotQA (Yang et al., 2018)	Multi-hop QA	Question and context passages

A.2 Processed Splits and Evaluation Conditions

Table 4 summarizes the processed data splits. The training, development, and full-test sets contain 5,000, 327, and 1,666 questions, respectively. The full test follows the constrained-QA instruction used for training supervision and serves as the large-scale zero-call evaluation in RQ1.

We additionally construct Test500 from the same five benchmarks, with 100 questions from each. Ordinary and Constrained Test500 use the same questions and candidate knowledge: Ordinary allows both supplied and parametric knowledge, while Constrained requires answers to rely on the supplied knowledge. Retrieval Test500 replaces the candidate knowledge with retrieved passages and task context and is used in RQ2 to evaluate ranking and selection under a different knowledge distribution. Ordinary Test500 is further used for the structural analysis in RQ3 and context compression in RQ4.

Table 4: Processed KNOT splits, evaluation conditions, and subset-removal observations for training and RQ1. Retrieval Test500 is used for RQ2 ranking and joint interventions rather than sampled-subset prediction.

Split / condition	Questions	Removal observations
Train	5,000	63,836
Dev	327	2,646
Full (Constrained)	1,666	13,505
Ordinary Test500	500	8,000
Constrained Test500	500	8,000
Retrieval Test500	500	–

Table 5: Benchmark composition of the processed splits. Each cell gives questions / removal observations. Each Test500 condition contains 100 questions per benchmark.

Benchmark	Train	Dev	Full test
MMLU	435 / 4,957	79 / 756	433 / 4,143
MedQA	1,698 / 19,068	54 / 218	285 / 1,215
TruthfulQA	128 / 1,471	38 / 148	181 / 786
SQuAD	1,803 / 25,677	79 / 794	395 / 3,890
HotpotQA	936 / 12,663	77 / 730	372 / 3,471

The benchmark composition in Table 5 distinguishes the heterogeneous training mixture from the benchmark-balanced Test500 evaluation. Candidate counts are computed once per question rather than once per removal observation: the train, development, and full-test sets contain 54,231, 3,632, and 18,467 units, respectively. Ordinary and Constrained Test500 share 5,080 units (10.16 per question); Retrieval Test500 contains 6,764 units, comprising 6,000 retrieved units and 764 native-context units.

A.3 Candidate Construction

Mixed candidate knowledge. Training, development, the full test, and Ordinary/Constrained Test500 use heterogeneous candidate knowledge constructed from task context, external Wikipedia knowledge, decomposition outputs, and intermediate reasoning. This design reflects the multiple forms of knowledge available in modern QA while keeping each candidate set compact enough for subset-level interventions. Task context is segmented into sentence- or short-passage-level units.

External knowledge is drawn from the English `wikimedia/wikipedia` release `20231101.en`. We first construct a question-oriented passage collection by filtering Wikipedia pages using question terms, available context, entity cues, and title information. This narrows retrieval to plausibly relevant material while retaining multiple related or overlapping candidates for the same question. External candidates are then retrieved from this collection using hybrid semantic-vector and BM25 search implemented with txtai,[2] followed by lightweight reranking based on lexical overlap, entity cues, and title matching. Short text spans are extracted from the highest-ranked passages so that individual knowledge units remain localized for removal-based intervention.

Qwen3.5-35B-A3B additionally generates concise subquestions and reasoning needs. Retrieved and generated candidates are combined with task-context units, then normalized, deduplicated, length-controlled, and capped to form compact candidate sets. The resulting candidate sets contain plausible knowledge from heterogeneous sources rather than only verified supporting evidence, allowing dependency to be studied under redundancy, substitution, and potentially unhelpful information. Table 6 summarizes the resulting source composition.

Retrieval Test500. Retrieval Test500 uses a separate full-corpus retrieval protocol over the same Wikipedia release. We build a txtai BM25 (Robertson & Zaragoza, 2009) index over 20,556,638 passages and retrieve 12 parent passages for each question. Each parent is segmented into sentence-aware spans of at most 256 Qwen2.5 encoder tokens, and the highest-scoring span is retained as one retrieved knowledge unit. Native task context is segmented with the same token limit and preserved in full. Each Retrieval Test500 candidate set therefore contains 12 retrieved units together with all segmented native-context units. This setting provides a more conventional retrieval-derived candidate distribution for evaluating dependency-guided ranking and selection beyond the mixed candidate construction used for training.

Training and development questions are disjoint from Test500 after text normalization.

[2] https://neuml.github.io/txtai/

Table 6: Training candidate-space audit, counting each question once. Words reports the median candidate length in whitespace-delimited words, not model tokens. Source percentages count candidate units: C, task context; W, Wikipedia; D, decomposition; R, reasoning; O, option discrimination.

Dataset	Questions	Units/q	Words	C%	W%	D%	R%	O%
MMLU	435	11.97	21	0.0	41.7	16.6	8.7	33.0
MedQA	1698	9.00	21	0.0	55.6	22.2	22.2	0.0
TruthfulQA	128	8.98	19	0.0	55.5	22.3	22.3	0.0
SQuAD	1803	11.85	20	24.1	42.2	16.9	16.9	0.0
HotpotQA	936	12.00	20	25.0	41.7	16.7	16.7	0.0

A.4 Removal-subset Sampling and QA

Training, development, and full test. For each question, we sample nonempty removal subsets spanning individual units, small combinations, larger groups, and complement-style removals. Single-unit removals reveal local dependency, while joint removals expose effects that can be masked by redundant or substitutable knowledge. Larger and complement-style removals examine how the answer changes when only part of the candidate knowledge remains. Together, these interventions provide learning examples across diverse knowledge combinations without enumerating the full subset space.

Table 7 summarizes seven sampling families. Singleton and random removals cover local effects and diverse combinations; high-priority, mixed-source, and task-structured subsets probe groups likely to affect the answer; background-need controls and complement removals provide contrasting interventions. Fig. 5 shows the corresponding distributions by removal size.

Table 7: Subset-sampling strategies and dependency scores in the training set.

Strategy family	Generated	Retained	Retained (%)	Mean dependency
Singleton removals	6,904	6,904	100.00	0.078
Random subsets	8,329	8,329	100.00	0.224
High-priority subsets	13,285	13,278	99.95	0.624
Mixed multi-source	2,145	2,144	99.95	0.666
Task-structured subsets	9,583	9,579	99.96	0.421
Background-need control	1,196	1,196	100.00	0.047
Complement removals	22,412	22,406	99.97	0.321
Total	63,854	63,836	99.97	0.367

Test500 subset prediction. For each Ordinary and Constrained Test500 question, we independently sample 16 distinct nonempty removal subsets across the available subset sizes. Sampling is independent of method scores and observed dependency targets, producing 8,000 subset-removal observations per condition.

Answer generation. For each sampled subset S, the target QA model is evaluated with the complete candidate knowledge and with S removed, while the question and original answer options remain fixed. Training and full-test generation follow the constrained-QA instruction, whereas Ordinary Test500 permits parametric knowledge. Generation uses Qwen3.5-35B-A3B with temperature 0, top-$p = 1$, disabled thinking, and a 96-token answer limit. Multiple-choice tasks return option text, while generative tasks produce concise answers.

A.5 Hybrid Sensitivity Labels

The dependency target in Eq. 7 combines correctness degradation and semantic answer change:

$$\mathcal{S}(q, S; C(q)) = 0.4[c(a; q) - c(a_S; q)]_+ + 0.6\delta(a, a_S).$$

The correctness term captures whether removing knowledge turns a correct answer into an incorrect one, while the semantic term records answer-content changes even when correctness is unchanged.

Correctness. Correctness is first determined by normalized reference-answer or original-option matching. Ambiguous cases are judged independently by Qwen3.5-35B-A3B, Llama-3.3-70B-Instruct[3], and Gemma-4-31B-it[4]. Agreement from any two judges determines the Boolean label; cases that remain unresolved are manually adjudicated.

Semantic change. Semantic change is computed from bidirectional entailment between the full-context answer a and the removal answer a_S using `DeBERTa-v2-xlarge-MNLI`, based on DeBERTa (He et al., 2021) fine-tuned on MultiNLI (Williams et al., 2018). Let $e(a, b)$ denote the entailment probability from a to b. We use

$$\delta(a, b) = 1 - \min\{e(a, b), e(b, a)\},$$

with identical answers assigned zero change. The bidirectional score captures substantive answer changes even when correctness is unchanged.

A.6 Observed Learning Contrast

Table 8: Dependency-score distributions and within-question contrasts. Range $\geq .05$ counts questions with an observed dependency range of at least 0.05.

Split	Questions	Records	Mean $\pm$ SD	$\mathcal{S} = 0$	$\mathcal{S} \geq .95$	Range $\geq .05$
Train	5,000	63,836	0.367 ± 0.456	55.3%	29.7%	4,073
Dev	327	2,646	0.329 ± 0.445	59.5%	24.5%	199
Full	1,666	13,505	0.334 ± 0.449	59.2%	27.8%	1,075

Table 8 summarizes dependency-score distributions and within-question contrasts across the three splits. In the training set, the mean dependency score is 0.367, with 55.3% of observations equal to zero and 18,968 observations at least 0.95. The distribution is concentrated near the endpoints, providing contrast between low- and high-impact removals.

Moreover, 4,073 of the 5,000 training questions (81.5%) exhibit a within-question dependency range of at least 0.05 across their sampled removal subsets. This within-question variation provides direct supervision for distinguishing the effects of different knowledge subsets for the same question.

Fig. 5 shows substantial overlap in dependency scores across removal sizes, indicating that the number of removed candidates alone does not determine the intervention effect.

Fig. 6 provides a complementary correctness view. On the training set, question-macro accuracy decreases from 94.6% with full knowledge to 61.8% after sampled removals, showing that the intervention supervision captures substantial changes in QA behavior.

A.7 Label Integrity and Semantic Reliability

We examine both the computational consistency of the supervision labels and the reliability of the semantic-change component.

Computational integrity. All 79,987 training, development, and full-test observations are linked to their paired QA responses and component judgments. Table 9 verifies removal-mask validity, label ranges, reconstruction of the mixed target, bidirectional NLI aggregation, and response termination. No duplicated question–mask pair is found within a split.

[3] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct
[4] https://huggingface.co/google/gemma-4-31B-it

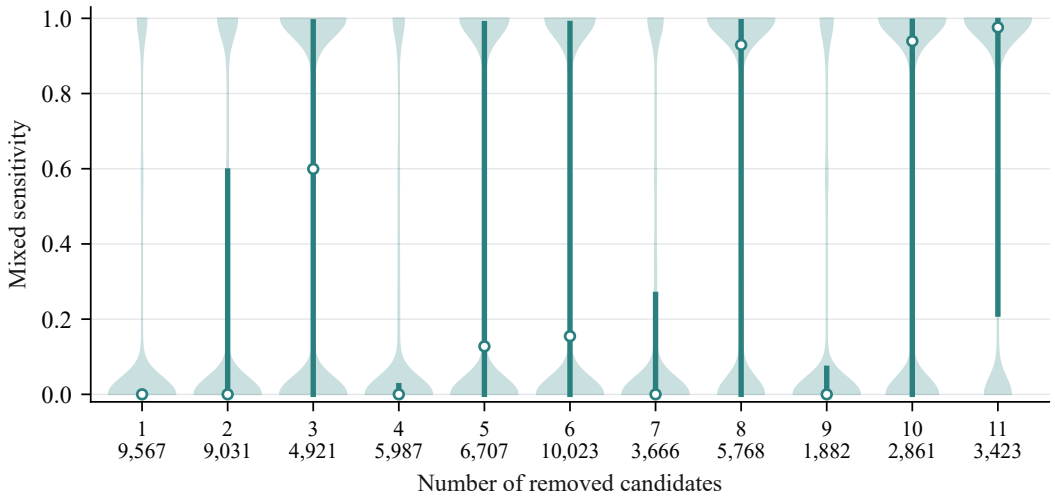

Figure 5: Dependency scores across removal sizes in the 63,836 training observations. White markers show medians and interquartile ranges; widths indicate within-group density.

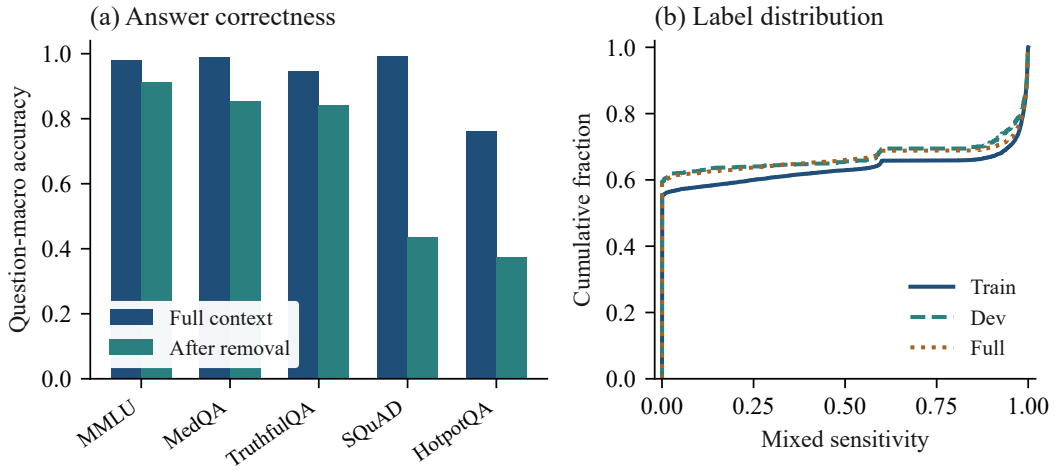

Figure 6: Behavioral effects of candidate removal. (a) Question-macro answer accuracy with full knowledge and after sampled removals. (b) Empirical cumulative distributions of dependency scores for training, development, and full-test observations.

Semantic reliability. We assess whether the semantic-change score distinguishes substantive answer changes from surface variation using two complementary checks. First, we examine answer pairs that are semantically equivalent but differ in surface form, including case/whitespace differences and verified option-prefix variants. Second, we conduct a score-masked review of 100 development answer pairs, with 20 questions from each benchmark, judging whether answer meaning changes without access to the NLI scores or correctness labels.

Fig. 7 shows complementary strengths and limitations of the semantic-change score. Identical and case/spacing variants remain concentrated near zero, whereas option-prefix variants show greater dispersion, revealing sensitivity to some answer-format changes. In the score-masked review, changed answer pairs are concentrated at high scores, while most equivalent pairs remain near zero.

Among the 100 reviewed pairs, 50 are judged semantically equivalent and 50 changed. The semantic-change score achieves an area under the receiver operating characteristic curve (AUROC) of 0.983 (95% confidence interval [0.947, 1.000]). The remaining errors are concentrated in fine-grained equivalence cases such as option-prefix variation, formatting, and numerical paraphrase. Overall,

Table 9: Integrity checks over all retained training, development, and full-test observations.

Check	Train	Dev	Full
Valid binary mask and length	63,836/63,836	2,646/2,646	13,505/13,505
Valid label range	63,836/63,836	2,646/2,646	13,505/13,505
Reproduced mixed-label formula	63,836/63,836	2,646/2,646	13,505/13,505
Reproduced NLI aggregation	63,836/63,836	2,646/2,646	13,505/13,505
Both answers stop normally	63,836/63,836	2,646/2,646	13,505/13,505
Repeated question–mask keys	0	0	0
Identical answers with nonzero change	0/35,315	0/1,574	0/7,991

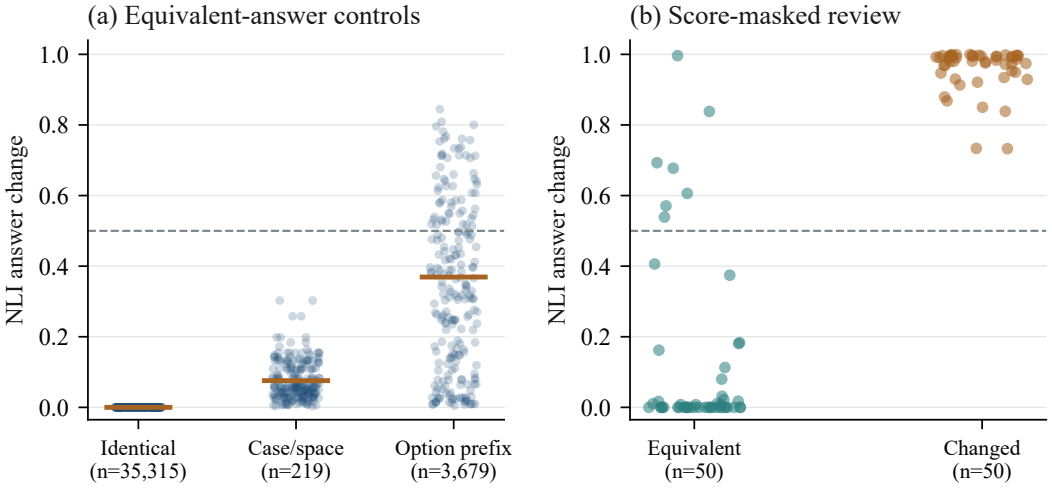

Figure 7: Reliability analysis of the semantic-change score. (a) Scores for answer pairs that are identical or differ only in case/spacing or a verified option prefix; orange bars show group means. (b) Scores for 100 development answer pairs independently judged as semantically equivalent or changed without access to the NLI score. The dashed line at 0.5 is shown as a visual reference.

the supervision provides clear within-question behavioral contrast, while the semantic component remains sensitive to some answer-format variations.

B Coverage Model: Properties and Controls

The main model represents knowledge dependency through latent-factor coverage. This section analyzes the coverage function and specifies the representation, training configuration, and structural controls used in the experiments.

B.1 Properties of the Coverage Function

For any candidate subset $U \subseteq C(q)$, the noisy-OR coverage of factor r is

$$O_r(U) = 1 - \prod_{k_i \in U} (1 - a_{ir}).$$

For a removal subset $S \subseteq C(q)$, two quantities are relevant. The first is the coverage carried by the removed subset itself,

$$O_r(S).$$

The second is the coverage actually lost after removal,

$$O_r(C(q)) - O_r(C(q) \setminus S).$$

These quantities differ when removed knowledge overlaps with what remains: S may strongly cover a factor, yet removing it may cause little loss if retained candidates provide the same coverage.

Under noisy-OR, the second quantity has a simple form. Because S and $C(q) \setminus S$ partition $C(q)$, their non-coverage factorizes:

$$1 - O_r(C(q)) = [1 - O_r(S)]P_r(C(q) \setminus S),$$

where

$$P_r(C(q) \setminus S) = \prod_{k_j \in C(q) \setminus S} (1 - a_{jr}) = 1 - O_r(C(q) \setminus S).$$

Rearranging gives

$$O_r(C(q)) - O_r(C(q) \setminus S) = O_r(S)P_r(C(q) \setminus S).$$

Thus, the coverage actually lost equals the coverage carried by the removed subset, discounted by how much of factor r remains uncovered by retained candidates.

We therefore define

$$v_r(S) = \frac{1}{2}\left[O_r(S) + O_r(S)P_r(C(q) \setminus S)\right] = \frac{1}{2}O_r(S)\left[1 + P_r(C(q) \setminus S)\right].$$

The first term captures the standalone coverage of the removed subset, while the second captures the coverage genuinely lost after removal. When retained candidates already cover the same factor, the second term becomes small; when the removed subset provides coverage unavailable elsewhere, it becomes large.

B.2 Representation and Training Configuration

Table 10 lists the architecture and optimization settings.

Each encoder input contains the question, the original answer options when applicable, and one candidate knowledge unit. We use frozen Qwen2.5-1.5B-Instruct representations and fuse them with a 12-dimensional lexical feature vector before set encoding. The features capture candidate-set size and position, candidate length and lexical diversity, question–candidate overlap, similarity to other candidates, and question length. Lexical statistics are computed from lowercased alphanumeric tokens after stopword removal. The complete vector contains a constant; normalized candidate-set size and candidate position; log-scaled word count and distinct-token count; question–candidate Jaccard overlap, its centered value, and its share of total overlap; mean and maximum Jaccard similarity to other candidates; standardized candidate length; and normalized question-token count. A learned residual gate controls how much set-level context is incorporated into each candidate representation.

The calibration in Eq. 18 preserves monotonicity. We parameterize

$$\gamma = \mathrm{clip}_{[0.1,5]}(\mathrm{softplus}(\omega)),$$

and clip the logit input as

$$\mathrm{clip}_\epsilon(x) = \min(1 - \epsilon, \max(\epsilon, x)).$$

The empty removal set is assigned a dependency score of zero.

RQ1, RQ2, and RQ4 use the same 6-factor, 3-layer checkpoint trained on 5,000 questions. RQ3 retrains the structural variants on the same 1,000-question subset under a common training setting. The text encoder remains frozen, while the lexical projection, set encoder, candidate gate, factor mapping, factor weights, and calibration are optimized jointly. Candidate representations are shared across removal subsets of the same question. Checkpoint selection uses development MSE averaged first within each question and then across questions, so each question receives equal weight.

B.3 Structural Controls

RQ3 compares KNOT with three structural variants that isolate the roles of retained-context coverage, the structured subset decoder, and lexical fusion.

Table 10: KNOT architecture and optimization settings for the reported checkpoints.

Setting	Value
Frozen text encoder	Qwen2.5-1.5B-Instruct
Maximum question–candidate input	2,048 tokens
Set-encoder layers / attention heads	3 / 8
Set-encoder dropout	0.12
Lexical feature dimension	12
Latent factors	6
Training objective	Subset MSE
Training batch size	32 subset observations
Optimizer	AdamW (Loshchilov & Hutter, 2019)
Learning rate	5×10^{-5} for the set encoder; 10^{-4} otherwise
Weight decay / gradient norm limit	10^{-4} / 1.0
Maximum epochs / early-stopping patience	10 / 3
Checkpoint criterion	Question-macro development MSE
Random seed	42

Noisy-OR only. This variant removes the retained-context term and uses only the standalone coverage of the removed subset:

$$v_r(S) = O_r(S).$$

All other components, including factor activations, factor weights, calibration, and the subset-MSE objective, remain unchanged. Singleton dependency scores are obtained from the same modified function.

MLP decoder. This variant retains the candidate representations $\tilde{h}_i$ but replaces removal–retention coverage with a generic nonlinear set decoder. Each candidate is first projected to 128 dimensions:

$$z_i = \mathrm{GELU}\Big(W\,\mathrm{LN}(\tilde{h}_i) + b\Big).$$

The subset prediction is then

$$\widehat{\mathcal{S}}_{\mathrm{MLP}}(q, S; C(q)) = \sigma\left(\mathrm{MLP}\left[\frac{1}{N}\sum_{k_i \in S} z_i\,;\ \frac{1}{N}\sum_{k_i \in C(q)} z_i\,;\ \frac{|S|}{N}\right]\right).$$

The decoder therefore receives three signals: a representation of the removed subset, a representation of the full candidate set, and the relative removal size. The MLP has one 128-dimensional hidden layer with a Gaussian error linear unit (GELU) and a scalar output. Empty removal is assigned score zero, and singleton scores are obtained by setting $S = \{k_i\}$.

No lexical fusion. This variant removes the 12-dimensional lexical feature projection and feeds the frozen text representation r_i directly into the set encoder. The set encoder, coverage function, training objective, and all remaining components are unchanged.

All variants are trained on the same 1,000-question subset and use the same frozen text encoder, development set, subset-MSE objective, and checkpoint-selection criterion. The comparison therefore isolates how the three modified components affect subset prediction and unit-level knowledge selection.

C Evaluation Protocol

KNOT produces both subset-level predictions and unit-level dependency scores, whereas several baselines natively output only unit scores. Appendix C.1 describes baseline scoring and target-model observation costs, Appendix C.2 specifies subset-effect prediction, and Appendix C.3 defines unit-ranking evaluation.

C.1 Baseline Scoring

We consider three groups of baselines. Feature-based predictors include Size-only Ridge (Hoerl & Kennard, 1970), text–lexical Ridge, MLP, HGB (Friedman, 2001), ExtraTrees (Geurts et al., 2006), and Lexical Relevance. Tables abbreviate the text–lexical input as Text+Lex. These methods use input-derived features only: the regressors predict subset-removal effects directly, while Lexical Relevance ranks candidates by question–candidate overlap.

Amortized attribution baselines include FastSHAP (Jethani et al., 2022) and Stochastic Amortization (Covert et al., 2024). FastSHAP learns unit contributions through weighted subset reconstruction, while Stochastic Amortization learns reusable unit attributions from noisy training targets. Both natively produce unit-level scores. For RQ1, we estimate a subset effect by summing the scores of units in the removal subset. This additive readout is the natural extension for methods that assign contributions to individual units but do not learn a separate set-level predictor.

Methods using additional target-model observations include LOO, Monte Carlo Shapley, LLM-Rank, ContextCite, and KNOT+5. LOO and Monte Carlo attribution require a utility that converts each intervention outcome into a scalar contribution signal. We use

$$d_{\text{lex}}(a, b) = 0.7[1 - \text{TokenF1}(a, b)] + 0.3\,\mathbf{1}[\text{norm}(a) \neq \text{norm}(b)].$$

TokenF1 provides a graded measure of answer-content change, while the normalized mismatch term captures discrete answer switches. This answer-only utility is directly computed from the paired QA responses produced by each intervention, providing a common attribution signal for LOO and Monte Carlo methods while keeping their attribution rule separate from the correctness–NLI dependency target used for evaluation.

LOO evaluates individual removals. MC-S4 and MC-S16 approximate Shapley values (Shapley, 1953) using four and sixteen sampled orders, respectively. KNOT+5 applies LOO to KNOT's five highest-ranked candidates and reranks only this prefix. LLM-Rank prompts a model to assign one score in $[0, 1]$ to each candidate. ContextCite (Cohen-Wang et al., 2024) uses 64 context perturbations and fits a sparse surrogate to the logit-transformed probability of the fixed full-context answer.

For cost comparison, Calls/Q denotes additional target-model observations under the intended $N = 12$ candidate construction. Preprocessing may leave fewer candidates, so actual mean counts can be lower. On Ordinary/Constrained Test500, LOO, MC-S4, and MC-S16 use 10.2, 36.1, and 123.7 observations per question on average; on Retrieval Test500, the corresponding values are 13.5, 49.9, and 183.8. KNOT+5 uses five observations, LLM-Rank one ranking observation, and ContextCite 64 context evaluations in addition to the full-context answer. Because these operations have different computational costs, Calls/Q measures observation count rather than runtime.

C.2 Subset Prediction

RQ1 evaluates subset-effect prediction using MAE, RMSE, Pearson, and Spearman correlation. KNOT and direct subset regressors use their learned set-level predictors. Unit-scoring methods, including FastSHAP and Stochastic Amortization, are extended to subset prediction by summing the scores of units in the removal subset, following the additive readout defined above.

The constrained full test evaluates zero-call subset prediction. Test500 additionally includes intervention-based methods such as LOO and Monte Carlo attribution, whose results use test-time target-model observations and are therefore reported separately.

C.3 Unit Ranking

RQ2 evaluates whether predicted unit scores recover the ordering of observed singleton effects and whether the resulting selections produce meaningful joint effects.

Let e_i denote the observed singleton-removal effect of candidate k_i. We use NDCG@3 (Järvelin & Kekäläinen, 2002) as the unit-ranking metric, treating e_i as the relevance of each candidate. For $k_q = \min(3, N_q)$,

$$\text{NDCG@3}(q) = \frac{\sum_{j=1}^{k_q} e_{\pi_q(j)} / \log_2(j+1)}{\sum_{j=1}^{k_q} e_{\pi_q^*(j)} / \log_2(j+1)},$$

where π_q is the predicted ranking and π_q^* is the ranking induced by the observed singleton effects. NDCG@3 therefore rewards methods that place candidates with larger observed effects near the top of the ranking.

Selected groups are evaluated separately through joint removal and retention interventions, reported as Drop and Suff; their construction is detailed in Appendix D.4.

D Additional Experimental Results

This section provides complete results and supplementary analyses corresponding to RQ1–RQ4.

D.1 Complete Subset-Prediction Results

Tables 11 and 12 report the complete RQ1 results across all four prediction metrics. The constrained full test compares zero-call subset prediction, while Ordinary and Constrained Test500 additionally include methods using test-time target-model observations, following the protocol in Appendix C.2.

Table 11: Subset-effect prediction on the constrained full test. All methods use zero additional target-model observations.

Method	MAE↓	RMSE↓	Pearson↑	Sp.↑
Lexical Relevance	.547	.693	.221	.233
Size-only Ridge	.412	.446	.176	.156
Text+Lex Ridge	.388	.444	.322	.306
Text+Lex MLP	.337	.428	.460	.418
Text+Lex HGB	.333	.411	.465	.431
Text+Lex ExtraTrees	.338	.406	.469	.432
FastSHAP	.289	.377	.573	.520
Stochastic Amortization	.298	.377	.548	.497
KNOT	**.252**	**.346**	**.639**	**.563**

Table 12: Subset-effect prediction on Ordinary and Constrained Test500.

Method	MAE↓	RMSE↓	Pearson↑	Sp.↑
Ordinary Test500 — no additional target-model observations				
Size-only Ridge	.254	.304	.490	**.513**
Text+Lex Ridge	.260	.319	.426	.423
Text+Lex MLP	.264	.342	.426	.402
Lexical Relevance	.650	.750	.259	.256
Text+Lex HGB	.245	.316	.445	.421
Text+Lex ExtraTrees	.256	.320	.423	.403
FastSHAP	.247	.317	.454	.366
Stochastic Amortization	.249	.308	.471	.414
KNOT	**.220**	**.297**	**.539**	.427
Ordinary Test500 — additional target-model observations				
LLM-Rank	.654	.762	.216	.199
LOO	.185	.339	.515	**.746**
MC-S4	.140	.249	.633	.624
MC-S16	**.133**	**.235**	**.694**	.688
ContextCite	.254	.403	.523	.525
Constrained Test500 — no additional target-model observations				
Size-only Ridge	.324	.381	.341	.339
Text+Lex Ridge	.312	.379	.372	.357
Text+Lex MLP	.306	.399	.359	.331

Table 12: Subset-effect prediction on Ordinary and Constrained Test500 (continued).

Method	MAE↓	RMSE↓	Pearson↑	Sp.↑
Lexical Relevance	.603	.735	.333	.315
Text+Lex HGB	.299	.387	.347	.334
Text+Lex ExtraTrees	.306	.382	.359	.350
FastSHAP	.277	.362	.456	.407
Stochastic Amortization	.296	.371	.403	.380
KNOT	**.250**	**.348**	**.514**	**.456**
Constrained Test500 — additional target-model observations				
LLM-Rank	.607	.748	.328	.298
LOO	**.167**	.368	.439	.624
MC-S4	.184	.327	.672	.626
MC-S16	.178	**.302**	**.722**	**.660**
ContextCite	.378	.560	.541	.589

D.2 Evaluation on Additional QA Models

We further evaluate KNOT with two target QA models beyond the main setting: Qwen3.5-9B[5] and Llama-3.3-70B-Instruct. For each target model, QA responses and intervention supervision are generated independently, and all dependency estimators are trained separately under the same protocol. Each setting contains 1,000 training, 365 development, and 500 test questions, corresponding to 12,676, 2,844, and 3,683 subset observations. The test set contains 100 questions from each benchmark.

KNOT uses the same architecture as the main experiments, with six latent factors and three set-encoder layers, and the same subset-level training objective. Baselines are trained separately under the corresponding target-model supervision.

Table 13: Subset-effect prediction with two additional target QA models. Macro MAE first averages errors within each question and then across questions. Best displayed values within each target are bold.

Method	MAE↓	RMSE↓	Pearson↑	Sp.↑	Macro MAE↓
Qwen3.5-9B					
Lexical HGB	.3079	.3932	.5251	.5178	.3028
FastSHAP	.3203	.4136	.4970	.4658	.3225
Stoch.	.3478	.4368	.5156	.5097	.3305
KNOT	**.2678**	**.3586**	**.6478**	**.6157**	**.2639**
Llama-3.3-70B-Instruct					
Lexical HGB	.3059	.3923	.4747	.4609	.2924
FastSHAP	.3449	.4844	.4331	.4479	.2914
Stoch.	.3316	.4267	.4831	.4442	.3080
KNOT	**.2966**	**.3911**	**.5834**	**.5339**	**.2668**

Table 13 shows that KNOT achieves the lowest aggregate prediction error and the highest Pearson and Spearman correlations for both target QA models. Together with the main experiments, these results show that the same dependency-estimation architecture remains effective across different target-model families when trained from model-specific intervention supervision.

[5] https://huggingface.co/Qwen/Qwen3.5-9B

D.3 Complete Knowledge-Selection Results

Tables 14, 15, and 16 report complete unit-ranking and joint-intervention results under Ordinary, Constrained, and Retrieval Test500. Fig. 8 visualizes the Retrieval setting.

Table 14: Ranking and joint-intervention effects on Ordinary Test500.

Method	NDCG@3 ↑	Drop@25 ↑	Drop@50 ↑	Suff@25 ↑	Suff@50 ↑
Lexical Relevance	0.570	0.245	0.294	0.832	0.870
Text+Lex HGB	0.587	0.250	0.286	0.811	0.868
Text+Lex ExtraTrees	0.594	0.256	0.294	0.812	0.871
FastSHAP	0.609	0.255	0.301	0.827	0.877
Stochastic Amortization	0.605	0.263	0.293	0.826	0.859
KNOT	**0.670**	**0.293**	**0.303**	**0.839**	**0.885**
Additional target-model observations					
LLM-Rank	0.649	0.269	0.298	0.846	0.870
LOO	**0.916**	0.279	0.280	0.851	0.880
MC-S4	0.745	0.294	0.307	0.863	0.883
MC-S16	0.786	0.310	0.337	**0.885**	**0.916**
ContextCite	0.759	**0.333**	**0.348**	0.873	0.903
KNOT+5	0.847	0.298	0.303	0.849	0.885
Set-selection control (no extra QA calls)					
KNOT-Set	–	0.293	0.300	0.837	0.886

Table 15: Ranking and joint-intervention effects on Constrained Test500.

Method	NDCG@3 ↑	Drop@25 ↑	Drop@50 ↑	Suff@25 ↑	Suff@50 ↑
Lexical Relevance	0.540	0.245	0.355	0.829	0.900
Text+Lex HGB	0.576	0.282	0.357	0.838	0.886
Text+Lex ExtraTrees	0.586	0.275	0.365	0.821	0.896
FastSHAP	0.612	0.327	0.480	0.871	0.914
Stochastic Amortization	0.608	0.336	0.482	0.865	0.902
KNOT	**0.695**	**0.376**	**0.506**	**0.897**	**0.928**
Additional target-model observations					
LLM-Rank	0.655	0.320	0.441	0.875	0.905
LOO	**0.990**	0.307	0.310	0.705	0.764
MC-S4	0.835	0.398	0.450	0.940	0.952
MC-S16	0.898	0.461	**0.561**	0.939	**0.970**
ContextCite	0.803	**0.468**	0.541	**0.947**	0.965
KNOT+5	0.878	0.387	0.506	0.905	0.928
Set-selection control (no extra QA calls)					
KNOT-Set	–	0.377	0.509	0.896	0.928

Table 16: Ranking and joint-intervention effects on Retrieval Test500.

Method	NDCG@3 ↑	Drop@25 ↑	Drop@50 ↑	Suff@25 ↑	Suff@50 ↑
Lexical Relevance	0.541	**0.291**	**0.330**	**0.831**	0.868
Text+Lex HGB	0.510	0.262	0.300	0.811	0.856
Text+Lex ExtraTrees	0.492	0.260	0.306	0.816	0.873
FastSHAP	0.390	0.160	0.240	0.733	0.817
Stochastic Amortization	0.394	0.162	0.256	0.748	0.846
KNOT	**0.549**	0.266	0.274	0.821	**0.877**
Additional target-model observations					
LLM-Rank	0.494	0.249	0.301	0.823	0.885
LOO	**0.919**	0.292	0.317	0.848	0.900
MC-S4	0.679	0.320	0.349	0.867	0.911
MC-S16	0.776	0.349	**0.391**	**0.892**	**0.932**
ContextCite	0.692	**0.360**	0.383	0.887	0.915
KNOT+5	0.733	0.274	0.274	0.827	0.877
Set-selection control (no extra QA calls)					
KNOT-Set	–	0.267	0.290	0.829	0.867

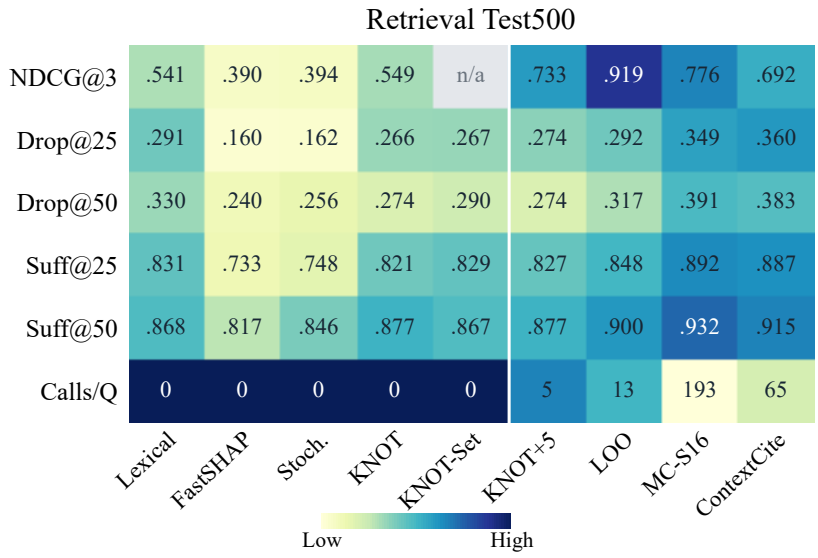

Figure 8: Knowledge ranking and joint-intervention effects on Retrieval Test500. Complete numerical results appear in Table 16.

D.4 Joint Intervention Protocol

RQ2 evaluates selected knowledge groups through joint removal and retention. For a selection fraction $p \in \{0.25, 0.50\}$, we select

$$k = \lceil pN \rceil$$

candidates. *Drop* removes the selected k candidates and measures the resulting answer change, whereas *Suff* retains only the selected k candidates and measures how much of the original answer behavior is preserved. The question, answer options, and model configuration remain fixed.

Set-specific selection. KNOT-Set directly uses the learned subset predictor to construct a group for each objective. For Drop, it greedily adds the candidate that produces the largest increase in predicted removal effect. For Suff, let R denote the currently retained group; KNOT-Set greedily adds the

candidate that most reduces the predicted effect of removing its complement,

$$\widehat{\mathcal{S}}_{\Theta}(q, C(q) \setminus R; C(q)).$$

Because Drop and Suff optimize different subset objectives, they may produce different groups. KNOT-Set therefore does not define a single unit ranking and is not evaluated with NDCG@3.

D.5 Correctness under Knowledge Selection

We further examine how dependency-guided selection affects answer correctness under the Top-25% joint interventions on Ordinary Test500. For questions answered correctly with the full context, *Removal: wrong* counts cases that become incorrect after removing the selected candidates, *Retention: correct* counts cases that remain correct when only the selected candidates are retained, and *Both* counts cases satisfying both conditions.

Table 17: Correctness outcomes under Top-25% joint interventions on Ordinary Test500. Headers give the number of initially correct questions. Best values across both method groups are bold.

Method	SQuAD (92)	HotpotQA (49)	MMLU (82)	MedQA (85)	TruthfulQA (81)
Both: removal makes the answer wrong and retention keeps it correct					
Random	6	6	0	0	1
Lexical Relevance	38	9	0	**4**	2
Text+Lex HGB	42	9	0	1	2
Text+Lex ET	41	10	0	1	1
FastSHAP	42	12	0	1	5
Stoch.	43	13	0	3	2
KNOT	**51**	12	0	**4**	4
LLM-Rank	46	10	0	2	1
LOO	43	12	0	2	2
MC-S4	48	9	**1**	1	3
MC-S16	46	12	0	2	6
ContextCite	49	**17**	0	1	**8**
KNOT+5	49	12	0	3	5
KNOT outcome decomposition					
Removal: wrong	51	15	0	5	5
Retention: correct	90	44	81	84	79

Table 17 shows that the correctness effects vary substantially across benchmarks. They are strongest on SQuAD, where KNOT satisfies both conditions for 51 of 92 initially correct questions. This pattern is consistent with the task structure: SQuAD is grounded in a supplied passage, and task-context units form a substantial part of its candidate knowledge. Removing highly dependent units can therefore remove information needed for the answer, while retaining them is often sufficient to preserve correctness.

HotpotQA also contains task-context knowledge, but its multi-hop structure can distribute answer support across multiple complementary units, making the effect of a fixed Top-25% selection less concentrated. MMLU shows almost no correct-to-incorrect transitions. Under Ordinary QA, the model may rely on parametric knowledge, and MMLU contains no native task-context candidates. MedQA and TruthfulQA similarly show weaker correctness effects and also contain no native task-context candidates.

Overall, dependency-guided selection has the clearest correctness effect when the candidate knowledge contains information directly required by the task. Parametric knowledge and support distributed across multiple complementary units can reduce the effect of removing a fixed selected subset.

D.6 Capacity Sensitivity

We examine the effect of latent-factor count and set-encoder depth on subset-effect prediction using the same 1,000-question training subset, 327-question development set, and Ordinary Test500 evaluation (Fig. 9).

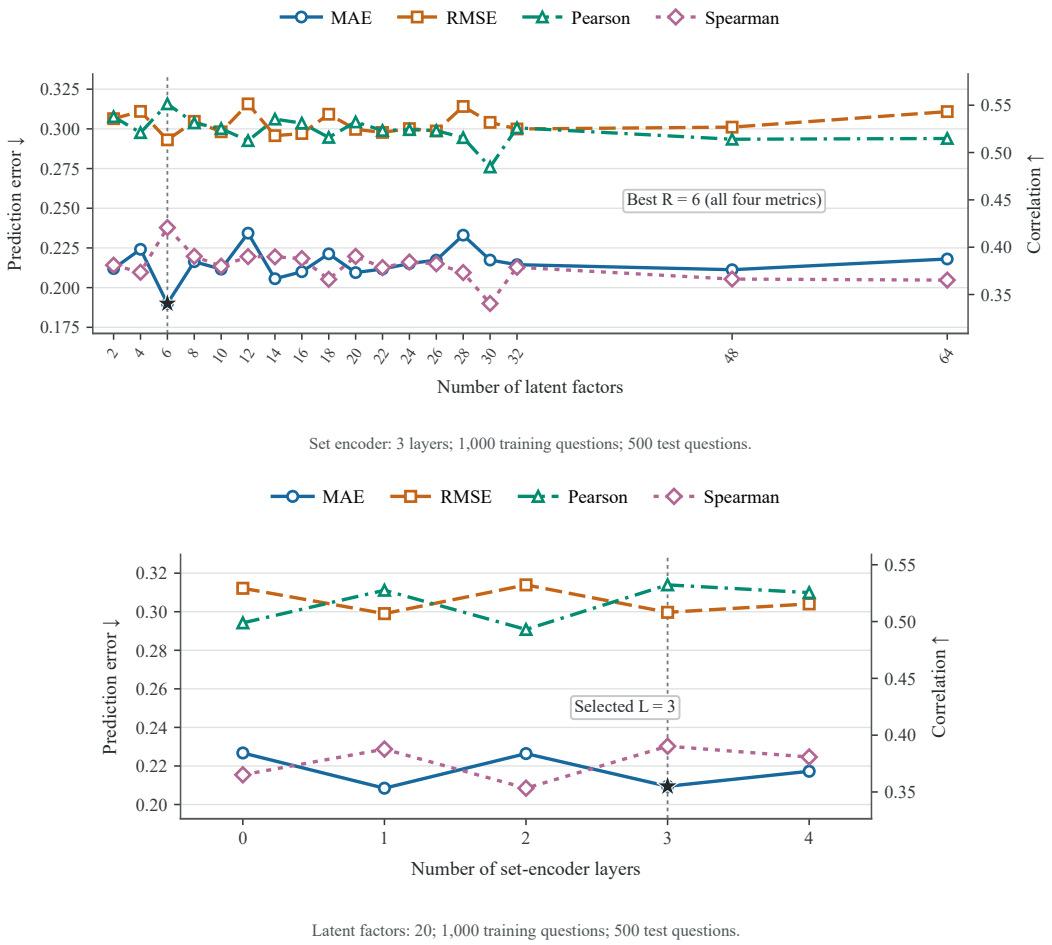

Figure 9: Capacity sensitivity on Ordinary Test500. Top: varying the number of latent factors with three set-encoder layers. Bottom: varying the number of set-encoder layers with 20 latent factors.

With three set-encoder layers, six latent factors achieve the best results across all four prediction metrics. In the depth analysis, three layers provide the strongest Pearson and Spearman correlations while maintaining near-best MAE and RMSE. We therefore use six latent factors and three set-encoder layers in the main KNOT configuration.

The factor scan covers 2–32 in steps of two, followed by 48 and 64; the depth scan covers zero through four layers with 20 factors. The two analyses vary factor count and encoder depth separately.

D.7 Context Compression

Table 18 reports the complete fixed-target compression results. The experiment evaluates nine methods on Ordinary Test500 at target input reductions from 10% to 60%. Each method ranks candidate knowledge by its own scores and retains the highest-ranked units so that the complete QA input is as close as possible to the target length. Questions, answer options, prompts, and generation settings remain fixed, and knowledge units are retained intact. Harm denotes the fraction of initially correct full-context answers that become incorrect after compression.

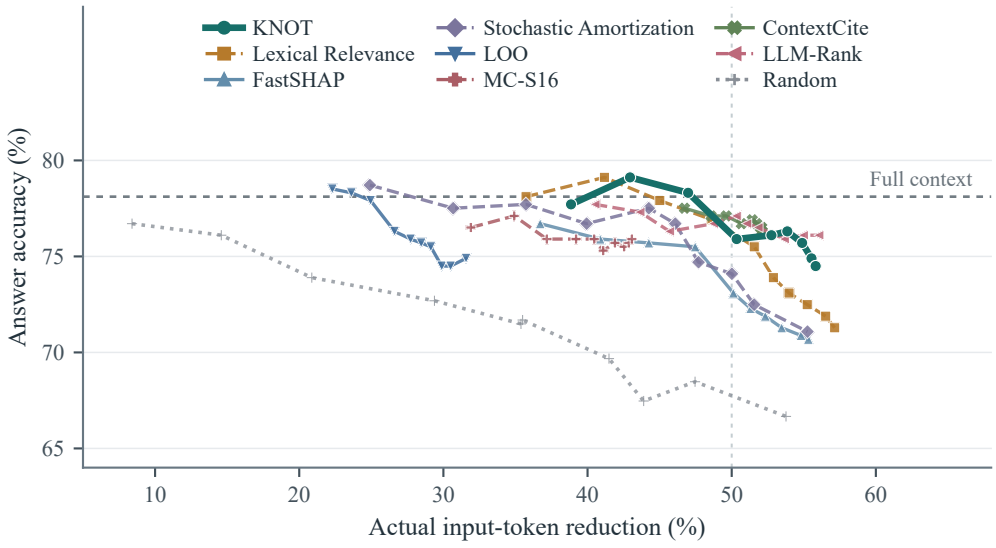

Figure 10: Threshold sensitivity of cumulative-score retention on Ordinary Test500. Each method is evaluated at ten values of τ; the horizontal axis reports achieved input-token reduction, and the dashed line marks full-context accuracy.

Token reduction is measured over the complete chat-formatted input using the target QA model's tokenizer. Because the question, options, and prompts remain fixed and at least one knowledge unit must be retained, aggressive targets cannot always be reached exactly. At the 60% target, the achieved reduction is about 52–53%, and many questions already retain only one knowledge unit. Fig. 4 therefore reports achieved input reduction.

Development-selected retention policy. We additionally evaluate an adaptive retention rule. Let s_i denote the dependency score of candidate k_i. Candidates are ranked from highest to lowest, and we retain the shortest prefix $\mathcal{R}_\tau$ whose cumulative positive score reaches a fraction τ of the total positive score:

$$\sum_{i \in \mathcal{R}_\tau} \max(s_i, 0) \geq \tau \sum_{i=1}^{N} \max(s_i, 0).$$

Thus, τ controls how much positive dependency score is retained: smaller τ keeps fewer high-scored candidates, while larger τ retains more context. For KNOT, τ is selected on Dev200 and then fixed for Test500. The selected value $\tau = .60$ yields 47.0% input reduction, .783 answer accuracy, and .023 harm.

Threshold sensitivity. We evaluate

$$\tau \in \{.10, .20, .25, .30, .35, .40, .50, .60, .70, .80\}$$

for each scoring method. Here, τ controls the fraction of positive score retained rather than the input-reduction rate. Because score distributions differ across methods, the same τ can yield different retained lengths. Fig. 10 reports achieved input-token reduction against answer accuracy.

Table 18: Complete fixed-target compression results on Ordinary Test500. Columns denote target reductions of the complete model input; achieved reductions are measured with the QA-model tokenizer. Full-context accuracy is 77.96%. Harm is the fraction of initially correct answers that become incorrect after compression. Best results are bold.

Method	10%	20%	30%	40%	50%	60%
Answer accuracy (%)						
KNOT	78.16	78.16	78.56	**79.16**	**77.15**	76.35
Lexical Relevance	77.96	78.36	79.16	78.56	76.95	**77.15**
FastSHAP	77.96	77.56	**80.16**	78.36	76.35	73.95
Stochastic Amortization	77.96	**78.76**	78.56	77.96	75.55	73.95
Random	75.95	75.35	72.55	71.94	69.34	66.33
Additional target-model observations						
LLM-Rank	77.56	77.35	78.36	77.56	76.55	75.95
LOO	77.56	77.96	78.16	76.75	74.95	73.95
MC-S16	**78.36**	77.96	77.35	77.15	76.55	75.95
ContextCite	78.16	78.36	77.76	77.56	**77.15**	76.15
Actual token reduction (%)						
KNOT	9.67	19.97	30.15	39.68	47.81	52.94
Lexical Relevance	10.01	20.11	30.24	39.81	48.00	53.10
FastSHAP	9.79	20.16	30.12	39.59	47.40	52.49
Stochastic Amortization	9.79	20.25	29.90	39.39	47.50	52.60
Random	10.03	19.75	30.04	39.71	47.67	52.73
Additional target-model observations						
LLM-Rank	9.75	20.00	30.18	39.74	48.00	53.04
LOO	9.91	19.67	30.30	39.68	47.44	52.19
MC-S16	10.02	19.98	30.07	39.69	47.63	52.58
ContextCite	9.84	20.00	29.94	39.72	47.52	52.42
Original-correct-answer harm (%)						
KNOT	2.06	2.31	1.80	2.06	3.08	4.11
Lexical Relevance	2.31	2.57	**1.03**	2.31	4.88	5.66
FastSHAP	2.06	3.08	1.80	2.83	4.63	7.20
Stochastic Amortization	1.80	2.06	2.31	3.08	4.88	7.71
Random	5.66	7.46	10.28	11.05	14.65	17.22
Additional target-model observations						
LLM-Rank	2.57	3.08	2.06	2.31	3.86	4.37
LOO	1.54	2.06	3.08	4.11	6.17	7.46
MC-S16	**1.03**	**1.29**	2.06	2.57	3.08	3.86
ContextCite	**1.03**	**1.29**	2.06	**1.80**	**2.31**	**3.08**