Domain-Specific Jargon in Large Language Models: A Comparative Analysis between General-Purpose and Specialist Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have shown remarkable proficiency on general-purpose tasks, yet their performance often degrades in highly-specialized technical domains. Moreover, little is known about how parametric knowledge of domain-specific terms is encoded within these models. We address this gap by contributing two novel medical jargon evaluation benchmarks and evaluate a general-purpose Llama-3.1 model against a variant fine-tuned on medical-domain data. Surprisingly, the general-purpose model outperforms the medically fine-tuned model on both tasks. Using mechanistic interpretability tools, we find systematic patterns of miscalibration for the medically-finetuned model. Instead of reorganizing parametric knowledge, fine-tuning simply amplifies the jargon-favoring tendencies of a small set of model components. We find that applying component reweighting strategies against the benchmark tasks successfully suppresses these components and closes the gap with the general-purpose baseline. We also observe that some jargon sensitive components transfer knowledge to the same tasks involving materials science jargon, suggesting they encode a partially domain-agnostic notion of specialized terminology. Our results highlight that domain-specific finetuning alone is insufficient in both encoding knowledge of jargon into LLMs and achieving good performance in relevant downstream tasks.

1 Introduction

Domain-specific terminology, or jargon, refers to language that professionals use that often presents significant cognitive barriers to non-expert individuals (Castellví, 1999). Jargon can take the form of a specialized term coined for technical use, such as *hydromyelia* in medicine. Alternatively, jargon can also reuse an existing word that takes on an entirely different meaning. For example, the common word *tears* can refer to physical ruptures in muscle tissue. In recent years, the natural language processing (NLP) community has seen a growing interest in domain-specific language (Wang et al., 2023; Yang et al., 2023; Ling et al., 2025), yet little attention has been paid to understanding how LLMs internally represent and acquire knowledge about domain-specific jargon.

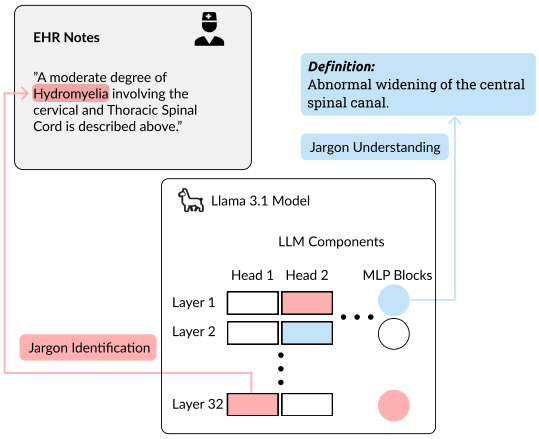

Figure 1: Overview of our framework. Given an EHR note containing a medical term (*hydromyelia*), Llama-3.1-8B model is evaluated on two tasks: Jargon Understanding (selecting the correct definition) and Jargon Identification (classifying whether the term is jargon in context). The model's outputs are then decomposed into per-component contributions from individual attention heads and MLP blocks to localize jargon knowledge.

Specialized terminology can impede effective communication between experts and the general public (Vilhena et al., 2014; Ryba et al., 2021; Cervetti et al., 2015; Bullock et al., 2019). In high-stakes domains such as medicine, the ability to accurately interpret specialized terminology has significant practical consequences. Consider a clinical AI assistant designed to summarize Electronic Health Records (EHRs) for patients. If the model misinterprets domain-specific medical jargon, it may inadvertently provide patients with misleading summaries of their health data (Croxford et al., 2025). Preventing such failure modes requires not

only measuring performance, but also understanding why models succeed or fail. To address this gap, we investigate the internal representations of domain-specific terminology in the Llama-3.1-8B model family (Grattafiori et al., 2024). We construct two complementary benchmarks that probe distinct aspects of jargon comprehension: 1) A jargon understanding (JU) task that evaluates a model's ability to interpret domain-specific meanings of jargon and 2) A jargon identification (JI) task that tests a model's ability to correctly identify jargon usages in text[1]. Figure 1 shows examples of these tasks on an EHR note.

Using these benchmarks, we compare the performance of a general-purpose Llama-3.1-8B-Instruct model against Llama-3.1-8B-UltraMedical (Zhang et al., 2024a), a domain-adapted variant explicitly fine-tuned on medical text. Moving beyond behavioral evaluation, we adopt a mechanistic approach introduced by Chang et al. (2024) to decompose the model's outputs into individual contributions of its internal components. By evaluating individual components against our benchmark tasks, we trace how parametric knowledge of jargon is encoded within the model.

To preview our results, we find that the medically fine-tuned UltraMedical model shows inferior performance compared to the general-purpose Llama-3.1 model on both benchmarks. Specifically, our analyses show that domain-specific fine-tuning does not significantly alter behaviors of individual components within the model. Instead, fine-tuning reweights jargon-sensitive components within the model that often results in miscalibration (Guo et al., 2017; Jiang et al., 2021).

We make the following contributions in this paper: 1) Two complementary benchmark tasks for evaluating jargon comprehension for LLMs; 2) A comparative evaluation between general-purpose and domain-specific fine-tuned LLMs on their abilities to process jargon across medical and materials science domains; and 3) Mechanistic interpretation of LLMs against jargon-related tasks to better understand how LLMs encode parametric knowledge about domain-specific terms.

2 Related Work

2.1 Processing jargon with LLMs.

As LLMs are increasingly deployed in high-stakes environments such as healthcare and scientific research, their ability to process specialized vocabulary has become a key capability. Domain-specific fine-tuning has emerged as the most common approach for improving this capability (Chen et al., 2023; Wu et al., 2024; Zhang et al., 2024a). Existing research on jargon in LLMs has largely concentrated on two complementary tasks: jargon detection (Rakedzon et al., 2017; Lao et al., 2021; Ke et al., 2022; Kwon et al., 2022; Guo et al., 2024) and jargon definition generation (Huang et al., 2022; August et al., 2022; Liu et al., 2021). Beyond these two dominant tasks, Lucy et al. (2023) applies word sense induction to quantify scholarly jargon and Papandreou et al. (2025) investigate jargon-aware prompting strategies for LLM-based medical text simplification.

These efforts have substantially advanced understanding of how LLMs detect jargon and generate accessible explanations, yet little is known about how internal mechanisms of LLM give rise to the parametric knowledge used to process jargon. Instead of designing tasks with the goal of improving downstream accuracy, our proposed JU and JI tasks provide a simplified setup for testing both contextual awareness and semantic understanding that better suits mechanistic analysis of LLMs. Such analyses allow us to trace parametric knowledge of jargon within internal components of an LLM, giving insights into how internal mechanisms give rise to LLMs' abilities in processing jargon.

2.2 Mechanistic interpretability of LLMs.

Mechanistic interpretability tools have been developed to complement behavioral evaluation for LLMs. For example, Meng et al. (2022) use causal tracing to demonstrate that factual associations are localized within specific mid-layer feed-forward modules. Hewitt and Liang (2019) introduce control tasks associating word types with random outputs to establish probe selectivity. Recent work by Marinescu et al. (2025) utilizes activation patching and layer lesioning to construct internal knowledge maps of medical LLMs, demonstrating that specialized concepts localize in distinct network layers and highlighting the need to apply interpretability techniques directly to domain-specific vocabulary.

In this paper, we adopt the interpretation framework proposed by Chang et al. (2024). This method provides an efficient tool for decomposing the output of an LLM into direct contributions of its internal components — Specifically, the Multi-Layer Perceptron (MLP) blocks and individual attention

[1]Code and data will be released upon publication.

heads. We pair the interpretation framework with our jargon benchmark tasks to gain mechanistic insights into how domain-specific finetuning affects processing of jargons in LLMs.

3 Jargon evaluation benchmark

3.1 Data sources

We create our benchmark using README (Yao et al., 2024), a large dataset of medical jargon entries. To complement our analysis, we also use MatScholar (Song et al., 2023), a materials science jargon dataset, to construct a parallel jargon identification task for cross-domain analysis.

README. We use `README-exp_good`, a 113,659 entry subset of README retrieved from the Unified Medical Language System (UMLS; Lindberg et al., 1993) that has passed Yao et al. (2024)'s internal quality check. Each entry consists of four fields: 1) a medical jargon drawn from an EHR note, 2) the surrounding EHR context, 3) a jargon definition retrieved from UMLS, and 4) a patient-accessible lay definition. Our pipeline uses only the term, the EHR context, and the jargon definition. See Appendix C for further details on how the dataset is used.

To ensure the benchmark contains genuinely specialized terminology, we further filter `README-exp_good` using Sentence-BERT (SBERT; Reimers and Gurevych, 2019) and Wiktionary (Ylonen, 2022). Each definition entry in Wiktionary contains tags that indicate the topic or usage context (e.g., medicine, colloquial). We use these tags to check whether a term's medical meaning is too close to its common meaning. If a README entry's jargon definition is sufficiently similar to one of the non-medical Wiktionary definition entry of the corresponding lexical item, we consider the entry a non-jargon entry. See Appendix D for more details. After de-duplication and filtering, we obtain a jargon dataset $\mathcal{J}$ with 5977 entries and a non-jargon dataset $\mathcal{N}$ with 2095 entries. We partition $\mathcal{J}$ into train, test splits, which are used to construct the downstream benchmark tasks described below. Full data statistics are provided in Appendix E.

MatScholar. MatScholar is a named entity recognition (NER) corpus of materials science paper abstracts. Each entry consists of a tokenized sentence paired with a sequence of BIO tags. Because MatScholar provides only span annotations but no dictionary definitions, we cannot construct a JU benchmark using this dataset. We therefore only use MatScholar to construct a JI benchmark for the purpose of testing cross-domain generalization of model components.

3.2 Jargon Understanding (JU) task

We first develop a jargon understanding (JU) benchmark to assess models' semantic comprehension of jargon terms. We set up the task as multiple-choice question (MCQ) prompts, a prevailing approach for evaluating LLMs (Hendrycks et al., 2020; Zhang et al., 2024b) that restricts the output space to a predefined set of options and thus avoids the scoring complexity of free-form generation. For every MCQ, the model is presented with a specific jargon term from $\mathcal{J}$ alongside an EHR note as context. The model must then select the correct definition from five candidate options. An example prompt template for JU can be found in Appendix F.

To create distractors, we rank all non-target definitions in the training pool by SBERT cosine similarity to the target definition d_w in descending order. To prevent near-duplicates of the target from being selected as distractors, we exclude any candidate whose similarity to d_w exceeds an upper threshold $\tau_{\max} = 0.9$. The distractor set for jargon term w is then defined as the top four ranked candidates that fall below this threshold. The target and the four distractors are then assigned to labels A-E, with the correct position distributed uniformly across items to prevent position bias. We construct the JU benchmark for both Train and Test splits. Distractors for both splits are drawn exclusively from the Train-set definitions to prevent test-time leakage. In total, we construct 4,183 training and 1,495 testing JU examples.

3.3 Jargon Identification (JI) task

We also construct a jargon identification (JI) benchmark to evaluate the models' abilities to situate jargon usages in linguistic contexts. We set up the task as Binary question answering (Yes/No), a standard paradigm for evaluating factual knowledge in LLMs (Hendrycks et al., 2020; Kamalloo et al., 2023). Specifically, given a jargon term and its surrounding context, the model must determine whether the term functions as a jargon in that context. We prompt the model to define jargon as terminology that a layperson is unlikely to understand. The evaluation dataset is constructed using a balanced binary sampling strategy. Positive instances are sampled from the jargon set $\mathcal{J}$. Negative in-

stances are sampled from a non-jargon set $\mathcal{N}$. An example prompt of the JI task can be found in Appendix G. We allocate 1,495 negatives to the test split to maintain a 1:1 class ratio, since primary evaluation occurs on the Test set. The remaining 600 negatives are allocated to the train split for component reweighting, along with 600 randomly sampled positives from $\mathcal{J}$. In total, the JI benchmark contains 1,200 training examples and 2,990 test examples.

3.4 Cross-domain benchmark construction

To test whether jargon-relevant components generalize beyond the medical domain, we construct a similar JI benchmark using MatScholar. Tokens or token spans tagged B- or I- are taken as the positive class. For the negative class, we use LLM-as-a-judge procedure to avoid yielding trivial negatives. We score O-tagged tokens on a 1–5 scale of perceived technical complexity, where 1 corresponds to basic English (e.g., the, however) and 5 to deceptive distractors that resemble specialized vocabulary but are not materials science entities (e.g., stochastic, orthogonal). To minimize label noise that could arise from sampling exclusively at the upper end of the scale, we draw negatives uniformly from scores 3–5. The full rubric and prompt are provided in Appendix H. We also use BoolQ (Clark et al., 2019), a reading-comprehension benchmark with yes/no questions, as a non-jargon control task. Solving BoolQ requires general language understanding rather than recognition of specialized terminology. We reformat BoolQ to match JI's A/B prompt structure.

4 Methodology

Our mechanistic analyses rely on the component decomposition and reweighting framework proposed by Chang et al. (2024), which expresses an LLM's output logits as a linear sum of contributions from individual attention heads and MLP blocks. Concretely, for a transformer with L layers, the final hidden state is the sum of the per-layer attention and MLP outputs that each component writes to the residual stream. The output logits can be rewritten as

$$\text{logits} = \sum_{j}^{N} w_j (U_Y \cdot C_j) \qquad (1)$$

Where C_j is the post-layernorm activation of component j, U_Y is the output embedding matrix restricted to a task's label words, and w_j are the weights assigned to each component (1 by default). The term $U_Y \cdot C_j$ is interpreted as component j's direct contribution to the output logits, enabling per-component accuracy measurements and per-component reweighting. We note that this captures only the direct contribution of each component to the output, not the indirect influence it exerts by modifying the residual stream read by subsequent layers. For both JU and JI, we score each choice option by reading the model's logit at the final input position over the option-letter tokens (A–E for JU, A/B for JI) and selecting the argmax.

Under this framework, each component in the model can be reweighted by adjusting the learnable weight parameters w_j. We follow Chang et al. (2024) and optimize these weights on a training set by minimizing cross-entropy loss with L1 regularization. All other model parameters remain frozen throughout the reweighting process. Because the decomposition is linear, reweighting can be applied directly without retraining the model, making the procedure efficient at scale.

Model	JU	JI
Llama-3.1-8B-Instruct	**76.1**	**58.8**
Llama-3.1-8B-UltraMedical	74.1	53.1

Table 1: Prediction accuracy on both jargon evaluation tasks. Bold indicates the best score.

5 Results

Unless otherwise noted, all results in this section are reported on the held-out test splits described in Section 3 (1,495 items for JU; 2,990 items for JI).

5.1 Comparative analysis

We first establish whether medical fine-tuning produces the behavioral improvement on jargon tasks that domain adaptation is typically assumed to deliver. Table 1 shows the benchmark task performance of both the general-purpose Llama-3.1-8B-Instruct model and the domain-adapted Llama-3.1-8B-UltraMedical model. Across both tasks, the base Llama-3.1 Instruct model outperforms the domain-adapted UltraMedical model, with the gap being most pronounced on JI. This counterintuitive result motivates the analyses that follow: We examine model calibration on both tasks and error structure on JI to characterize how medical fine-tuning shapes model behavior.

Calibration analysis. A well-calibrated model assigns confidence scores that match its empirical

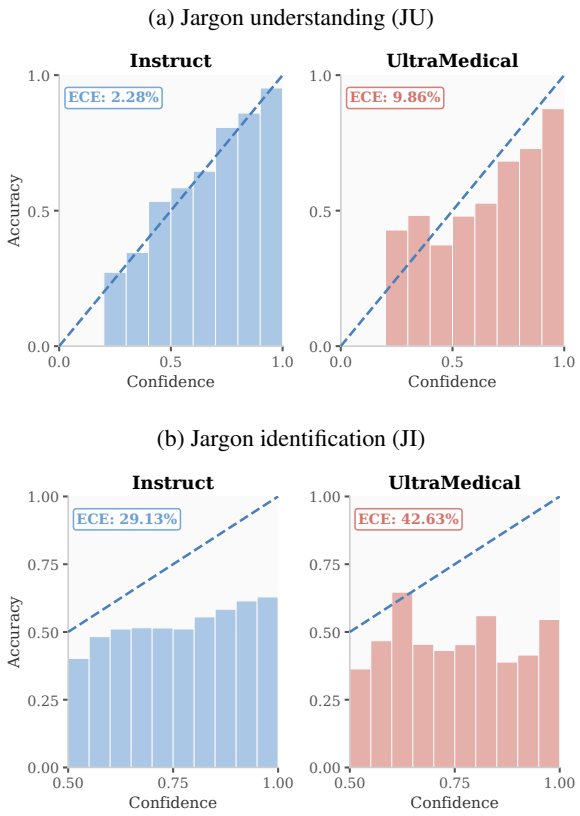

Figure 2: Model calibration on both tasks. The dashed diagonal lines represent perfect calibration.

accuracy: Predictions made with 80% confidence should be correct 80% of the time (Guo et al., 2017; Jiang et al., 2021). Miscalibration is particularly consequential in clinical settings, where downstream decisions depend not only on model outputs but on the reliability of the confidence those outputs carry. We measure calibration using Expected Calibration Error (ECE; Naeini et al., 2015): The weighted average gap between confidence and accuracy across binned predictions. Results in Figure 2 shows that, on both tasks, the two models differ sharply in calibration quality. The Instruct model tracks the diagonal more closely, while UltraMedical's accuracy is consistently lower in higher-confidence bins, which is the standard signature of overconfidence. In a clinical setting, where downstream decisions depend on the reliability of model confidence, this form of miscalibration is a concrete cost of domain adaptation: The model's output becomes harder to trust precisely in the high-confidence region where users would be more inclined to believe the model.

Error analysis. We perform a qualitative error analysis on the JI results to characterize common patterns in the models' errors. We focus on disagreement cases: Items where one model answers correctly and the other does not. Table 2 shows clear distinction between the models' errors. UltraMedical's errors are almost entirely false positives (376 of 377 disagreement cases), while Instruct's are almost entirely false negatives (196 of 197). Medical fine-tuning tends to shift the decision boundary toward over-classification: Terms appearing in clinical contexts are systematically treated as specialized. The Instruct model exhibits bias in the reverse direction, occasionally missing context-dependent polysemy in which everyday words take on specialized clinical meanings. Full examples from each category of errors are provided in Appendix I and J.

	Instruct	UltraMedical
Total Errors	197	377
False Positives	1	376
False Negatives	196	1
Mean Confidence Margin	1.64	3.04

Table 2: JI Disagreement Error Summary. Each column counts cases where the named model is wrong while the other model is correct. Absolute Mean Confidence Margin is calculated by averaging the Margin $= |\text{logit}(A) - \text{logit}(B)|$, where A = Jargon and B = Non-Jargon.

5.2 Localization of domain-specific knowledge

The behavioral analysis establishes that medical fine-tuning shifts model behavior toward overconfidence. We next investigate whether this shift reflects a reorganization of parametric knowledge encoded within the model. To examine this, we apply the method described in Section 4 to decompose each model's output into per-component contributions and identify components that carry jargon-relevant signal in each model.

Figure 3 shows per-component accuracy for both models on both tasks. For JU, high-performing components are concentrated in the upper layers (17–31) in both Instruct and UltraMedical, with the lower layers contributing near-zero individual accuracy. A small set of components stands out in both models — most notably L17H24, L26H3, MLP-29, and L31H1, which are the top four components for both Instruct and UltraMedical (differing only in rank order). The JI heatmaps exhibit a markedly more sparse distribution, indicating that contextual understanding of jargon is more spread-out within the models. We provide the full lists of top component in Appendix K.

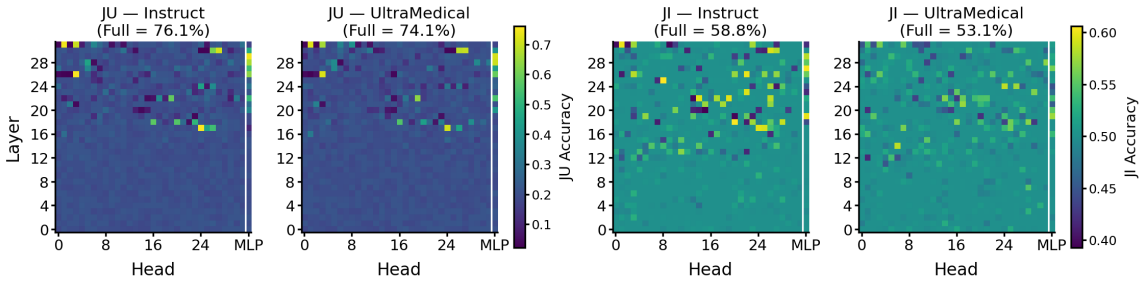

Figure 3: Per-component accuracy for Llama-3.1-8B Instruct and UltraMedical on JU (left) and JI (right).

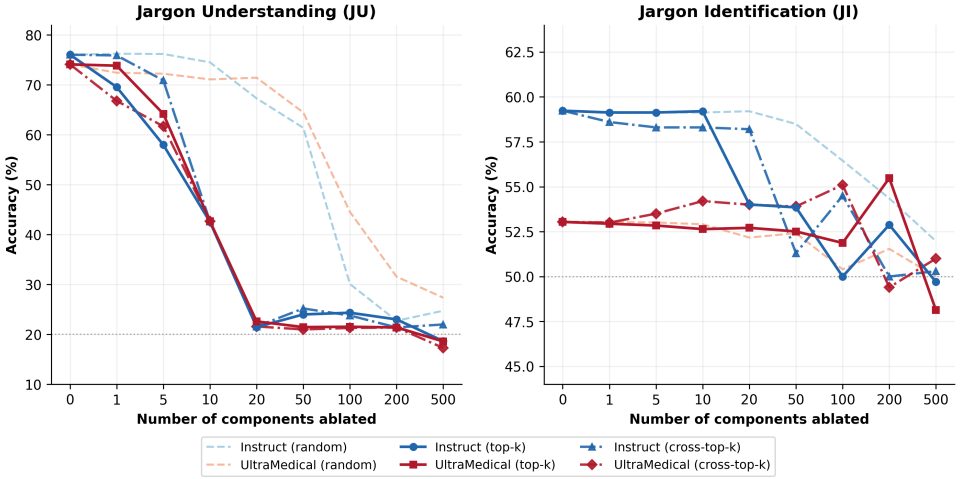

Figure 4: Cumulative ablation results on JU (left) and JI (right). Horizontal gray lines mark chance performance (20% on JU, 50% on JI).

Ranked correlations. To quantify how similarly the two models rank their components, we compute weighted Kendall's τ (Vigna, 2015) between the per-component accuracy of Instruct and UltraMedical. Unlike standard Kendall's τ, which weights all rank disagreements equally, the weighted variant places greater weight on disagreements near the top of the ranking, making it well-suited to our setting where the highest-accuracy components are of primary interest. See Appendix L for a full specification of the metric. On JU, the two models agree closely at the top (τ = 0.72). On JI, agreement is slightly weaker (τ = 0.59) but still substantial. Together, these patterns suggest that domain adaptation does not reorganize where jargon knowledge resides. The architectural locus is largely shared between the base and fine-tuned models, with fine-tuning modulating component contributions rather than relocating them.

Ablation studies To verify that the components identified above are causally responsible for task performance, we perform a cumulative ablation study in which the top-k components, ranked by individual accuracy, are removed simultaneously from the model. Following Michel et al. (2019), we ablate attention heads and MLP blocks by zeroing their outputs before they are written to the residual stream, removing their direct contributions while leaving all learned weights intact (see Appendix M for full details). Figure 4 shows the results. On JU, both models' accuracy decrease substantially quicker when ablating the top-k components, reaching chance performance by $k = 20$, whereas performance erodes more gradually when random components are being ablated. On JI, by contrast, both targeted and random ablation produce only shallow declines that track one another closely. The contrast between JU and JI ablation profiles suggests that JU relies on a small set of specialized components whereas JI relies on aggregate contributions from multiple components. We also cross-ablate each model with top-k components identified by the other model variant. Under both tasks, we find that cross-ablating top-k components closely tracks the conditions where components identified by the corresponding models are ablated, giving further evidence that domain-specific finetuning simply modulates the encoded knowledge instead of reor-

	JU		JI	
	Baseline	Reweighted	Baseline	Reweighted
Instruct	76.1	76.9	58.8	57.8
UltraMedical	74.1	74.2	53.1	56.1

Table 3: Baseline and reweighted accuracy on JU and JI for both models.

Question	Orig. M.	Rew. M.
PAST MEDICAL HISTORY: Significant for B-cell lymphoma ... history of methadone use, hypertension, anxiety, deconditioning, nicotine **addiction**, depression, ...	2.01	0.0
Assessment of Current Medications: medication administration orders for isosorbide mononitrate ... Er **tablet**, By Mouth, BID, Routine, **DATE[Oct 21 07] 10:55:00**; heparin 25000 Unit(s) 250 mL ...	1.98	0.0

Table 4: Individual logit contributions of L30H27 and full-model margins for two non-jargon terms incorrectly classified as jargon by the UltraMedical model. Orig. M. and Rew. M. denote the full-model logit margin (jargon − non-jargon) before and after reweighting.

ganizing it. Interestingly, when cross-ablating the UltraMedical model, performance on JI increases notably. We postulate that the fine-tuned UltraMedical model relies on components sensitive to jargon semantics to appropriate jargon terms as conventional terms.

5.3 Component reweighting

So far, our results suggest that domain-specific finetuning simply modulates existing jargon-sensitive components within the model. We now investigate whether the modulation learned by domain-specific finetuning is optimal w.r.t the downstream tasks. We follow the procedure described in Section 4 to learn component weights. See Appendix O for detailed training procedure.

Results. Table 3 shows the results before and after reweighting. Reweighting yields a 3.0% accuracy gain on JI for UltraMedical. For Instruct on JI, the same procedure produces a 1.0% decrease in accuracy. Gains on JU are marginal for both models ($\leq$0.2%). Our results show that reweighting the finetuned model on JI results in a substantial performance improvement (3.0%), indicating that fine-tuning does not necessarily learn modulations of model components that are beneficial toward domain-specific downstream tasks.

Error analysis. MLP-29 and L30H27 receive the largest weight reductions in both models, but the suppression is qualitatively different. In UltraMedical, the optimization effectively eliminates both components from the JI prediction: L30H27 receives a weight of approximately 0 and MLP-29 a weight of 10^{-4}. In Instruct, the same components are only partially suppressed, with weights of 0.39 and 0.27 respectively. Notably, these are also among the highest-accuracy components on JU (Section 5.2; Appendix K).

By examining the JI error structure, the UltraMedical gain comes from correcting 270 false positives at the cost of introducing 146 new false negatives, a net shift that directly counteracts the over-confidence bias documented in Section 5.1. The components being downweighted most aggressively are the same ones whose individual contributions most strongly favor the jargon label. To illustrate how this suppression affects individual predictions, Table 4 shows two non-jargon terms that UltraMedical originally misclassified as jargon. For both *addiction* and *tablet*, the baseline full-model margin favors the jargon label by roughly +2.0. After reweighting, L30H27's weight is reduced to 0, zeroing out its contributions and flipping both predictions to the correct non-jargon label. The same component that drives accurate definition selection on JU is also the primary source of over-classification on JI, and the reweighting procedure recovers JI accuracy by selectively dampening its contribution at inference time.

For Instruct, the absence of a directional bias means the optimizer has no systematic error pattern to correct. Instruct's baseline JI errors are roughly balanced between false positives and false negatives, and applying the same suppression to MLP-29 and L30H27 produces a corresponding shift toward false negatives without a compensating reduction in false positives. The 1.0% degradation is the expected consequence: The intervention removes signal that was contributing to balanced predictions, with no bias for the suppression to cancel out.

This asymmetry suggests that medical finetuning amplifies the jargon-favoring tendencies of specific pre-existing components, and that the reweighting gain stems from suppressing this amplified bias rather than from a generic recalibration of the model. Instruct, on the other hand, loses performance due to the suppression of components that were not over-amplified to begin with.

Component	Med-JI	Mat-JI	BoolQ
L20H10	0.562	0.769	0.505
L18H26	0.553	0.735	0.539
L29H16	0.532	0.673	0.533
L26H4	0.537	0.607	0.448
L19H5	0.571	0.605	0.349
L15H11	0.552	0.619	0.531
L26H27	0.570	0.590	0.510
MLP-19	0.598	0.595	0.532
L21H18	0.578	0.589	0.501
Full Accuracy	0.591	0.739	0.816

Table 5: Components in the top-50 of both Med-JI and Mat-JI but outside the top-50 of BoolQ.

5.4 Cross-domain generalization

Section 5.2 identifies a small set of components that disproportionately contribute to jargon comprehension on the medical jargon benchmarks. A natural follow-up question is whether these components encode a domain-agnostic notion of specialized terminology, or whether they are specific to the medical domain. To investigate this, we construct a parallel jargon identification benchmark in the materials science domain (Mat-JI) and compare per-component accuracy across the two domains. We additionally include BoolQ (Clark et al., 2019), a binary reading-comprehension task reformatted into the same A/B prompt structure, as a non-jargon control to rule out the possibility that the top components reflect general A/B classification competence rather than jargon comprehension. Details about the construction of Mat-JI are provided in Appendix H.

We apply the component decomposition procedure from Section 4 to compute per-component accuracy on Medical Jargon Identification (Med-JI) and Mat-JI for Llama-3.1-8B-Instruct, then filter for components that rank in the top-50 of both jargon tasks but fall outside the top-50 of BoolQ. This filter isolates 9 components whose strong jargon performance is not explained by general task competence. Table 5 lists these components.

Two components stand out as the strongest candidates for domain-agnostic jargon knowledge. L20H10 and L18H26 achieve near full model accuracy on Mat-JI. Meanwhile, their Med-JI accuracies likewise rank among the highest individual contributions while performing near chance at BoolQ. The remaining seven components in Table 5 satisfy the filter criterion but exhibit weaker jargon signal. These components are concentrated in the upper-middle layers rather than the final layers that dominate the unfiltered rankings on Med-JI alone, suggesting that jargon recognition may operate on top of a separate, later-stage answer-formatting circuit. Together, these findings indicate that a small subset of upper-middle-layer components carries jargon-relevant knowledge that transfers across two unrelated technical domains, supporting the interpretation that domain-agnostic knowledge about jargon is also encoded within these models.

This partial domain-agnosticism has an important practical implication: Interventions developed on medical jargon, such as the reweighting procedure in Section 5.3, may transfer to other technical domains without the need to perform per-domain retraining. The fact that this signal is present in the general-purpose model and transfers across two unrelated technical domains suggests that jargon identification is not a capability acquired through domain-specific fine-tuning, but rather a partially domain-agnostic structure already encoded during pretraining.

6 Conclusion

We have presented a mechanistic comparison of jargon comprehension in general-purpose and medically fine-tuned LLMs. Using two novel jargon comprehension benchmarks designed for mechanistic interpretation, we evaluate the Llama-3.1-8B family models on the medical domain as a case study. Our results suggest that while domain-specific fine-tuning is widely assumed to improve performance on specialized text, it can in fact degrade comprehension by being overconfident in domain-specific lexical items. Component-level analysis reveals that this bias does not stem from a reorganization of internal knowledge, but can instead be attributed to placing excessive emphasis on specific jargon-sensitive components, resulting in a re-modulation of existing knowledge that is not optimal for improving performance on related downstream tasks. We also observe evidence that pre-trained models possess domain-agnostic knowledge of jargon, again questioning the necessity for domain-specific fine-tuning. Overall, our results underscore the limitations of domain-specific fine-tuning for improving model performance on downstream tasks involving specialized terminology and emphasize the need for more robust methodologies for processing domain-specific language.

Limitations

We use a SBERT-based filtering pipeline to construct and filter our benchmarks. We acknowledge that automatic methods may not always yield the optimal results but opted to use such a pipeline to enable scale in our evaluation benchmark. While constructing the dataset, we only retain terms whose medical definitions are sufficiently distant from their conventional meanings as jargon, thus prioritizing precision on the positive class. This is appropriate given that our central claims concern model behavior on confidently specialized terminology. However, a tradeoff does exist and it would be interesting to extending the benchmark by having human annotations with annotator confidence scores that would enable more finer-grained analysis of more ambiguous cases of jargon usages.

Due to computational resource constraints, our analysis focuses on the Llama-3.1-8B family alone, comparing one general-purpose model against one medically fine-tuned variant. While such a controlled pairing isolates the effect of fine-tuning while holding architecture and scale fixed, it would be interesting to test whether our findings extend to other model families and model scales. We also acknowledge that we have only examined English jargon usages and leave multilingual extensions to future work.

Ethics Statement

README is distributed under CC-BY-NC 4.0 license for research purposes. MatScholar is released under the MIT License, and we use BoolQ under CC-BY-SA-3.0 license. README's EHR contexts were anonymized by the original dataset authors as part of their release. We acknowledge the potential risk for personal information leakage when using medical domain data, but we perform no further identification of individuals to ensure that no additional risk is introduced. We also did not collect new data from human subjects. README is distributed for research purposes; our derived benchmarks inherit the same access conditions and are intended only for research on model behavior, not for clinical deployment.

We used AI assistants to expedite the coding process. All code snippets produced by AI assistants were verified by the authors before being used. For writing, we only used AI assistants for light copy-editing and phrasing suggestions.

References

Tal August, Katharina Reinecke, and Noah A Smith. 2022. Generating scientific definitions with controllable complexity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8298–8317.

Olivia M Bullock, Daniel Colón Amill, Hillary C Shulman, and Graham N Dixon. 2019. Jargon as a barrier to effective science communication: Evidence from metacognition. *Public Understanding of Science*, 28(7):845–853.

M Teresa Cabré Castellví. 1999. *Terminology: Theory, methods and applications*, volume 1. John Benjamins Publishing.

Gina N Cervetti, Elfrieda H Hiebert, P David Pearson, and Nicola A McClung. 2015. Factors that influence the difficulty of science words. *Journal of Literacy Research*, 47(2):153–185.

Ting-Yun Chang, Jesse Thomason, and Robin Jia. 2024. When parts are greater than sums: Individual llm components can outperform full models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10280–10299.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, and 1 others. 2023. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 2924–2936.

Emma Croxford, Yanjun Gao, Elliot First, Nicholas Pellegrino, Miranda Schnier, John Caskey, Madeline Oguss, Graham Wills, Guanhua Chen, Dmitriy Dligach, and 1 others. 2025. Evaluating clinical ai summaries with large language models as judges. *npj Digital Medicine*, 8(1):640.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Yue Guo, Joseph Chee Chang, Maria Antoniak, Erin Bransom, Trevor Cohen, Lucy Lu Wang, and Tal August. 2024. Personalized jargon identification for

enhanced interdisciplinary communication. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4535–4550.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pages 2733–2743.

Jie Huang, Hanyin Shao, Kevin Chen-Chuan Chang, Jinjun Xiong, and Wen-mei Hwu. 2022. Understanding jargon: Combining extraction and generation for definition modeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3994–4004.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 5591–5606.

Liang Ke, Xinyu Chen, and Haizhou Wang. 2022. An unsupervised detection framework for chinese jargons in the darknet. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 458–466.

Sunjae Kwon, Zonghai Yao, Harmon Jordan, David Levy, Brian Corner, and Hong Yu. 2022. Medjex: A medical jargon extraction model with wiki's hyperlink span and contextualized masked language model score. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11733–11751.

Yingying Lao, Chenghuan Zhang, Yilun Wei, and Dongli Han. 2021. Detecting and finding the true meaning of jargons. In *Proceedings of the 7th International Conference on Frontiers of Educational Technologies*, pages 45–50.

Donald AB Lindberg, Betsy L Humphreys, and Alexa T McCray. 1993. The unified medical language system. *Yearbook of medical informatics*, 2(01):41–51.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, and 1 others. 2025. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *ACM Computing Surveys*, 58(3):1–39.

Zequn Liu, Shukai Wang, Yiyang Gu, Ruiyi Zhang, Ming Zhang, and Sheng Wang. 2021. Graphine: A dataset for graph-aware terminology definition generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3453–3463.

Li Lucy, Jesse Dodge, David Bamman, and Katherine Keith. 2023. Words as gatekeepers: Measuring discipline-specific terms and meanings in scholarly publications. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6929–6947.

Razvan Marinescu, Victoria-Elisabeth Gruber, and Diego Fajardo. 2025. Medical interpretability and knowledge maps of large language models. *arXiv preprint arXiv:2510.11390*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Taiki Papandreou, Jan Bakker, and Jaap Kamps. 2025. Medical text simplification from jargon detection to jargon-aware prompting. In *Proceedings of the Fourth Workshop on Text Simplification, Accessibility and Readability (TSAR 2025)*, pages 36–46.

Tzipora Rakedzon, Elad Segev, Noam Chapnik, Roy Yosef, and Ayelet Baram-Tsabari. 2017. Automatic jargon identifier for scientists engaging with the public and science communication educators. *PloS one*, 12(8):e0181742.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ren Ryba, Zoë A Doubleday, Matthew J Dry, Carolyn Semmler, and Sean D Connell. 2021. Better writing in scientific publications builds reader confidence and understanding. *Frontiers in psychology*, 12:714321.

Yu Song, Santiago Miret, and Bang Liu. 2023. Matscinlp: Evaluating scientific language models on materials science language tasks using text-to-schema modeling. *arXiv preprint arXiv:2305.08264*.

Sebastiano Vigna. 2015. A weighted correlation index for rankings with ties. In *Proceedings of the 24th international conference on World Wide Web*, pages 1166–1176.

Daril A Vilhena, Jacob G Foster, Martin Rosvall, Jevin D West, James Evans, and Carl T Bergstrom. 2014. Finding cultural holes: How structure and culture diverge in networks of scholarly communication. *Sociological Science*, 1:221–23.

Bailin Wang, Zi Wang, Xuezhi Wang, Yuan Cao, Rif A Saurous, and Yoon Kim. 2023. Grammar prompting for domain-specific language generation with large language models. *Advances in Neural Information Processing Systems*, 36:65030–65055.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. 2024. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, 31(9):1833–1843.

Fangkai Yang, Pu Zhao, Zezhong Wang, Lu Wang, Bo Qiao, Jue Zhang, Mohit Garg, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2023. Empower large language model to perform better on industrial domain-specific question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 294–312.

Zonghai Yao, Nandyala Siddharth Kantu, Guanghao Wei, Hieu Tran, Zhangqi Duan, Sunjae Kwon, Zhichao Yang, and Hong Yu. 2024. README: Bridging medical jargon and lay understanding for patient education through data-centric NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12609–12629, Miami, Florida, USA. Association for Computational Linguistics.

Tatu Ylonen. 2022. Wiktextract: Wiktionary as machine-readable structured data. In *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pages 1317–1325, Marseille, France.

Kaiyan Zhang, Sihang Zeng, Ermo Hua, Ning Ding, Zhang-Ren Chen, Zhiyuan Ma, Haoxin Li, Ganqu Cui, Biqing Qi, Xuekai Zhu, and 1 others. 2024a. Ultramedical: Building specialized generalists in biomedicine. *Advances in Neural Information Processing Systems*, 37:26045–26081.

Ziyin Zhang, Zhaokun Jiang, Lizhen Xu, Hongkun Hao, and Rui Wang. 2024b. Multiple-choice questions are efficient and robust llm evaluators. *arXiv preprint arXiv:2405.11966*.

A Computational setup

All SBERT-based components of our pipeline — the Wiktionary-based jargon filtering (Appendix D), the within-term semantic deduplication (Appendix C), and the distractor ranking for the JU benchmark (Section 3.2) — use the all-mpnet-base-v2 checkpoint from the sentence-transformers library. Embeddings are compared using cosine similarity. SBERT encoding runs on a single NVIDIA RTX A6000 GPU and takes under 5 minutes for the full README-exp-good filtering and JU distractor ranking.

All experiments use Llama-3.1-8B-Instruct and Llama-3.1-8B-UltraMedical. Both models are used in inference-only settings: The decomposition procedure caches per-component projections to disk, and the reweighting procedure optimizes 1056 scalar weights against these cached projections on CPU, without further forward passes through the 8B models. Decomposition runs on two NVIDIA RTX A6000 GPUs (48GB each) and take approximately 5 minutes for JU and 8 minutes for JI per model. Cumulative ablation experiments (top-k, random, and cross-model variants across both tasks and both models) total approximately 2 hours of GPU time. Reweighting optimization on cached projections takes under one minute per task on CPU.

B Decomposition setup

Our JU and JI tasks are zero-shot evaluations with a single canonical prompt template per task. We use the publicly released code from Chang et al. (2024), with the following arguments held constant across all experiments:

- `n_shots`: 0 (zero-shot evaluation)
- `format`: 1 (single canonical prompt template per task)
- `seed_list`: [0] (single seed; see discussion below)

In our zero-shot setting, the `seed_list` argument passed to the decomposition and evaluation scripts does not affect results, the forward pass is deterministic and no demonstration sampling occurs at evaluation time. We use the default value for all other hyperparameters.

Dataset	Data Description	Size
Medical Domain		
README-exp-good	Unfiltered medical jargon data	113,659
Jargon Dataset $\mathcal{J}$	SBERT filtered jargon	5,977
Non-Jargon Dataset $\mathcal{N}$	SBERT filtered non-jargon	2,095
Train data	70% split from $\mathcal{J}$	4,183
Test data	25% split from $\mathcal{J}$	1,495
Dev data	5% split from $\mathcal{J}$	299
Material Science Domain		
Mat-JI	Material Science Jargon Identification task	2,990
Others		
BoolQ	Control group to isolate out non-jargon components	2,990

Table 6: Data statistics for the medical-domain jargon benchmark. Rows show the unfiltered README-exp-good source, the SBERT-filtered jargon ($\mathcal{J}$) and non-jargon ($\mathcal{N}$) sets, and the Train/Test/Dev splits derived from $\mathcal{J}$.

C Details of README

README (Yao et al., 2024) provides multiple subsets that differ in how their general definitions are sourced. The README-exp variants contain general definitions retrieved from UMLS, an authoritative biomedical vocabulary maintained by the National Library of Medicine. The README-syn variants, in contrast, contain general definitions generated by GPT-3.5-turbo through README's data augmentation pipeline. Within the expert-annotated branch, `README-exp_good` is the subset whose UMLS-retrieved general definitions passed README's Examiner stage, in which an LLM-based quality check filters out definitions deemed inaccurate, contextually misaligned, or otherwise unsuitable for the corresponding lay definition.

We only use `README-exp_good` as our jargon source. Restricting our benchmark to UMLS-retrieved definitions ensures that every jargon term in our evaluation is associated with a definition from an authoritative medical knowledge base rather than from a language model.

D Wiktionary-based definition filtering

Section 3.1 describes how Wiktionary tags are used to identify the conventional (non-medical) senses of each candidate jargon term. Let $\mathcal{T}_{\text{tech}}$ denote the set of Wiktionary tags that indicate specialized technical usage. For each term w in `README-exp_good`, we retrieve its set of Wiktionary definitions D_{wik}. We define the subset of conventional (non-medical) definitions as:

$$D_{conv} = \{d \in D_{wik} \mid \text{tags}(d) \cap \mathcal{T}_{\text{tech}} = \emptyset\} \quad (2)$$

We retain w as jargon if its README medical definition d_w is sufficiently distant from all conventional senses:

$$\mathcal{J} = \{w \in \mathcal{T} : \phi(w) < \tau\},$$
$$\text{where } \phi(w) = \max_{d \in D_{\text{conv}}} \text{sim}(f_\theta(d_w), f_\theta(d)) \quad (3)$$

Where f_θ represents the SBERT embedding, $\text{sim}(\cdot)$ is the cosine similarity function, and $\tau = 0.5$ chosen by manual inspection. Terms whose medical definition differs sufficiently from their non-medical senses (similarity $< \tau$) are retained as jargon. Terms that fail this criterion are reassigned to the non-jargon set.

The tag list $\mathcal{T}_{\text{tech}}$ is constructed by inspecting the Wiktionary tags that appeared most frequently across the candidate terms in `README-exp_good` and manually selecting those that indicate medical or technical usage.

The set $\mathcal{T}_{\text{tech}}$ comprises the following tags: jargon, slang, medicine, pathology, pharmacology, anatomy, surgery, immunology, physiology, biology, microbiology, biochemistry, genetics, zoology, botany, entomology, psychology.

E Data Statistics

Table 6 shows the data description and the data size of each of the benchmarks. The unfiltered `README-exp-good` source contains 113,435 entries.

We first normalize both the jargon term and its general definition using regular expressions. We then drop duplicate (jargon, general definition) pairs on the normalized strings, retaining a single representative entry per unique pair. Through

System Prompt:
You are a clinical assistant. Given a medical TERM (jargon), an optional EHR context, and several candidate general definitions, select the single best general definition. Reply with ONE letter only (A, B, C, D, or E)
User Input:
TERM: aorta
EHR CONTEXT: Over the guidewire, a 5-French Sos catheter was reshaped in the descending thoracic aorta and utilized to catheterize an iliolumbar branch extending from the infrarenal abdominal aorta.
Choose the best general definition:
A) The main trunk of the systemic arteries.
B) The continuation of the subclavian artery; it distributes over the upper limb, axilla, chest and shoulder.
C) The venous trunk which returns blood from the head, neck, upper extremities and chest.
D) The venous trunk which receives blood from the lower extremities and from the pelvic and abdominal organs.
E) An arterial trunk that contains parts of the posterior tibial artery and fibular artery.
Answer with a single letter only.

Table 7: Example prompt of JU

manual inspection, we identified many jargon entries with has more than one definition, but these definitions often differ only in word order. To remove these, we apply a within-term semantic de-duplication step using SBERT. For each jargon term, we encode all of its remaining general definitions and compute pairwise cosine similarities. We then perform a greedy selection: iterating through definitions in their original order, we retain a candidate only if its cosine similarity to every already-retained definition for that term falls below a threshold of $\tau = 0.92$. The threshold is chosen manually, which results in 8,072 remaining entries. Finally, We use SBERT to perform the Wiktionary based filtering described in Appendix D.

F Example prompt for JU

Table 7 shows an example prompt of JU.

G Example prompt for JI

Table 8 shows an example prompt of Medical JI.

H Construction of Mat-JI

This section provides the full construction pipeline for Mat-JI, the materials science jargon identification benchmark used in our cross-domain analysis.

System Prompt:
You are a clinical assistant. Given a TERM from an EHR (Electronic Health Record) note and its context, determine whether the term is medical jargon — specialized medical terminology that a layperson would likely not understand. Answer only with a single letter (A or B).
User Input:
TERM: stable
EHR CONTEXT: The rest of his issues, including his hepatic encephalopathy, hypertension, and hypernutrition, are stable, and he appears to be eating adequately on his own, so we do not need to reinsert his oral gastric tube again.
Is this term a medical jargon (specialized terminology a layperson would not understand)?
A) Yes
B) No
Answer with A or B only.

Table 8: Example prompt of Medical JI

MatScholar is a named entity recognition (NER) corpus consisting of materials science paper abstracts annotated with BIO tags for domain entities such as materials, properties, applications, and synthesis methods. Each token is labeled B- (beginning of an entity span), I- (inside an entity span), or O (outside any entity). Because MatScholar provides only NER annotations and no dictionary definitions, the SBERT-based filtering procedure used for the medical pipeline cannot be applied here. We therefore construct the benchmark directly from the BIO structure.

Positive Class. Each contiguous span of B-/I-tagged tokens is treated as a single jargon term, since these spans correspond to recognized domain entities by construction.

Negative Class. The naïve approach would be to sample negatives uniformly from O-tagged tokens. However, this yields trivial distractors: stop words (e.g., the, and), basic academic verbs (e.g., study, show), and other surface-level non-entities that any model can trivially classify. Such negatives would inflate accuracy without testing the model's ability to distinguish specialized terminology from general scientific vocabulary. To construct a more challenging negative class, we use an LLM-as-a-judge procedure to mine O-tagged tokens that are not materials science entities but are perceived as technical.

LLM annotation framework We use Claude Sonnet 4.5 as the judge model. The judge is

Based on the following term from a materials science abstract, how would you rate its perceived technical complexity? The term should be evaluated as a non-material science term. We are looking for terms that may sound technical but are not themselves materials science entities.

Context: [Abstract]
Term: [O-tokens]

On a scale of 1 to 5, where:
1 = Basic English (e.g., *the*, *however*, *because*)
2 = Standard academic vocabulary (e.g., *study*, *analyze*, *method*)
3 = Broad scientific vocabulary (e.g., *temperature*, *velocity*, *concentration*)
4 = College-level technical vocabulary (e.g., *synthesized*, *configuration*, *empirical*)
5 = Deceptive hard negative: mimics specialized nomenclature or is jargon from another field, but is not a materials science entity (e.g., *stochastic*, *orthogonal*, *eigenvector*)

Please only respond with a number from 1 to 5.

Table 9: Prompt used for negative word annotation in Mat-JI.

prompted to act as an expert computational linguist and rate each O-tagged token on a 1–5 ordinal scale of perceived technical complexity. We prompt Claude Sonnet 4.5 using the prompt shown in Table 9.

We sample negatives uniformly across scores 3, 4, and 5 rather than from score 5 alone. This shifts the bulk of the benchmark away from boundary cases while still ensuring meaningful difficulty. The stratified negatives are combined with the positive entity spans at a 1:1 ratio to produce a balanced 2,990-item Mat-JI evaluation set, matching the size of the medical JI test set. Table 10 shows an example of Mat-JI.

I Examples of UltraMedical's errors

Table 11 shows three error examples where UltraMedical misclassified a non-jargon term as jargon, while Instruct correctly classified it as non-jargon. Manual inspection of these cases reveals two recurring categories.

The first category consists of everyday English words that appear in clinical documentation but are not themselves specialized medical terminology. Words such as *elevated*, and *consumption* are widely understood outside clinical settings, but UltraMedical systematically treats their appearance in an EHR context as evidence of jargon status. The second category consists of lay-accessible medical terms, that is, words that are technically medical in origin but have become part of everyday health literacy. Examples include *vaccination* and *hospitalizations*, which the model overestimates as specialized terminology despite being broadly familiar to non-experts.

Together, these patterns suggest that UltraMedical's decision boundary is anchored more to the clinical context in which a term appears than to the term's intrinsic semantics. Fine-tuning on medical text, where the majority of training examples involve genuine specialized terminology embedded in clinical contexts, plausibly strengthens this contextual association at the expense of term-level discrimination.

System Prompt:

You are a research assistant. Given a TERM from a material science paper abstract and its context, determine whether the term is material science jargon — specialized terminology that a reader outside the field would likely not understand. Answer only with a single letter (A or B).

User Input:

TERM: couples

ABSTRACT CONTEXT: analysis of the temperature dependence of the elastic modulus indicates that in the particular range of carrier concentration there exists a narrow electronic band in the vicinity of the fermi level which couples to the shearing strain exx - eyy .

Is this term a material science jargon (specialized terminology a non-expert would not understand)?

A) Yes

B) No

Answer with A or B only.

Assistant Output (Gold Label):

B

Table 10: Example prompt of Mat-JI.

J Examples of Instruct's errors

For completeness, we also conducted a qualitative analysis of the 196 disagreement cases where the Instruct model misclassified a jargon term as non-jargon while UltraMedical correctly identified it as jargon. Instruct's errors reflect the opposite tendency: The model interprets terms according to their conventional meanings and is insufficiently sensitive to the specialized senses they acquire in clinical contexts.

Question	Logit (Jargon)	Logit (Non-Jargon)
This is likely a small region of ischemia caused by **elevated** demand in the context of the current illness, which should resolve without incident . . .	29.38	23.5
DISCHARGE DIAGNOSIS: Chronic obstructive asthma with exacerbation, obstructive bronchitis, a history of noncompliance with medical treatment, and the necessity for prophylactic **vaccination** and inoculation against respiratory viruses.	29.75	24.0
Longitudinal incisions were **simultaneously** made over the PIP joints of 2, 3, 4, and 5, and capsulotomies of the PIP joints were performed.	30.13	20.13

Table 11: Logit scores for the jargon and non-jargon labels on two examples where the Instruct model correctly classified the bolded term as non-jargon but UltraMedical misclassified it as jargon. The logit scores shown are from the UltraMedical model.

Table 12 illustrates this pattern with two representative examples. In the first, *dressings* appears in a post-surgical context where it refers to wound coverings, but the Instruct model assigns a higher logit to the non-jargon label, defaulting to the term's everyday culinary sense. The second example shows the same pattern for *compression*, which denotes a specific therapeutic protocol for venous disease in this clinical setting rather than the general physical act of squeezing. In both cases, the logit margins are narrow, suggesting that the model recognizes some clinical signal but ultimately weights the conventional sense more heavily.

Instruct maintains a more conservative decision boundary, requiring stronger contextual signals before classifying a term as jargon. This conservatism results in a reversed error pattern compared to UltraMedical: Rather than over-extending the jargon label to terms appearing in clinical contexts, Instruct fails to flag genuinely specialized usages when the surface form of the term remains familiar.

Question	Logit (Jargon)	Logit (Non-Jargon)
Overlying surgical drain, skin staples, and **dressings** are applied. . . .	33.5	37.5
Due to the patient's element of chronic venostasis disease, approximately 1 week of **compression** therapy was initiated prior to the surgery to reduce the edema.	33.75	34.25

Table 12: Logit scores for the jargon and non-jargon labels on two examples where the Instruct model incorrectly classified the bolded term as non-jargon but UltraMedical classified it as jargon. The logit scores shown are from the Instruct model.

K Top performing components on both tasks for both models

Table 13 and 14 show the top 5 performing components of the two models on JU and JI, respectively.

On JU, the same four components — L17H24, L26H3, MLP-29, and L31H1 — appear in the top-5 of both models, differing only in rank order, with individual accuracies approaching but not exceeding full-model performance (e.g., 75.5% vs. 76.1% for Instruct). On JI, by contrast, the top-5 lists share no components between the two models. However, the individual top components exceed full-model accuracy in both cases (61.3% vs. 59.1% for Instruct; 58.6% vs. 53.1% for UltraMedical). This pattern is consistent with Chang et al. (2024)'s observation that individual components can outperform full models on tasks where the aggregate output reflects competing signals, and reinforces the finding from Section 5.2 that JI relies on distributed contributions whose ranking is sensitive to fine-tuning, whereas JU is supported by a small set of specialized components shared across models.

Rank	Instruct	Accuracy	UltraMedical	Accuracy
1	L26H3	75.52%	L31H1	72.37%
2	L17H24	75.52%	L26H3	71.84%
3	MLP-29	74.45%	MLP-29	71.37%
4	L31H1	74.38%	L17H24	70.90%
5	L30H27	71.04%	MLP-28	70.03%

Table 13: Top 5 performing components for Llama-3.1-8B-Instruct and Llama-3.1-8B-UltraMedical on JU task. L26H3 means the 3rd attention head at layer 26. MLP-29 means the MLP at layer 29.

Rank	Instruct	Accuracy	UltraMedical	Accuracy
1	L18H20	61.34%	L24H24	58.63%
2	L25H28	61.00%	L14H6	58.46%
3	MLP-27	60.64%	MLP-19	57.86%
4	L31H1	60.47%	L24H26	57.39%
5	L22H19	60.03%	L21H17	57.26%

Table 14: Top 5 performing components for Llama-3.1-8B-Instruct and Llama-3.1-8B-UltraMedical on JI task.

L Weighted Kendall's τ for component-rank agreement

Here we specify the ranked correlation metric used in Section 5.2. We use the weighted Kendall's τ_w of Vigna (2015), which places greater weight on agreement among the highest-ranked components.

Definition. The standard Kendall's τ counts concordant versus discordant pairs across two rankings r and s of n items:

$$\tau = \frac{C - D}{\binom{n}{2}}, \tag{4}$$

where C and D are the numbers of concordant and discordant pairs. Each pair contributes equally regardless of where it falls in the ranking, which makes standard τ a poor choice when interest is concentrated on the top of the ranking. The weighted variant assigns each pair (i, j) a weight that depends on the ranks of its members:

$$\tau_w = \frac{\sum_{i<j} w_{ij} \operatorname{sgn}\big((r_i - r_j)(s_i - s_j)\big)}{\sum_{i<j} w_{ij}}, \tag{5}$$

with $w_{ij} = w(r_i) + w(r_j) + w(s_i) + w(s_j)$ and hyperbolic weighting $w(r) = 1/(r + 1)$, where ranks are taken in descending order of the underlying score (highest-scoring item has rank 0). Pairs involving high-ranked items therefore contribute substantially more to τ_w than pairs in the tail. This property makes τ_w well-suited to our setting, where the highest-accuracy components are of primary interest.

Implementation. Because weighted τ ranks its inputs in ascending order by default, we negate the accuracy vectors so that higher-accuracy components receive lower (top-of-list) ranks, ensuring the weighting emphasizes agreement among the strongest components. Aside from that, we use the Scipy implementation with default parameters to compute the metric.

M Ablation Procedure

Following Michel et al. (2019), we implement ablation via per-component gate variables $\xi \in \{0, 1\}$ applied multiplicatively to each component's output before it is written to the residual stream. Setting $\xi = 0$ zeroes the component's direct contribution to the forward pass without modifying any learned parameters. Gates are applied via forward-pass hooks and take effect at every token position.

For the i'th attention head $h_i^{(l)}$ of layer l, recall that the multi-head attention output can be written as a sum over per-head contributions,

$$a^{(l)} = \sum_{i=1}^{n} \xi_i^{(l)} h_i^{(l)} W_{O,i}^{(l)}, \tag{6}$$

Where $W_{O,i}^{(l)}$ is the corresponding slice of the output projection matrix. Ablating head i corresponds to setting $\xi_i^{(l)} = 0$, so its contribution $h_i^{(l)} W_{O,i}^{(l)}$ vanishes from $a^{(l)}$. For an MLP block $m^{(l)}$, we analogously apply a gate $\xi_m^{(l)} = 0$ to the block's output, removing its contribution to the residual stream at layer l. Both interventions are applied at every token position throughout the forward pass and modify only the activation tensors; no learned parameters are altered.

Unlike the reweighting procedure in Section 5.3, which scales each component's direct contribution to the output logits, ablation modifies the residual stream itself. The ablated component's effect on all downstream layers is therefore also removed, not only its direct contribution to the final logits.

For the random-ablation control in Figure 4, we sample components uniformly without replacement from the set of all components excluding the top-20 ranked by individual decomposition accuracy, using a fixed seed. The sampled set is then ablated cumulatively in the same manner as the top-k condition.

	Jargon Understanding (JU)		Jargon Identification (JI)	
	Instruct	UltraMedical	Instruct	UltraMedical
Full model	76.05%	74.10%	59.10%	53.10%
1	L17H24 (−6.49)	L17H24 (−7.29)	MLP-19 (−0.43)	L26H3 (−0.13)
2	L30H27 (−0.67)	MLP-29 (−0.67)	L17H24 (−0.1)	L31H1 (−0.03)
3	MLP-28 (−0.33)	L22H19 (−0.67)	L30H26 (−0.07)	MLP-29 (−0.03)
4	L31H1 (−0.20)	L31H1 (−0.27)	L26H22 (−0.07)	L31H3 (+0.03)
5	MLP-26 (−0.20)	L30H27 (−0.2)	L25H8 (−0.03)	L30H27 (+0.03)
Random (max drop)	−0.20	−0.4	−0.23	−1.1

Table 15: Individual ablation results for the top-5 components on JU and JI. Values denote accuracy change (%) relative to the full model when a single component is ablated. Random (max drop) is the largest accuracy drop observed across 10 individually ablated random components sampled outside the top-20 performing components.

N Individual Component Ablation

To complement the cumulative ablation analysis in Section 5.2, we also report results from ablating each of the top-5 components individually. Each component is removed in isolation using the procedure described in Appendix M, and the resulting accuracy change relative to the full model is recorded. As a control, we also report the largest accuracy drop observed across 10 individually ablated random components (sampled outside the top-20 highest accuracy components). The individual ablation results are shown in 15. On JU, attention head L17H24 stands out as singularly important in both models: ablating it alone drops accuracy by 6.49% in Instruct and 7.29% in UltraMedical, far exceeding the change from any other top-5 component and well outside the range observed under random ablation. The larger drop in UltraMedical suggests that medical fine-tuning increases the model's reliance on this particular component rather than distributing jargon knowledge more broadly. Beyond L17H24, the remaining top components produce modest drops of less than 0.7% when ablated individually.

On JI, no single component exhibits a comparable individual impact. The largest individual drops are 0.43% (MLP-19 in Instruct) and 0.13% (L26H3 in UltraMedical), and several ablations even produce small positive changes within the range of noise. Notably, the maximum random ablation drop on UltraMedical (-1.10) actually exceeds the largest top-component drop, indicating that no individual component is disproportionately important for this task. This pattern is consistent with the flat per-component accuracy distribution observed in Section 5.2 and the shallow cumulative ablation curves in Figure 4. JI relies on aggregated contributions from many components rather than a few specialized ones.

O Experiment setup for component reweighting

We optimize the component weights $w \in \mathbb{R}^{1056}$ (1,024 attention heads + 32 MLP) separately for each task and each model. Weights are trained for 1000 epochs with SGD (lr = 0.01, λ = 0.01) with early stopping (patience = 3) on the training loss computed on the JI training set (1,200 examples) and evaluated on the 2,990-item JI test set. The same procedure is applied to JU using its 4,183-item training set and 1,495-item test set.