Combining physics education and machine learning research to measure evidence of students' mechanistic sensemaking

Kaitlin Gili,[1,*] Kyle Heuton,[1] Astha Shah,[1] David Hammer,[2,†] and Michael C. Hughes[1,†]

[1]*Department of Computer Science, Tufts University, Medford, Massachusetts, USA*

[2]*Department of Education and Department of Physics and Astronomy, Tufts University, Medford, Massachusetts, USA*

(Received 13 April 2025; accepted 4 December 2025; published 22 December 2025)

Advances in machine learning (ML) offer new possibilities for science education research. We report on early progress in the design of an ML-based tool to analyze students' mechanistic sensemaking, working from a coding scheme that is aligned with previous work in physics education research (PER) and that is amenable to recently developed ML classification strategies using language encoders. We describe pilot tests of the tool, in three versions with different language encoders, to analyze sensemaking evident in college students' written responses to brief conceptual questions. The results show, first, that the tool's measurements of sensemaking can achieve useful agreement with a human coder, and, second, that encoder design choices entail a tradeoff between accuracy and computational expense. We discuss the promise and limitations of this approach, providing examples of how this measurement scheme may serve PER in the future. We conclude with reflections on the use of ML to support PER research, offering cautious optimism for strategies of codesign between PER and ML.

DOI: 10.1103/v6gw-lv4c

I. INTRODUCTION

Advances in machine learning (ML) and natural language processing [1–5] have generated a rush to develop applications in science education, often focused on supporting the analysis and assessment of student thinking [6]. Much of the promise is to make coding or grading of student work more efficient, as manual assessment of student thinking requires time-intensive, subjective human judgment. A key challenge is to provide successful ML support without placing burdensome requirements for expensive computational or data resources.

At the same time, scholars argue for patience and care: AI does not provide a "magic bullet" [7] for research or instruction. The benefits of these new technologies will depend on careful investigation, including collaboration and codesign between ML and physics education researchers. Examples include recent work by Fussell and her colleagues to propose a methodology for assessing the trustworthiness of machine coding, illustrated with analyses of students' open responses to questions about empirical science [8]. In subsequent work, they compare results across different large language models (LLMs) in analyzing skills evident in student lab notes [9].

We are focusing on the challenge of measuring students' *sensemaking*, which researchers have identified as an essential target for research and instruction, but have been constrained to study in specific episodes or small sets of data [10–21]. These studies took place in a variety of situations—different subject matter, levels of education, forms of data—and there is substantial variation across them in meanings of and nature of evidence for sensemaking. Odden and Russ [10] described the literature as "theoretically fragmented." To resolve this issue, they proposed a definition of sensemaking as "a dynamic process of building or revising an explanation in order to "figure something out"—to ascertain the mechanism underlying a phenomenon in order to resolve a gap or inconsistency in one's understanding." We take this as a starting point for our work, recognizing several features we intend ultimately to study: (i) the perception of some gap or inconsistency in one's understanding; (ii) reasoning with the purpose of addressing that gap; and, for science broadly and physics in particular; and (iii) an expectation of a *mechanism* underlying the phenomenon [22].

For now, we focus on developing ML to analyze student reasoning with respect to mechanism, drawing on an earlier analytic framework by Russ and her colleagues for human analysis of mechanistic reasoning [22]. We review that framework below and discuss the need for its adaptation. Designing an approach to ML involves a myriad of contingencies and decisions that depend on

*Contact author: kgili01@tufts.edu

†These authors contributed equally to this work.

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

the particularities of the situation. Here, the particular data we study consist of college students' brief 1–5 sentence explanations for answers to multiple-choice conceptual questions in introductory mechanics. The data comefrom a 100-student calculus-based course at Tufts University, which used *FlipIt Physics* [23] and emphasized students' learning to engage productively with confusion [24]. Given these data, two specific questions motivate our current efforts. RQ1: How can we refine an analytic framework from physics education research (PER) into a measurement scheme amenable to ML with this dataset? RQ2: For ML implementations of this scheme, how do design decisions that limit the computational resources needed impact how well the ML-provided labels agree with human coding?

The remainder of this article is organized as follows: Sec. II provides background with respect to the data we aim to analyze, prior work on mechanistic reasoning, and prior work on incorporating ML in science education research. Section III presents an account of adapting the framework from Russ *et al.* [22] into a measurement scheme for our particular dataset, addressing RQ1. Note that we will use the term *scheme* and the phrase *measurement scheme* throughout the paper to refer to this adaptation, which is amenable to ML. We present the scheme with examples of its application, and we describe the ML in technical detail. Part of our purpose in this account is to provide an example of convergence between PER and ML in the codesign of analytic tools.

Section IV delves into a specific aspect of the design, RQ2, to compare ML performance across three possible language encoders. In this section, we consider both the computational resources required for the model and the level of agreement with a human coder. We describe how this approach, which currently depends on problem-specific training, may eventually generalize to support analysis of student responses to new problems without that step of training. In Sec. V, we discuss the promises and limitations of this use of ML, specifically touching on utility, efficiency, generalizability, reliability, sustainability, and privacy.

Finally, in Sec. VI, we reflect on our efforts to date, considering both our specific aims in developing this particular tool as well as our and the broader community's more general aims of developing artificial intelligence for applications in education. As will be clear, we are not presenting settled science. Rather, we are giving examples to illustrate the challenges and possibilities of this work.

To enable others' explorations, we provide open-access data and code resources in a public Github repository [25], under the project name "AuSeM." There, we release our dataset of 385 total student responses across four physics problems, each manually annotated using our approach. We also provide a codebook of instructions for human annotators, which reflects updated clarifications that emerged from our interrater agreement process. These resources could be used to train other models or as a guide for further manual annotation efforts on other data. Our provided Python code can be used to reproduce our experiments (retrain each architecture on our data) or generate mechanistic sensemaking labels for *new* student responses on the same set of four problems.

II. BACKGROUND

A. The challenge to analyze sensemaking

The context for this effort is a calculus-based introductory course in Newtonian Mechanics at Tufts University. The course places unusual emphasis on students' sensemaking, with explicit goals of their learning to recognize and to engage proactively with confusion and uncertainty [26]. The data we examine come from spring 2022, when the course was using *FlipIt Physics* [23]. Before each class all semester long, students watched a 20-min "prelecture" video introducing new concepts, in preparation for the in-person lecture class. After the video, students were required to answer a multiple-choice conceptual question to check their understanding. For each problem, they submitted an answer as well as a written text explanation of their choice.

As a motivating example, consider the following problem from the fourth unit of the curriculum:

Example: Problem 4

1. The net force on the box is in the positive x direction. Which of the following statements best describes the motion of the box?
(a) Its velocity is parallel to the x axis.
(b) Its acceleration is parallel to the x axis.
(c) Both its velocity and acceleration are parallel to the x axis.
(d) Neither its velocity not its acceleration need to be parallel to the x axis.

2. Briefly explain your answer to the previous question.

Our challenge is to analyze written text data from student explanations as evidence of their engagement in mechanistic sensemaking. Here are some examples of students' explanations:

E1: "According to Newton's second law, F (net) = ma, we could know that the direction of a is the direction of F (net)" (answer b).

E2: "If the force is acting on it in the positive x direction, then the box must be moving in the positive x direction and will therefore have a parallel acceleration to the x axis since it is not speeding up. The velocity would also be parallel because its direction is positive. I disagree with myself as I type this but I'm not sure what my opposing argument is." (answer c).

E3: "If a 3 people were pushing the box but one person was pushing harder than the rest, the box would accelerate in the same direction as the greater force. However, with velocity, the final velocity has to take into account the initial velocity. Though the acceleration might go to the right, if the velocity was going upwards, the box might move at a slant." (answer b).

It is a challenge to decide what to take as evidence of sensemaking, or of non-sensemaking. The first makes sense to a physicist, but is it evidence of sensemaking for the student? The second does not make sense to a physicist (the box accelerates "since it is not speeding up"?), but the student noted experiencing uncertainty; perhaps, it is evidence of their trying to identify that uncertainty. Example E3 makes sense to a physicist, and it also shows the student constructing a tangible physical situation.

Ultimately, we hope to make progress in tapping these data for analysis of sensemaking, to study whether and how students in the course make progress in learning how to learn. In this work, we start in that direction by analyzing for evidence of mechanistic reasoning. Mechanism is consistently described in accounts of sensemaking in physics [19,21], and there is a well-established framework for analysis [22].

B. Mechanistic reasoning in science

Russ *et al.* [22] adapted work by philosophers of science Machamer *et al.* [27], who had analyzed the concept of mechanism within areas of biochemistry. This led to an analytic framework made up primarily of seven levels organized in a hierarchy:

1. Describing the target phenomenon that is observable and reproducible.
2. Identifying setup conditions in the spatial and temporal organization of the entities that lead to the phenomenon.
3. Identifying entities involved in producing the phenomenon.
4. Identifying activities, the "actions and interactions that occur among the entities."
5. Identifying properties of entities that are relevant for producing the phenomenon.
6. Identifying the spatial organization of entities that are specifically relevant for the actions and interactions among the entities.
7. Chaining backward or forward, in inferences about the entities and activities that could have led to (backward) or could follow from (forward) a given situation.

These seven categories are logically hierarchical in that higher levels generally require lower levels. For example, chaining forward or backward would generally require identifying entities, properties, and activities. Russ *et al.* [22] added two categories they considered nonhierarchical:

Analogies are explicit comparisons to other phenomena. Animated models in students' use of physical actions, whether manipulating objects available or gestures, to simulate the entities and activities they are describing.

In the 20 years since its introduction, this framework has been applied broadly in PER and other areas of science and engineering education research [21,28,29].

The first example explanation from our data, labeled E1 above, does not include any entities or activities; there is no evidence of mechanistic reasoning. That does not mean there was no mechanistic reasoning; only that none is evident in the provided text. Nor does it mean the reasoning is bad; it is correct.

The second explanation E2 gives clear evidence of entities (the box) and activities (its movement). Perhaps the student is also thinking of the force as an entity? Perhaps, too, there is a chaining, a force causing motion, but the logic is difficult to follow—including evidently for the student. It is "moderate evidence of mechanistic reasoning," by the framework of Russ *et al.*

Example E3 has evidence of entities (people, the box), activities (pushing, motion), and chaining (the box would accelerate in the direction of the greater force/push). There is also evidence, although it is not quite as clear, of spatial organization (if the force is sideways, by the people pushing, but the box is moving upward, the box might move at a slant). Overall, this example is "strong evidence of mechanistic reasoning."

Note that applying the seven-level general framework of Russ *et al.* to this problem involves both specification and simplification. In our data, student responses do not describe a target phenomenon, as the problem prompt does that already. Within the identification steps, there are only a small set of possibilities for what could arise as entities and activities. As we describe below, in our work to date, we take this step, seeking a measurement scheme that can be applied both manually and via ML.

C. Incorporating ML into science education research

ML tools are now rapidly integrating into PER, predominantly via text classification methods—algorithms that learn patterns in student text responses and sort each response into distinct categories or labels [30–35]. This rapid integration has been largely driven by recent advances in large language models (LLMs) [1,2,36] trained on vast amounts of language corpora (books, Wikipedia, websites, etc.). The availability of chatbots like ChatGPT, powered by LLMs, makes it possible for PER researchers to directly feed text into a chatbot window to be classified [37]. Below, we contrast different strategies for using LLMs in PER.

One paradigm for LLM-enabled PER is *in-context learning* (ICL), where an existing chatbot is provided a student text response and prompted via natural language

instructions to classify it into a desired labeling system. To be clear, the "learning" here is euphemistic; ICL does not entail retraining the LLM itself at all. Instead, it relies on the pretrained ability of the LLM to transform an instructional prompt and provided input text into useful output text. ICL is data efficient, with proponents often providing only a few example demonstrations in the instructions to clarify the desired labeling transformation. However, ICL has a number of important limitations. First, ICL's utility relies on the assumption that the LLM is useful for a specific downstream language task in education (correctness, reasoning, sensemaking, etc.). The underlying LLM was not trained to do well on these tasks, and it is important to verify at scale whether an LLM is reliable at certain tasks. ML researchers have sometimes questioned the utility of ICL, showing that "demonstrations have a marginal impact" [38] and "randomly replacing labels in the demonstrations barely hurts performance" [39]. Second, ICL is sensitive and variable. Outputs can vary depending on the exact prompt [40] and the selection and order of examples [41]. Even the same exact input may lead to different outputs due to the stochastic nature of chatbots. Third, LLMs reflect biases inherent in the source data [42]. Finally, there is a lack of transparency: closed-source versions such as ChatGPT do not allow users to inspect, understand, or modify how outputs are obtained. For these reasons, in-context learning methods deserve and require further investigation for their reliability in science education research.

Other works focus on training supervised ML predictors to accomplish their specific task, integrating accepted human annotation frameworks in science education with machine learning practices [29,32,33]. Watts *et al.* [29] is a prime example of this type of integration between science education and machine learning research. Watts *et al.* [28] had previously adapted Russ *et al.* [22] for use in organic chemistry; follow-up work [29] adapts further to apply ML binary classifiers. This entailed separating Russ's original categories into stand-alone binary (present or not present) codes specific to organic chemistry problems. For example, they defined the specific code *electron movement* within Russ's *identifying activities*. The authors treat each code as independent evidence of mechanistic reasoning, without the hierarchical organization feature of Russ's framework, which simplifies the ML methodology, a useful first step toward integrating ML and science education research. Student responses are used as training data for separate binary classifiers for each code, where various classifier types are attempted, and only the final results of the highest- performing classifier are reported.

Similar to Watts *et al.*, we take the step of adapting the framework to specific features of the tasks in our dataset, and we simplify to binary codes, as we describe in the following section. We will discuss similarities and differences in technical detail in Sec. III C.

Fussell *et al.* [8] provide an evaluation of different large language models for measuring specific skills in students' typed lab notes. The authors compare off-the-shelf pretrained open-source LLMs with ones that are *fine-tuned* (further trained) on training data specific to their task. The results showed that higher-resource models (e.g., more training data, computational power) often outperform lower-resource models for their particular task, although depending on the evaluation metric, there are exceptions. More specifically, the authors found that all fine-tuned models outperformed the off-the-shelf pretrained models, and they suggest that for theory-driven classification tasks (i.e., where there is a specific task in science education), fine-tuning might be more useful.

There are a few key distinctions between our current work and Fussell *et al*. Similar to Fussell *et al.*, we compare the performances of ML model variants in the context of our specific task; however, we do not attempt to provide more general insights about the models on other tasks or datasets. We will discuss the similarities and differences between Fussell *et al.* and our work in more technical detail in Sec. III C.

III. RQ1: PER-ML MEASUREMENT SCHEME

A. Refining the framework for a particular dataset

Russ *et al.* [22] designed their framework to be applicable across areas and levels of scientific reasoning. That breadth of applicability has been helpful for the research community, but its generality presents challenges for the development of an ML tool with context-specific datasets. Given a particular dataset, such as the conceptual question above about a net force acting on a box, it is possible and advantageous to tune it to the specifics of the task. The strengths and limitations of machine learning motivate other simplifications as well.

1. Task-specific considerations

We are interested in measuring student responses to how they solved a *particular* problem within a set of closed-ended introductory physics problems. Each problem contains its own specific entities, properties, activities, etc., that are relevant. The box question, for example, mentions only one entity, and there is a limited set of entities (such as people to push) that students might add to the situation. As such, the mechanistic evidence that we are looking for is problem dependent, which allows us to simplify the scheme in several ways.

1. *No need to redescribe the target phenomenon*: Each problem's instruction text makes the target phenomenon clear; students focused on *brief* explanations and thus did not restate that phenomenon in their own words. For these reasons, we do not include this first code from Russ's framework in our scheme.

2. *Restricted set of entities, properties, and activities*: The "net force on the box" example problem names only the box as an entity. There are, in principle, other entities one could mention as involved, such as the Earth or the surface under the box, but no students in this dataset mention them. For each problem in our data, we confine our analysis to a limited, analyst-defined set of entities, properties, and activities that could reasonably be relevant or that arise in the data.

2. ML design considerations

While we want custom annotations for each problem, developing a separate classifier for each problem would be prohibitively expensive and unlikely to work well without many training examples. We are thus interested in combining powerful language encoders [1–3,36] with *shared* classification heads that take advantage of similar word occurrences and grammatical structures across problems to *efficiently* annotate new examples. As such, it is important to construct multilabel classification categories that share similar linguistic features across student responses and problems. This influences three key design decisions.

1. *Share initial and subsequent spatial organizations*: We assume a description of the spatial organization of entities will contain similar semantic and structural features regardless of whether it refers to the initial, intermediate, or final conditions. These temporal states can be represented by a separate context embedding (further discussed in Sec. III C), such that evidence for each temporal state can still be individually annotated, but spatial linguistic features can be shared.
2. *Separate central from noncentral entities*: By definition, central entities are connected to the other codes (properties, activities, etc.). As such, central and noncentral entities will contain dissimilar relationships among words in a student response. Keeping these categories separate is especially useful for future work that incorporates dependencies between domain labels.
3. *Separate movement activities from interactions*: Descriptions of individual entity movement activities (e.g., the object is moving with an initial velocity) are different from descriptions of the interactions between entities (e.g., the hand pushes the book), both semantically and structurally within a sentence.

Like Watts *et al.* [29], we relax the hierarchy of Russ's framework, coding only for the presence or absence of evidence in each category. This simplifies our annotation scheme and ML method for now; later in Sec. VI, we discuss plans going forward to reintroduce some form of that hierarchy. This would be an improvement on our current modeling assumptions: for example, evidence of properties is likely to correlate with evidence of entities.

3. Proposed domains

Given the above considerations, we propose seven domains, broad categories of yes or no questions that measure evidence of mechanistic sensemaking in student thinking via the framework:

1. **Objects.** Identifying the central entities that interact to produce the physical phenomena at a single hierarchical level. These objects have properties, spatial layout, and movement relevant to the solution process. Examples: "car," "box," "ball," "particles," etc.
2. **Influences.** Identifying the noncentral environmental entities that affect the objects and their activities. Only the influence → object interaction is relevant to the phenomena. Examples: "Earth's gravitational field," "human hand," "electromagnetic field."
3. **Properties.** Identifying the relevant qualities that are confined to the physical boundary of an object. Examples: mass, temperature, and volume.
4. **Spatial organization.** Identifying the spatial states or orientations of an object. Examples: "downward," "upward," "left," "right," "x direction," "y direction."
5. **Movement activities.** Identifying the motion states of an object. Examples: "initial velocity," "slowing," "speeding," "accelerating."
6. **Interactions.** Identifying the physical exchanges between all object ←→ object and influence → object pairs. Examples: "gravitational force," "human pushing," "electromagnetic force."
7. **Chaining.** Identifying the characteristic (a property, spatial organization, or movement activity) changes that result from specific object ←→ object interactions and influence → object interactions. Examples: "constant acceleration of a ball caused by Earth's gravitational field exerting a downward net force"; "acceleration of a book initially at rest caused by a human push."

Our scheme corresponds with Russ's framework as follows: the properties domain directly maps to Russ's category 5; the objects domain and influences domain combine to obtain 3; the spatial organization domain encompasses 2 and 6; the movement activities and interactions domains combine to form 4, and the chaining domain maps to 7. We do not consider stating the target phenomena, analogies, or mental model animation in this work—although these are interesting additions for future research.

4. Example problem and associated coding

Consider the "force on a box" problem from Sec. II A. An example student explanation is shown in Fig. 1, with specific parts color-coded by our 7 domains. This coding

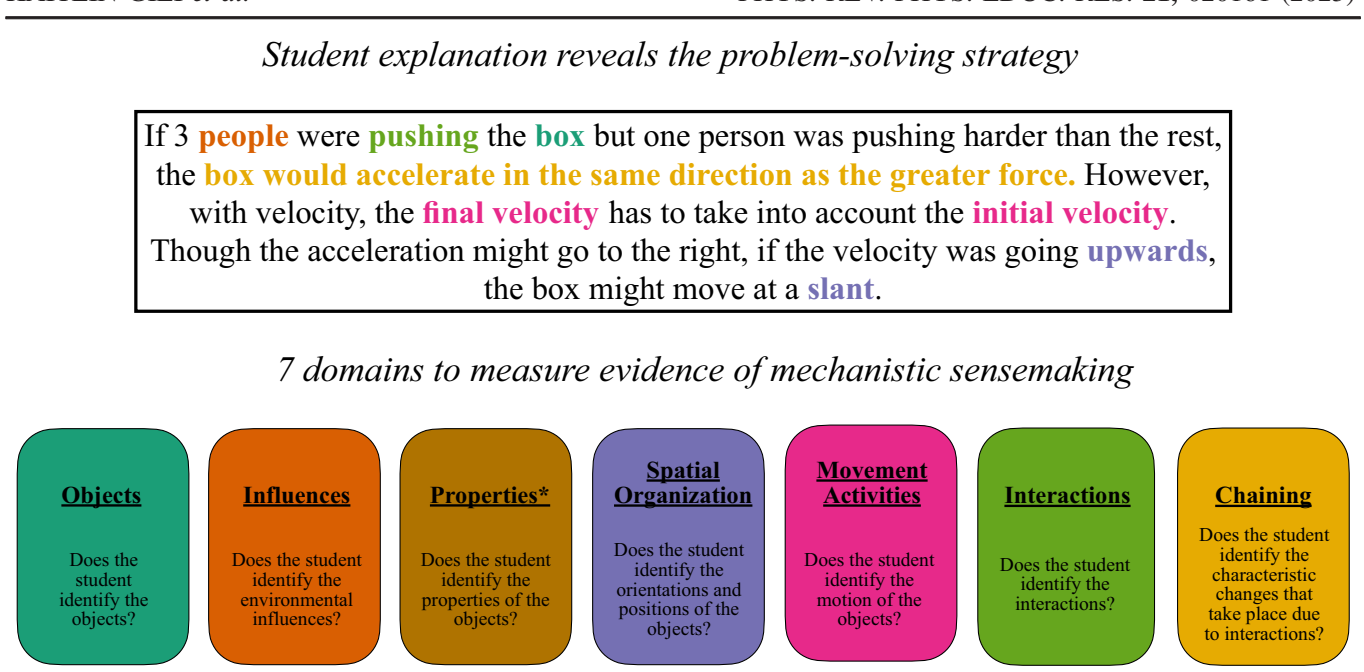


FIG. 1. An annotated student text explanation to a "force on a box" problem (defined in Sec. II A). We display the seven domains used to measure evidence of a mechanistic sensemaking process in the student's text. Each domain may contain a different number of yes-or-no criteria questions to answer depending on the problem. We color words and phrases that would prompt yes answers to the criteria questions within that same-colored domain. The * next to a domain name signifies that the student did not explicitly identify its criteria. More than one word or phrase can be annotated in the same color if multiplicities exist for that domain (e.g., spatial organization 1: "upward," spatial organization 2: "slant"). This particular student displays *high evidence* of mechanistic sensemaking ($s = 0.89$) by imagining a scenario where multiple interactions influence the box's dynamics and by describing the mechanistic relationship between a net external force and a change in the box's motion.

demonstrates how parts of an individual student response can be associated with each of the seven domains. These specific parts could be later used to answer yes-or-no questions to quantify evidence of sensemaking, as described in Sec. III B. Some domains correspond to multiple parts of the response (e.g., the object has more than one spatial organization in the context of the overall problem). Note that not all seven domains need to be coded for if some are irrelevant to a problem, although all domains are relevant to the example problem here.

In Secs. III B and III C below, we introduce how we apply quantitative labels to measure mechanistic sensemaking via our scheme and operationalize our modular classification model, for readers who are interested in the technical details.

B. Measurement as multilabel binary annotation

For a given physics problem, our measurement scheme requires the annotator to develop a problem-specific list of yes-or-no *criteria questions* for each of the seven domains. Thus, given a dataset of potential problems and student responses to be measured, an annotator can follow the proposed steps:

Step 1. Select P problems from an introductory physics course on classical mechanics, where there exists a physical target phenomenon to be understood.

Step 2. For each problem $p \in \{1, .., P\}$ and for each domain $d \in \{1, .., 7\}$, determine the set of yes-or-no criteria questions C_d^p that are pertinent to the specific problem. For example, in problem 4 from Sec. II A, there is only a single object, and thus $|C_1^4| = 1$, with the criteria question "*Does the student identify the object?*." However, there are two spatial organization criteria questions $|C_4^4| = 2$: *Does the student identify the object spatial organization before the net applied force? Does the student identify the object spatial organization after the net applied force?* We denote the total number of criteria questions for problem p as $T_p = \sum_{d=1}^{7} |C_d^p|$. The full list of criteria for problem 4 is displayed in Appendix A.

Step 3. Given a student's text response to problem p, the annotator will use the yes-or-no criteria questions to obtain a vector of binary digits representing their answers. The resulting criteria vector for a student response n is denoted as $\mathbf{y}^n \in \{0, 1\}^{T_p}$. To further compute a sensemaking score $s \in [0, 1]$ one can sum the number of 1s and divide by the length of the vector T_p. This score is to be interpreted as the *amount of evidence of mechanistic sensemaking*. An annotator could decide to weight the domains (or specific criteria) differently. In this work, we keep the weights equal. To finish our example, the criteria vector for the student response in Fig. 1 is $\mathbf{y}^n = [1, 1, 0, 1, 1, 1, 1, 1, 1]$ with the corresponding sensemaking score 0.89, a "yes"

code for all domains but "property." Considering the response itself, a high score seems appropriate: The student imagined a tangible scenario with multiple influences on the box.

C. Multilabel binary classification model

Here, we describe an ML model we constructed for our task that can automatically map a student's text response to a numerical sensemaking score, where a higher value indicates greater evidence of mechanistic sensemaking. While our approach relies on problem-specific criteria, some aspects of the model are shared across problems to gain statistical strength. The overall model is modular components like the encoder can be varied to trade off between model capacity and resource consumption.

1. From text to embeddings

Given a student response n, we can select an *encoder* to map all text in the response to a numerical representation $\mathbf{x}^n$. A simple encoder design is *counting*; we determine a vocabulary $\mathcal{V}$ as the set of words across all student responses in the training dataset, with an additional vocabulary entry for unknown words, and record the number of times each one appears in the nth example. This numerical representation, or embedding $\mathbf{x}^n \in \mathbb{R}^{|\mathcal{V}|}$, typically called *Bag-of-Words*, reduces the richness of the text explanation to the pattern of word occurrence [43,44]. From here on, we will refer to this encoder as `BoW`.

To capture more complex patterns (e.g., relative co-occurrence between all words in the response, in-context word occurrence, and word positional associations with grammar rules), one can use a *large language model (LLM) encoder* [45]. For example, Google's *BERT* (Bidirectional Encoder Representations from Transformers) model combines tokenization, self-attention, and a multilayer perceptron neural network - all pretrained on English Wikipedia and BookCorpus to generate token embeddings [1]. A token is a set of characters containing a unique ID in BERT's vocabulary, such that an entire text can be converted to tokens with individual embeddings $\mathbf{x}^n_{token} \in \mathbb{R}^{768}$, which carry information about all other tokens in the passage. BERT also uses a sentence-level embedding, known as $\mathbf{x}^n_{CLS} \in \mathbb{R}^{768}$, which can be used as a full-response representation rather than using distinct representations for each token. From here on, we will refer to this encoder as `BERTfrozen`. As a final encoder option, the sentence embedding $\mathbf{x}^n_{CLS}$ can be fine-tuned by training BERT's previously trained weights $\{\boldsymbol{\theta}\}$ on new downstream classification tasks [46]. We will refer to this encoder as `BERTfinetuned`. This allows for the $\mathbf{x}^n_{CLS}$ embedding to capture patterns in the text that are specific to the target prediction task. It requires significantly more computational resources, however. Current LLMs, such as Google's Gemini or OpenAI's ChatGPT, require far more still. Part of our interest is to minimize the computational demands of the ML.

2. Shared classifiers across problems

Given an embedding representation $\mathbf{x}^n$ for a student response, we want to correctly assign it to a criteria vector $\mathbf{y}^n$. Thus, we have a multilabel task [47]. If we were to construct *separate* binary classification models for each criteria question, then we would be required to train $T_p * P$ independent classifiers. Rather, we propose to work from an informed assumption: shared domains will have similar word occurrences and grammatical structures across P problems. For example, the word "speeding" should indicate evidence of sensemaking in the movement activities domain, regardless of the problem. Furthermore, words that should be classified as objects will share similar positions in the sentence across problems due to object-subject-verb rules. Thus, for P problems, we propose to train seven classifier heads, one for each domain. We reduce from a scaling linear in P to constant in P.

In order for this shared set of classifiers to effectively annotate multiple problems, each with multiple yes-or-no criteria questions, we design each classifier to take an additional input beyond the embedding of student response text: a criteria *context* embedding $\mathbf{q}^c_d$ that encodes the text of the yes-or-no question being asked (e.g., *Does the student identify the object spatial organization before applied force (e.g., upward)?*). The index $c \in \{1, .., |C^p_d|\}$ specifies the criteria question the classifier should label.

Each classifier must then take in student response embedding $\mathbf{x}^n$ and criteria context embedding $\mathbf{q}^c_d$. There are multiple possible ways to combine the two inputs into a joint representation $\mathbf{z}^n_c$ of response and context vectors that is the ultimate feature vector for the classifier. As a starting point, we have selected a simple approach of adding the embeddings together: $\mathbf{z}^n_{c,d} = \mathbf{q}^c_d + \mathbf{x}^n$. Possible alternatives would be to concatenate the two vectors together, or to produce a single embedding from the combined text of the question and response. We leave the exploration of alternative methods for future work.

Figure 2 visually illustrates how the model makes a probabilistic prediction for a specific yes-or-no criterion given a student's response to a question. Given an embedding $\mathbf{z}^n_{c,d}$, we use a domain-specific logistic regression (LR) classifier to produce the predicted probability $\rho^n_{c,d} \in (0, 1)$. We set $\rho^n_{c,d} = \sigma(\mathbf{W}_d \mathbf{z}^n_{c,d} + b_d)$, where $\mathbf{W}_d, b_d$ are the weight vector and bias scalar for domain d, and σ denotes the logistic sigmoid function.

The figure also shows how that prediction impacts the loss, which measures the difference between the model's current predictions and human coding. For training, we have a dataset of student embeddings with human-annotated labels (the criteria vectors) $D_{\text{Train}} = \{\mathbf{x}^n, \mathbf{y}^n\}_{n=1}^N$. Each $y^n_{c,d} \in \{0, 1\}$ represents a single binary label with

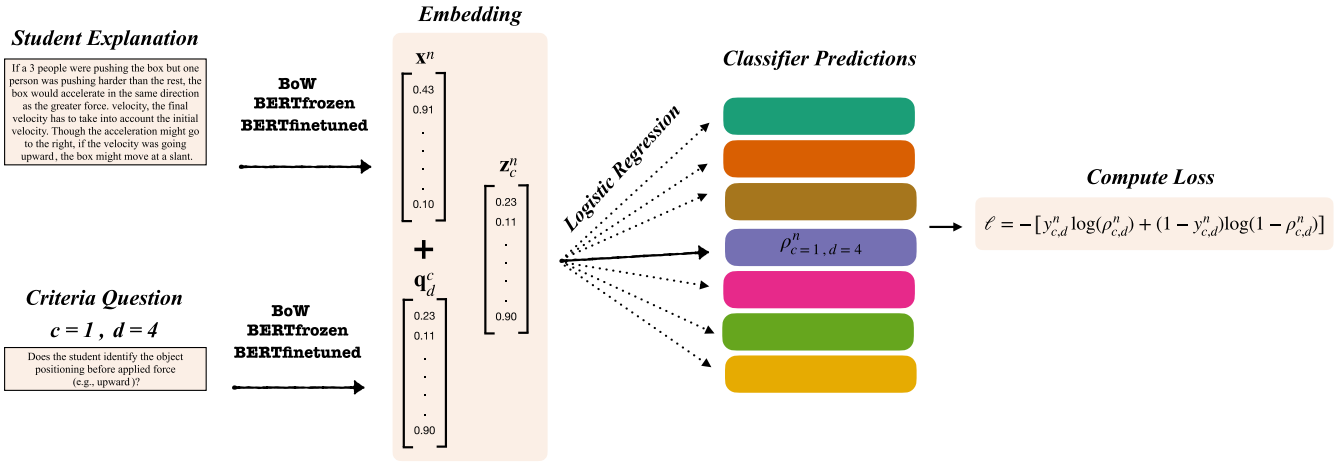


FIG. 2. How the proposed model makes a probabilistic prediction to answer a yes-or-no criteria question given a student response. Using a selected encoder (`BoW`, `BERTfrozen`, and `BERTfinetuned`), the student response is embedded into a numerical vector representation $\mathbf{x}^n$. An embedded representation of the criteria context $\mathbf{q}^c$ is added to $\mathbf{x}^n$, producing an updated embedding $\mathbf{z}_c^n$ contains information about both the student response n and the current criteria question c. Finally, out of seven possible domain-specific logistic regression classifiers, the current question's domain d selects its specific classifier to produce a predicted probability $\rho_{c,d}^n$ of answering "yes" to the criteria question. As a training objective, we use the cross-entropy loss by comparing this probability to the supervisory ground-truth label $y_{c,d}^n$. Each LR classifier's weights and bias are updated via gradient descent to minimize the sum of this loss over the entire training set. When using `BERTfinetuned` as the encoder, encoder parameters are also updated to minimize the loss.

respect to criteria question c of domain d for the student response n. Each $y_{c,d}^n$ is assumed conditionally independent of other labels given $\mathbf{x}^n$ and all model parameters. We minimize the following binary cross-entropy loss

$$\ell(\rho_{c,d}^n, y_{c,d}^n) = -y_{c,d}^n \log(\rho_{c,d}^n) - (1 - y_{c,d}^n) \log(1 - \rho_{c,d}^n). \tag{1}$$

This loss is *aggregated* over all explanations n, domains d, and criteria c. We also include an added regularization term that penalizes the sum-of-squares of the weight parameters. For the precise loss notation utilized in our methods, including regularization terms, see Appendix C. The training procedure encourages all seven classifiers to accurately predict their assigned pieces of the criteria vector $\mathbf{y}^n$ for an embedded student response $\mathbf{x}^n$.

For some encoder choices, we do not update any encoder parameters during training; only the classifier weights and biases $\{\mathbf{W}_d, b_d\}_{d=1}^D$ are updated via gradient descent. When *fine-tuning* BERT, our gradient descent procedure updates encoder parameters $\{\boldsymbol{\theta}\}$ alongside classifier weights. Both are explored in our model comparative analysis in Sec. IV.

3. Technical comparison with related work

Similar to Fussell *et al.* [8], we train a *BagofWords* (`BoW`) encoder and a `BERTfinetuned` encoder. We also observed that the `BERTfinetuned` encoder achieves better performance than the `BoW` encoder for our specific task. In general, our later results support their conclusion that fine-tuning adds value for a theory-driven classification in PER. They also compared two model architectures with multiple LLaMa encoders [2], which require significantly more computational resources than our BERT encoders. Similar to Watts *et al.* [29], we train multiple binary classifiers for task-specific evidence of mechanistic reasoning. They determined that convolutional neural networks (CNNs) outperform other model architectures such as naive Bayes, logistic regression, support vector machines, and transformers. Their results show high accuracy on the test data with slightly lower Cohen's kappa values to account for the data imbalance. We chose to continue with BERT encoders, expecting them to provide a better inductive bias for our task than CNNs, especially when pretrained on large language corpora and then fine-tuned on our limited data.

IV. RQ2: HOW ML DESIGN CHOICES CAN IMPACT PERFORMANCE

We seek a performance comparison of the proposed classification models outlined in Sec. III C with different encoding and training strategies. More specifically, our intention is to estimate how these models will perform given explanations from new students that they have never seen before, but on the same set of problems. Below, we outline our implementation and evaluation procedures, then present and discuss results.

A. Dataset of student responses

Our dataset, which can be found in Ref. [25], consists of 385 total deidentified student responses across four problems distributed to 100 students in the first few lectures of

the 2022 Tufts University Introduction to Physics course. The retrospective analysis of these deidentified human subjects' data was approved by our institutional review board.

All problems have the same structure as problem 4 in Sec. II A shown above and are presented to the students as prelecture digital assignments [23]. See Appendix A for the exact problem text. The same 100 students are asked to solve each problem and explain how they obtained their solution. A few students chose not to respond to a particular problem, which brings our total number of explanations to 385.

B. Annotating ground-truth labels

Complete annotation for all four problems was carried out by a domain expert: the first author of this work, who holds a doctorate in physics. This included defining the yes-or-no criteria questions for each problem (provided in Appendix A) and assigning binary values for each student response. These problems vary in their criteria vector size: $\{T_1 = 7, T_2 = 11, T_3 = 11, T_4 = 9\}$. Some descriptive statistics summarizing annotations across problems are provided in Appendix B. The annotations can be found in Ref. [25].

We validated interrater reliability *post hoc*. We emphasize that this reliability assessment was conducted to provide evidence for the consistency across human raters using our provided codebook. As fully described in Sec. V C, we found "substantial agreement" or better after a collaborative round of revision. Only the original annotator's labels were used to train the ML models; labels from secondary raters used to quantify IRR did not impact model training at all. In future work, we plan to incorporate training examples from multiple expert annotators.

C. Experimental setup

To evaluate model performance on never-before-seen responses, we randomly select 15 students who responded to all four questions to use as a held-out test set. We use the rest of the data (325 samples) for training and validation. To prevent data leakage, we split the data by student to ensure that all responses from the same student across four problems end up either exclusively in the training set or exclusively in the test set.

Using this train and test split, we fit the models `BoW`, `BERTfrozen`, and `BERTfinetuned` described earlier. All models are trained with the Adam optimizer [48] for 1000 epochs. Each training job for a `BoW` and `BERTfrozen` model was run on 4 CPU cores, while each `BERTfinetuned` job was run with 1 GPU (Nvidia A100) supported by 4 CPUs, all hosted on Tufts' High Performance Computing (HPC) center. Unlike `BERTfinetuned`, `BoW` and `BERTfrozen` do not require GPU systems to be trained effectively. Convergence was verified by observing the loss function behavior. We perform a grid search over key hyperparameters: learning rate α and regularization strength λ (λ_1 for classifier weights and λ_2 for *BERT* model weights); reproducible details are available in Appendix D. For `BoW` and `BERTfrozen`, we use fivefold cross-validation, also split by student ID, to select hyperparameters. We pick the combination with the best average area under the receiver operating characteristic (AUROC) score across folds.

The primary performance metric we report and evaluate is the *AUROC*, the area under the receiver operating characteristic (AUROC) curve. For a given classifier, this curve depicts the true positive rates (y axis) and false positive rates (x axis) achieved across different numerical thresholds for turning predicted probabilities into binary decisions. There are two main reasons we believe AUROC to be the most useful metric for our context: First, AUROC does not require us to choose an arbitrary probability threshold to distinguish label 1 predictions from label 0 predictions. Instead, AUROC sums over all possible thresholds, providing an overall numerical summary where increasing values indicate better performance. AUROC is bounded in $(0, 1)$. Second, due to imbalanced annotations in our training dataset (i.e., some multilabel tasks have more of the positive or negative class rather than a 50-50 split), the AUROC is an easier-to-interpret metric to report than precision or accuracy, as its chance baseline is 0.5, independent of the imbalance. For these reasons, we use AUROC as the primary performance metric throughout our analyses.

For the interested reader, we include additional metrics in the appendix: the area under the precision-recall curve (AUPRC) can be found in Appendix E and confusion matrices with Cohen's kappa values for a decision threshold of 0.5 can be found in Appendix F.

D. Fine-tuning details

For `BERTfinetuned`, we require a larger amount of computational resources per experiment. Our grid search over hyperparameters entails 324 separate efforts to train this encoder's millions of weights. We thus used a single training and validation set to run the hyperparameter grid search, avoiding the extra costs of fivefold cross-validation. To better preserve the prior information learned in *BERT's* original training, we use L2SP regularization [49,50] when training `BERTfinetuned`. This means we enforce two separate sum-of-squared-difference penalties in the loss, one for *BERT* encoder weights and another for the LR classifier head weights. Each has a different desired mean and a different strength hyperparameter that is tuned. We set the batch size to $N_{\text{batch}} = 32$, which has been shown empirically to work well with deep neural networks [51].

During fine-tuning, we originally incorporated an eighth domain classifier that we now believe is out of scope for this particular work. As the domain-specific classifiers are independent, this does not influence the results of `BoW` or `BERTfrozen`. However, we acknowledge this may

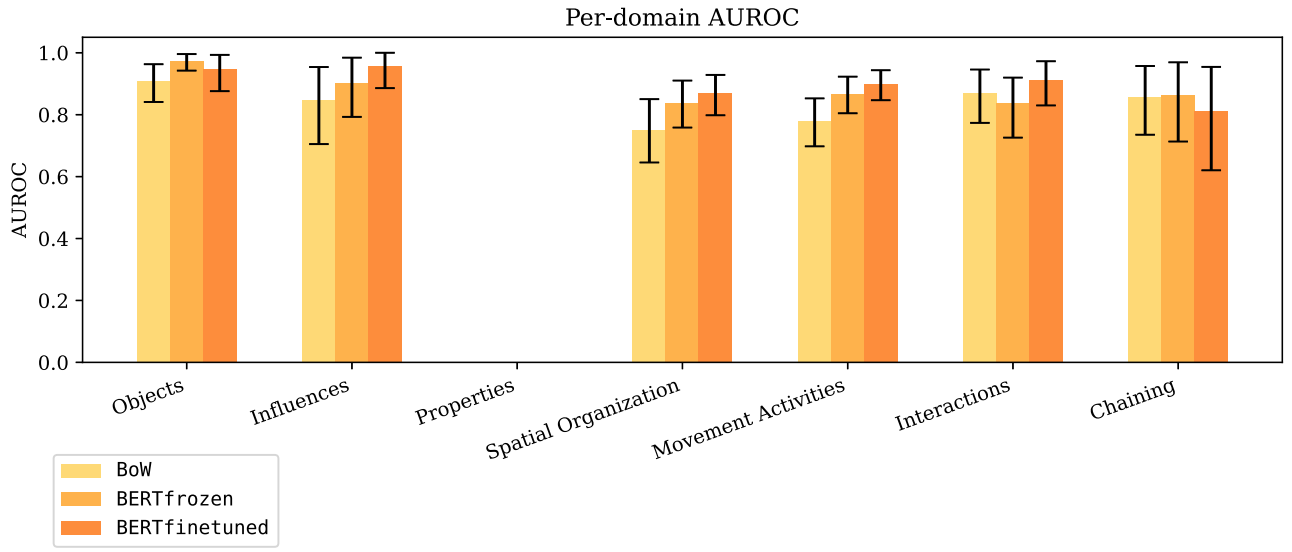


FIG. 3. Model performance across sensemaking domains. For each domain, we show the average AUROC and 95% confidence interval from our 1000 bootstrapped test samples. Here, we see that `BERTfrozen` has the best performance in two domains (objects and chaining), and `BERTfinetuned` has a higher average AUROC in the other 5 (other than properties). We observe that all three models are quite good at detecting evidence that a student has sensemade an object, but do worse on domains with less data. The properties domain only has eight positive examples in the entire dataset, which were all used in training. As such, we are unable to report the AUROC for the properties domain.

impact our reported performance of `BERTfinetuned` as the embeddings across classifiers are shared. We chose not to retrain as the model resources required to do another hyperparameter optimization (another set of 324 HPC experiments) is high. Retraining without the eighth domain would likely only increase the performance of `BERTfinetuned` on seven key domains, and, as we show below, it is already performing better than other variants.

E. Results from initial validation phase

The best model for `BoW` achieved an average AUROC score of 0.839 across validation folds with the hyperparameters: $\alpha = 0.01, \lambda_1 = 0.001$. Similarly, for `BERTfrozen`, the best model achieved an AUROC score of 0.890 with the hyperparameters: $\alpha = 0.001, \lambda_1 = 0.001$. For `BERTfinetuned`, after running 324 fine-tuning experiments on Tufts' HPC services, we picked a model with final AUROC of 0.916 on the validation set, which had the following hyperparameters: $\alpha = 1 \times 10^{-5}$, $\lambda_1 = 0.0001$, and $\lambda_2 = 0.1$.

F. Results on held-out test data

We retrain our selected models (with model-specific optimal hyperparameter settings) on all training and validation data. We then evaluate on the held-out test set of 60 responses from 15 randomly selected students. In Fig. 3, we show the average AUROC across independent test predictions for each of the seven domain classifiers and for each model type. To create approximate uncertainty intervals, we use bootstrap resampling of our test set [52,53] to generate 1000 same-sized test sets that represent plausible unseen student responses, and compute the AUROC of each one. Intervals in Fig. 3 depict the 2.5–97.5th percentile range of these bootstrapped AUROCs.

Beyond the per-domain results in Fig. 3, we show the overall average AUROC for each model in Table I. This microaverage is computed by weighting each annotation label equally across all domains. As such, these values represent the annotation quality of the multitask classification setup as a whole for each model. In Table I, we also report the time taken to train the final selected model. To assess the significance of differences between models, we provide the 95% confidence interval in Table I in the *difference* in AUROC between models, computed by taking percentiles from our bootstrapping procedure. These intervals can be interpreted as indicating the uncertainty in the AUROC difference across models. When comparing models A and B, if the interval is all positive and an exact zero is outside of the interval, we can be 95% confident that there exists an actual positive difference between the models, and model A is indeed better than model B.

`BoW` performs decently overall with an AUROC score of 0.839 and trains efficiently in 1 min. For the objects domain in particular, it performs nearly as well as the other two models. `BERTfrozen` performs slightly better with an AUROC score of 0.887 and takes 7 min to train. It performs the best for two domains: objects and chaining. It is able to reliably outperform `BoW` on average, as zero is not within the confidence interval for the AUROC difference reported in Table I. (The interval containing zero would indicate that no significant difference between models is plausible.)

TABLE I. Average model performance and comparison. Each training job for a BoW or BERTfrozen model was run on 4 CPU cores, while each BERTfinetuned job was run with 1 GPU (Nvidia A100) and 4 CPU cores.

	AUROC (95% CI)	Train time (min)
BoW	0.839 (0.803, 0.873)	1
BERTfrozen	0.887 (0.860, 0.913)	7
BERTfinetuned	0.916 (0.893, 0.938)	160

	Average (95% CI)
$\Delta\text{AUROC}_{\text{Finetuned}-\text{Frozen}}$	0.029 (0.009, 0.049)
$\Delta\text{AUROC}_{\text{Finetuned}-\text{BoW}}$	0.077 (0.052, 0.104)
$\Delta\text{AUROC}_{\text{Frozen}-\text{BoW}}$	0.048 (0.021, 0.075)

BERTfinetuned achieves an AUROC score of 0.916. It performs the best for influences, spatial organization, movement activities, and interactions. It takes significant time to train, 160 min. That time might not be prohibitive, depending on the time constraints of the user; training is only needed when new student data are available. An additional factor to consider is the amount of computational resources. BERTfinetuned requires training on GPU hardware, which may be more challenging to access. The decision to train one model over another ultimately depends on the time, computational resources available, and the extent to which one seeks moderately improved performance.

One major limitation in the results is that we cannot report AUROC results for yes-or-no criteria labels in the properties domain in Fig. 3. In our dataset, class imbalance is extremely large for this domain, and no positive examples were included in the test set, making AUROC undefined. More data, especially more positive examples, are necessary for improvement.

V. BENEFITS AND LIMITATIONS

A. Utility

As we described in the introduction, we are working to automate the measurement of mechanistic sensemaking *evidence* for research purposes across a set of student responses to conceptual problems in introductory Newtonian mechanics. This will allow larger-scale research studies than are possible with human annotation alone. Currently, annotating problems by hand is burdensome. Our annotator required about 5 h to apply our measurement scheme to ∼100 student responses for one problem. In contrast, once trained, the processing time for our models to annotate the same amount of data is less than 1 min. Furthermore, training a model to perform this task is doable in less than 3 h. We hypothesize that with much more labeled training data across many additional problems, the model proposed here could perform well on unseen problem instances in introductory mechanics. However, the amount of data needed to cross such a threshold is unclear. Regardless, initial efforts to improve generalization across problems while reducing the required annotation time should be directed toward collaborative annotation as well as semisupervised learning methods [54].

Training models to measure mechanistic sensemaking on more data from new problems opens the door for interesting studies that have been difficult for PER. Consider this example hypothesis: "Evidence of mechanistic sensemaking early in a course is better than early canonical correctness at predicting both sensemaking and correct understanding at the end of the course." Testing that hypothesis would involve many hours of human analysis; the ML we have begun to develop would be able to carry out this investigation quickly and efficiently. ML-enabled measurement could be used to conduct a randomized controlled experiment in PER, such as the one in Kuo *et al.* [55], but specifically for understanding the variables that lead to more mechanistic sensemaking in students. It could also be used to evaluate various introductory physics problems as being of low or high quality depending on whether a large enough body of students displayed (or did not display) evidence of mechanistic sensemaking in their responses (e.g., problem 1 in our dataset).

B. Generalizability

Russ *et al.* [22] developed a framework for assessing mechanistic reasoning that has been in use in PER for almost 20 years. To develop ML, we adapt the categories based on a particular dataset to produce a scheme with seven domains. Researchers with similar datasets within the same conceptual domain of classical mechanics may find the scheme and the ML it supports useful as is. New contexts, however, may require new projections of the framework of Russ *et al*. For example, evidence of mechanistic reasoning in student explanations of electricity and magnetism or quantum mechanics will look different. For example, it is likely that *analogies* would be an essential domain for both of these subject areas.

C. Reliability

By designing and defining the seven domains for our dataset, and providing annotated examples, we hoped to reduce ambiguity for human annotators—those who might look to reproduce results on our data or to apply the scheme to their own. To measure this, we conducted an interrater reliability (IRR) test. Two annotators who took smaller roles in the development of the domains, one professor of computer science and one senior undergraduate student in computer science, used the scheme to measure the last 15 student responses across the 4 physics problems (60 student responses total). The first author provided them with a codebook containing the domain definitions, the four problems, the criteria vectors for each problem, and five

TABLE II. Prediscussion.

	Average accuracy	Average Cohen's kappa
Problem 1	1.0	1.0
Problem 2	0.86	0.66
Problem 3	0.82	0.63
Problem 4	0.83	0.48

annotated student examples per problem with some explicit reasons as to why certain coding decisions were made. Annotators provided initial ratings, then engaged in a collaborative discussion to resolve ambiguity and clarify disagreement. After this discussion, new ratings were reassessed.

Among the three annotators (two new raters alongside the first author who conducted all manual annotation for this work), the average accuracy and Cohen's kappa values [56] achieved prediscussion are displayed in Table II. The values that emerged post a discussion are presented in Table III.

Problem 1 showed perfect agreement, but very few of the student examples show evidence of mechanistic sensemaking. (This suggests a form of utility for instruction: perhaps this is not a useful problem to assign.) For the remaining three problems, IRR scores were reasonably high (>0.8 accuracy). Cohen's kappa values incorporate chance due to data imbalance and are lower across problems, showing moderate to substantial agreement according to the categories in [56].

Discussion improved scores to ≥ 0.88 accuracy, and all Cohen's kappa values increased to indicate substantial and almost perfect agreement [56]. The postdiscussion scores should not be interpreted as quantitative evidence that the coders independently achieved agreement. Rather, this value reflects how well the coders were able to collaboratively resolve ambiguities in the original codebook and reach consensus on the labels. Following the discussion, the codebook language was updated to improve its clarity. Due to time constraints, the coders did not conduct another round of annotation and compute Cohen's kappa values that would indicate how well the coders could achieve independent agreement. We recognize this as a limitation of our IRR investigation, and we encourage future IRR studies of the tool to further validate it.

For all results in Sec. IV, we trained ML to reproduce the labels provided by our single primary annotator. The 60 annotations from 2 other raters studied here were not included in our ML model training, as the IRR procedure was conducted in response to helpful reviewer feedback on the draft post model training based on the first author's coding alone. Thus, this IRR investigation does not influence the interpretation of the ML results. We highlight that this is a limitation of the current work; we do not know how our ML model will perform on measurements from multiple annotators, nor do we have a model that corrects for mislabeling.

TABLE III. Postdiscussion.

	Average accuracy	Average Cohen's kappa
Problem 1	1.0	1.0
Problem 2	0.90	0.79
Problem 3	0.94	0.88
Problem 4	0.88	0.69

Investigating these directions is important for future larger-scale studies. Once an ML model is trained, the same student response will always receive the same annotation. Prediction is fully deterministic.

D. Sustainability and privacy

As ML research advances, new encoders may be able to improve the overall classification performance. However, many of these models require HPC access and large amounts of energy resources for training [57]. As language modeling in ML improves, it is worth benchmarking new methods while also taking into account the amount of compute resources required to obtain only modestly better performance. We only consider models that are open-access, which allows us to train classifiers for our specific task and does not require sharing student data with a third party.

VI. CONCLUSION AND FUTURE WORK

In this work, we contribute a PER-ML measurement scheme to quantify evidence of mechanistic sensemaking in student responses to introductory mechanics problems. By introducing an ML implementation, our scheme can be more efficiently applied to many student responses on many problems than manual coding. This can enable large-scale quantitative research studies in PER, though this technology has both promises and limitations (see Sec. V) and is not a "magic bullet."

In our work to date, we studied three model variants with different language encoders that require different levels of computational resources. We conducted a comparative analysis of real student responses. Results show that the ML models have good average performance on data from new students (0.916 AUROC for the best model). Performance does differ across domains of the scheme, in part reflecting the variance in the amount of available data per domain.

Throughout the research process, we identified new open questions and challenges. We thus conclude with reflections that we hope will motivate further research.

Recall that we describe two steps of refinement from the prior framework of mechanistic reasoning [22]. The first was to design a scheme for *measuring* evidence of mechanistic reasoning in the context of student explanations in introductory mechanics. The second step was to train ML based on human-annotated measurements for 100 students across four mechanics problems. Reflections below consider these steps in reverse order.

(1) *How can the ML implementation of this measurement scheme be improved?* A natural first reaction to this question is *more data*. We hypothesize that with more labeled training data across new problems, the supervised learning models proposed here will be able to achieve better performance on the current task *and* generalize to unseen problems. Given collaborative annotation efforts, such a model could be realized more efficiently and would save all users time in the long run. This also means that our model performance will depend on strong interrater agreement and accounting for human error in labeling. Considering future methods to account for different types of label *noise* is an interesting direction for future work.

There are a few paths that we could take to improve performance. The first is to experiment with newer encoders from the rapidly advancing state-of-the-art frontier [58]. We could use the LLaMa encoder [2] previously used by Fussell *et al.* [8]. For an even newer encoder, we could explore the open-access, open-source 7-billion parameter Mistral model [36]. As off-the-shelf language models improve, one might not need as much task-specific training data to obtain quality performance, though it is hard to imagine doing well with less than our 4-problem, 100-student dataset. Task-specific training or in-context learning [59] will always be required. Out-of-the-box language encoders or encoder-decoder models (e.g., ChatGPT) are trained on general corpora of language, whereas the mechanistic sensemaking of interest in PER reflects more particular disciplinary practices.

Another approach to improve the ML would be to leverage the dependency structure across our domains. As a brief intuitive example, consider the perspective of an annotator. If the annotator identifies that the student response did not discuss even the central object in the object's domain, it is essentially almost certain that properties, spatial organizations, and movement activities are also not provided. This correlation between domains could be exploited for better modeling and may provide valuable inductive bias for achieving quality performance with less labeled data.

Furthermore, it is important that we consider the question of which metrics for the ML performance are most useful for our specific context. In this work, we used AUROC as our primary metric for the reasons outlined in Sec. IV, and we chose to include a few secondary metrics (e.g., AUPRC, confusion matrices, and Cohen's kappa values) in Appendixes E and F that are typically used in ML and education research practices but seem less relevant here. Understanding the differences between these metrics and their benefits and limitations within a specific context remains an ongoing effort within ML research, as shown in recent work that attempts to clearly delineate situations that favor AUROC instead of AUPRC and vice versa [60]. Other recent work at the intersection of Physics Ed. and ML has proposed new metrics for evaluating the trustworthiness of logistic regression ML models for linguistic coding tasks [9]. In future work, we aim to analyze the metrics in Ref. [9] and compare them to the standard metrics mentioned above so that we better understand the set of metrics that can provide us with maximal insight into the quality of our measurement scheme.

Finally, we can pursue semisupervised learning methods [54], which optimize model outputs to match labels that we do have, while also learning from patterns in a large, easier-to-collect unlabeled dataset (in our case, student response text without criteria vector annotations). We believe each of these paths forward is a potentially valuable direction for improvement.

(2) *How can new useful measurement schemes be achieved?* We set out—and remain interested—to assess student reasoning for sensemaking, particularly sensemaking relevant to the disciplinary practices of physics. For the present effort, we limited our attention to mechanistic sensemaking, drawing on prior work in PER and working to connect it to progress in ML. We limited our attention further to introductory mechanics. Other areas, such as electricity and magnetism or quantum mechanics, would require different measurement schemes for evidence of mechanistic sensemaking. For example, it is likely that *analogies* would be an essential domain for both of these subject areas, which we did not include in our scheme. We are not prepared to hypothesize that ML could create such schemes, but it would be worth exploring. There are also many other approaches to explore for incorporating insight from PER and ML research. There are other bases in the literature regarding evidence of sensemaking, broader than mechanistic. There are also other data modalities (e.g., hand-written solutions with diagrams, graphs, and sketches; audio and visual recordings). These choices will influence our ML decisions, and vice versa. Depending on the ML constraints (e.g., complexity and interpretability of the model, and the amount of data and compute resources needed), frameworks from the PER literature will need to be adapted. Thus, as shown in this work, we put forth that the research process that integrates PER and ML is *interactive*. Ideas from the PER and ML communities will influence one another until *convergence*, which is why we aim to encourage collaborations between longtime members of the PER community and ML researchers who are interested in bringing their domain expertise to advancing the mission of PER. It is through such collaborations that we can develop meaningful measurement schemes that take advantage of the efficiency promised by ML and unlock new insights into understanding student thinking.

ACKNOWLEDGMENTS

We are grateful for financial support from the U.S. National Science Foundation via awards GCR 1148 No. 2428640 and CAREER No. 2338962. We wish to acknowledge other members of the Hughes Lab research group, namely Ethan Harvey and Cynthia Feeney, for thoughtful ML discussions. We would also like to

acknowledge nonlab members Gabriel and Ken, who contributed invaluable resources to this work.

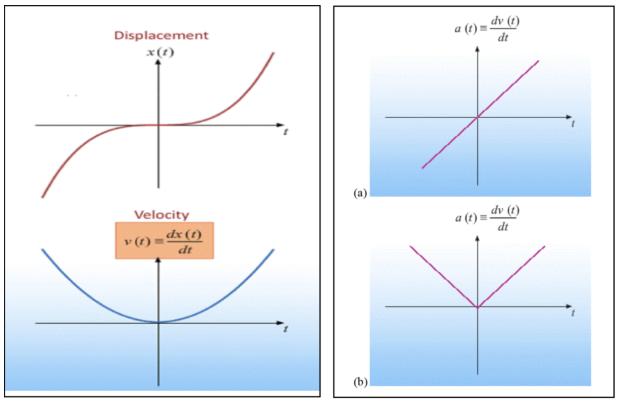
DATA AVAILABILITY

The data that support the findings of this article are openly available [25], embargo periods may apply.

APPENDIX A: PHYSICS PROBLEMS WITH CRITERIA QUESTIONS

We provide all four physics problems as they originally appeared to students from Ref. [23]. For each problem, we show the specific annotation scheme—i.e., the specific criteria questions for each problem.

Problem 1



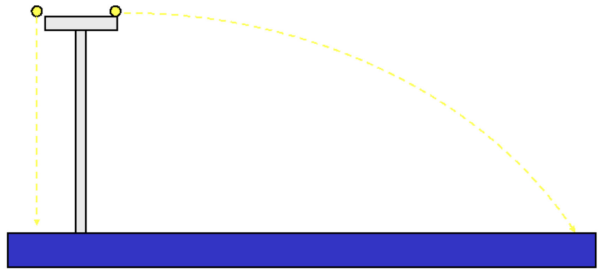
1. Which plot best represents the acceleration curve associated with the displacement and velocity curves shown here?

2. Briefly explain your answer to the previous question.

Annotation Procedure 1

(1) Does the student identify the object (object, car, etc.)?
(2) Does the student identify the object orientation (e.g., left or right)?
(3) Does the student identify movement 1 of the object(e.g., slow down)?
(4) Does the student identify movement 2 of the object (e.g., zero)?
(5) Does the student identify movement 3 of the object (e.g., speed up)?

Problem 2



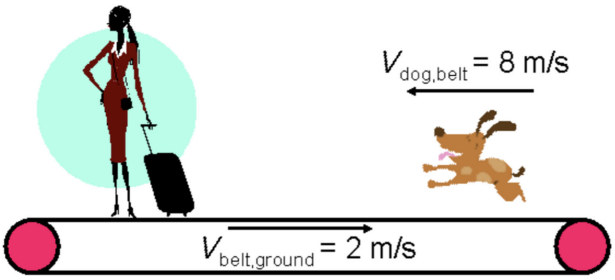
A physics demo launches a ball horizontally while dropping a second ball vertically at exactly the same time.
1. Which ball hits the ground first?
(a) Dropped ball
(b) Launched (horizontally) ball
(c) Both hit at the same time

2. Briefly explain your answer to the previous question.

Annotation Procedure 2

(1) Does the student identify object 1 (e.g., ball 1)?
(2) Does the student identify object 2 (e.g., ball 2)?
(3) Does the student identify the possible environmental influences on object 1 (e.g., gravity)?
(4) Does the student identify the possible environmental influences on object 2 (e.g., gravity)?
(5) Does the student identify object 1 spatial organization (e.g., down)?
(6) Does the student identify object 2 spatial organization (e.g., horizontal up)?
(7) Does the student identify object 1 motion post the drop (e.g., speeding up)?
(8) Does the student identify object 2 motion post the drop (e.g., speeding up)?
(9) Does the student identify the interaction between object 1 and gravity (e.g., gravitational force)?
(10) Does the student identify the interaction between object 2 and gravity (e.g., gravitational force)?
(11) Does the student identify the mechanistic relationship between gravity and a change in motion (e.g., Earth's gravitational force is a mechanism that causes the object to move down at a constantly increasing speed)?

Problem 3



A girls stands on a moving sidewalk (conveyor belt) that is moving to the right at a speed of 2 m/s relative to the ground. A dog runs on the belt toward the girl at a speed of 8 m/s/ relative to the belt.
1. What is the speed of the dog relative to the girl?
(a) 10 m/s
(b) 6 m/s
(c) 8 m/s

Annotation Procedure 3

(1) Does the student identify object 1 (e.g., girl)?
(2) Does the student identify object 2 (e.g., dog)?
(3) Does the student identify object 3 (e.g., moving sidewalk, conveyor belt)?
(4) Does the student identify the spatial organization of the dog (e.g., left)?
(5) Does the student identify the spatial organization of the belt (e.g., right)?
(6) Does the student identify the movement of the girl (e.g., at rest)?
(7) Does the student identify the movement of the dog (e.g., 8 m/s constant speed)?
(8) Does the student identify the movement of the belt (e.g., 2 m/s constant speed)?
(9) Does the student identify the interaction between the girl and the belt (e.g., moving her to the right)?
(10) Does the student identify the interaction between the dog and the belt (e.g., moving the dog to the right)?
(11) Does the student identify the mechanistic relationship between motion of the observer and motion of the observed? (e.g., the observer moving relative to the object impacts the observed motion of the object)

Problem 4

1. The net force on a box is in the positive x direction. Which of the following statements best describes the motion of the box?
(a) Its acceleration is parallel to the x axis.
(b) Its velocity is parallel to the x axis.
(c) Both its velocity and acceleration are parallel to the x axis.
(d) Neither its velocity not its acceleration need to be parallel to the x axis.

Annotation Procedure 4

(1) Does the student identify the object (e.g., object, box, etc.)?
(2) Does the student identify the possible environmental influence acting on the object (e.g., applied net force)?
(3) Does the student identify the object properties (e.g., mass)?
(4) Does the student identify the object spatial organization before applied force (e.g., upwards)?
(5) Does the student identify the object spatial organization after applied force (e.g., horizontal)?
(6) Does the student identify the object movement prior to the applied force (e.g., constant velocity)?
(7) Does the student identify the object movement after the applied force (e.g., final velocity)?
(8) Does the student identify the interaction between the object and the net applied force (e.g., change in direction)?
(9) Does the student identify the mechanistic relationship between an applied net force and a change in motion? (e.g., a net force applied in the x direction will be a mechanism that causes a change in motion/acceleration in the x direction)

APPENDIX B: ANNOTATION TABLES

We provide an annotation summary. We report the number of student responses per problem in Table IV, and the number of those explanations that are labeled as positive for each specific criterion in Table V.

TABLE IV. The number of student responses per problem.

Problem	Number of responses
1	89
2	98
3	100
4	98

TABLE V. The number of positive criteria labels per problem.

Criteria	Problem 1	Problem 2	Problem 3	Problem 4
1	9	89	97	66
2	4	89	96	62
3	6	49	78	8
4	1	48	36	21
5	4	40	9	46
6		40	80	20
7		46	92	44
8		46	29	41
9		21	54	42
10		20	36	
11		18	13	

APPENDIX C: LOSS FUNCTION DEFINITION

Given the training dataset $D_{\text{Train}} = \{\mathbf{x}^n, \mathbf{y}^n\}_{n=1}^N$ of assumed independent student embeddings with human-annotated labels, we train all parameters Θ to minimize the following loss function:

$$\begin{aligned} \mathcal{L}(\Theta) &= \mathcal{L}^{\text{BCE}}(\Theta) + \mathcal{R}(\Theta), \\ \mathcal{L}^{\text{BCE}}(\Theta) &= \sum_{p=1}^{P} \sum_{i=1}^{N_p} \sum_{d=1}^{D=7} \sum_{c \in C_d^p} \frac{\ell_{c,d}^{i,p}}{DPN_p \max(1, |C_d^p|)}, \\ \ell_{c,d}^{i,p} &= -y_{c,d}^{i,p} \log(\rho_{c,d}^{i,p}) - (1 - y_{c,d}^{i,p}) \log(1 - \rho_{c,d}^{i,p}). \end{aligned} \tag{C1}$$

Here, P is the total number of problems, N_p is the total number of student responses per problem in the dataset D_{Train}, $D = 7$ is the total number of domains, and C_d^p is the set of all context criteria (indexed by c) in a particular domain d for a specific problem p. In the binary cross-entropy expression, $y_{c,d}^{i,p}$ is the ground truth prediction for the i^{th} student response associated with problem p with respect to domain d and specific criteria $c \cdot \rho_{c,d}^{i,p}$ is the corresponding predictive probability. Note that by *ground truth*, we mean the human-annotated labels provided in this work.

The loss representing fit to the training data is averaged over all examples. The overall loss also includes a regularization term $\mathcal{R}(\Theta)$ over all weights. When we train `BERTfinetuned`, we have both a regularization term in the loss for classifier weights $\{\mathbf{W}_d, b_d\}_{d=1}^D$ and the encoder weights of *BERT* $\{\boldsymbol{\theta}\}$. We use an L2SP loss [49,50], which encourages weights for the LR classifiers to be close to zero, while encouraging BERT encoder parameters to be near pretrained values.

APPENDIX D: HYPERPARAMETER STRATEGY

We outline our hyperparameter selection procedure so all steps can be reproduced. For all experiments, we set the following seeds to control pseudorandomness: 123 when splitting the data, 360 when running experiments, and 126 when bootstrapping test sets.

`BoW` and `BERTfrozen`. We randomly select 15 students for our held-out test set (60 explanations) and use the rest of the data for a randomized fivefold cross-validation procedure, where the folds are split by student ID. Splitting by student ID ensures that there is no data leakage—i.e., we do not want the model learning student-specific correlations. As such, we ensure that all responses from a specific student are in the same fold. All splits are provided in our GitHub Repo "AuSeM."

We perform a grid hyperparameter search while optimizing for the AUROC score over the following: learning rates $\alpha = \{0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000005, 0.000001\}$, and weight decays $\lambda_1 = \{0.1, 0.01, 0.001, 0.0001, 1 \times 10^{-5}, 0\}$. We average the value of AUROC from the last epoch from all 5 folds and pick the hyperparameter combination with the best AUROC score. The best model for `BoW` achieved an AUROC score of 0.875 with the hyperparameters: $\alpha = 0.01, \lambda_1 = 0.001$. Similarly, for `BERTfrozen`, the best model achieved an AUROC score of 0.880 with the hyperparameters: $\alpha = 0.001$, $\lambda_1 = 0.0001$.

`BERTfinetuned`. We use the same held-out test set, but instead of conducting a cross-validation procedure, which is computationally expensive for fine-tuning, we use the rest of the data as a train and val set for hyperparameter selection. The train and val split is done by subject ID and provided in our GitHub Repo.

We run a hyperparameter grid search with an Adam optimizer for 1000 epochs and batch size to $N = 32$ over the following: learning rate $\alpha = \{0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001, 0.0000005, 1 \times 10^{-7}, 1, 1 \times 10^{-8}\}$, classifier weight decay $\lambda_1 = \{0, 0.1, 0.01, 0.001, 0.0001, 0.00001\}$, and BERT encoder weight decay $\lambda_2 = \{0, 0.1, 0.01, 0.001, 0.0001, 0.00001\}$. We use two weight decay variables such that we can regularize the weights from *BERT* separately from the weights of our LR, as we want to best preserve the information learned in the pretrained model [49,50]. This totals to 324 fine-tuning experiments for `BERTfinetuned` on Tufts' HPC services.

We sort all models by optimal final validation AUROC and select the model that achieves the highest final AUROC $= 0.946$. We observe that the model's behavior appears stable. This model's hyperparameters are $\alpha = 1 \times 10^{-5}$, $\lambda_1 = 0.0001$, and $\lambda_2 = 0.1$. Note that prior to sorting by final AUROC, we sorted by max AUROC to

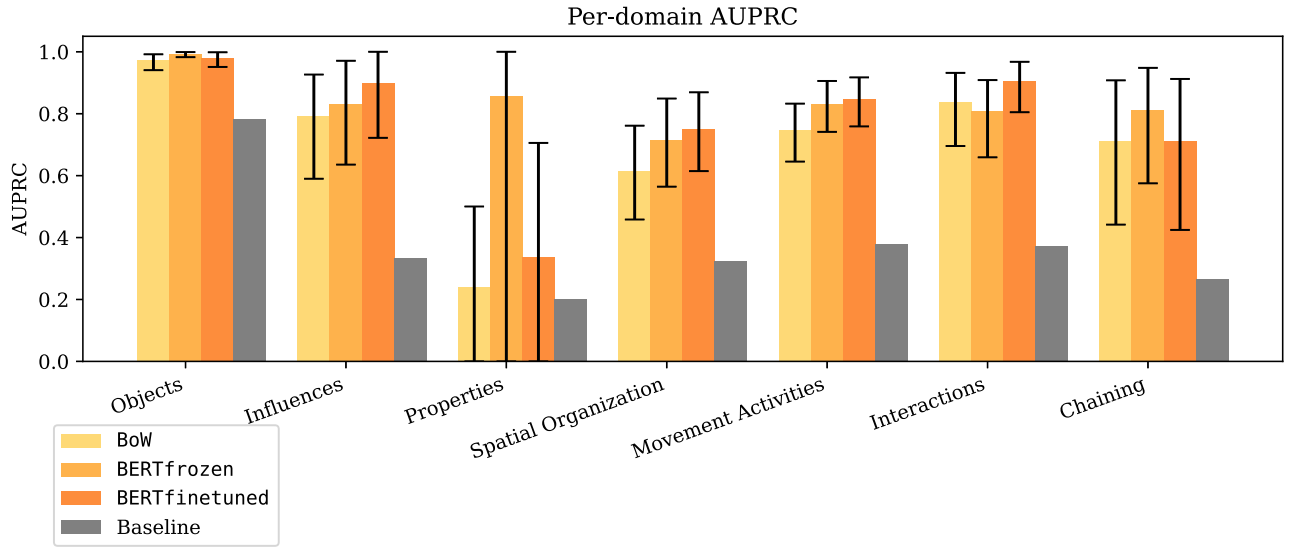


FIG. 4. AUPRC model performance across sensemaking domains. For each domain, we show the average AUPRC and 95% confidence interval from our 1000 bootstrapped test samples. `BERTfrozen` has the best performance in three domains (objects, properties, and chaining), and `BERTfinetuned` has a higher average AUPRC in the other four domains. All three models struggle in domains where there is limited data, especially in the properties domain where zero lies in the 95% confidence interval. In gray, we show the baseline AUPRC value, the fraction of positive examples.

determine whether a reasonable early stopping point could be identified. While some models did have higher intermediate validation AUROC values, this was only the case for unstable models with too-high learning rates. As such, we determined them to be unreliable and did not find a place where early stopping would be useful.

APPENDIX E: AUPRC SCORES

We show the test set AUPRC scores across domains for each classifier in Fig. 4. The microaveraged test set AUPRC score for each model is in Table VI.

APPENDIX F: CONFUSION MATRICES AND COHEN'S KAPPA VALUES

Below, we present for each method a confusion matrix covering all yes-or-no criteria questions for the 15-student test set. To produce binary predictions, we threshold predicted probabilities at 0.5. `BERTfinetuned` has the fewest classification errors overall. However, `BERTfrozen` makes the same number of false positive mistakes (45).

We also report Cohen's kappa values computed from the matrix elements. Cohen's kappa is typically used to measure agreement between annotators, based on the fraction of agreed-upon labels (i.e., observed accuracy) while accounting for independent biases outside of the shared understanding that would lead to agreement [56]. The Cohen's kappa values increase from moderate to substantial agreement from `BoW` to `BERTfrozen` to `BERTfinetuned`. It is important to keep in mind that we chose 0.5 as a standard, arbitrary threshold. Different thresholds could produce higher Cohen's kappa values. For deployment applications where one type of error has a larger cost than another, careful selection of the model and threshold value is important.

TABLE VI. Average model performance and comparison.

	AUPRC (95% CI)	Train time (min)
`BoW`	0.829 (0.786, 0.868)	1
`BERTfrozen`	0.876 (0.842, 0.906)	7
`BERTfinetuned`	0.895 (0.862, 0.926)	160

`BoW` on test set:

Actual\Predicted	0	1
0	249	60
1	66	165

Cohen's kappa: 0.522

`BERTfrozen` on test set:

Actual\Predicted	0	1
0	264	45
1	64	167

Cohen's kappa: 0.583

`BERTfinetuned` on test set:

Actual\Predicted	0	1
0	264	45
1	39	192

Cohen's kappa: 0.641

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv:1810.04805.

[2] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, LLaMa: Open and efficient foundation language models, arXiv:2302.13971.

[3] P. BehnamGhader, V. Adlakha, M. Mosbach, D. Bahdanau, N. Chapados, and S. Reddy, LLM2Vec: Large language models are secretly powerful text encoders, arXiv:2404.05961.

[4] J. Haab, N. Deutschmann, and M. R. Martínez, Is attention interpretation? A quantitative assessment on sets, arXiv:2207.13018.

[5] N. Mylonas, I. Mollas, and G. Tsoumakas, An attention matrix for every decision: Faithfulness-based arbitration among multiple attention-based interpretations of transformers in text classification, arXiv:2209.10876.

[6] *Applying Machine Learning in Science Education Research: When, How, and Why?*, edited by P. Wulff, M. Kubsch, and C. Krist, 1st ed., Springer Texts in Education Vol. 1 (Springer, Cham, 2025), pp. XIII + 369, 10.1007/978-3-031-74227-9.

[7] C. Krist, M. Kubsch, and P. Wulff, Introduction, in *Applying Machine Learning in Science Education Research: When, How, and Why?* (Springer Nature Switzerland, Cham, 2025), pp. 1–12, 10.1007/978-3-031-74227-9_1.

[8] R. K. Fussell, M. Flynn, A. Damle, M. F. J. Fox, and N. G. Holmes, Comparing large language models for supervised analysis of students' lab notes, Phys. Rev. Phys. Educ. Res. **21**, 010128 (2025).

[9] R. K. Fussell, E. M. Stump, and N. G. Holmes, Method to assess the trustworthiness of machine coding at scale, Phys. Rev. Phys. Educ. Res. **20**, 010113 (2024).

[10] T. O. B. Odden and R. S. Russ, Defining sensemaking: Bringing clarity to a fragmented theoretical construct, Sci. Educ. **103**, 187 (2019).

[11] J. Radoff, L. Z. Jaber, and D. Hammer, "It's scary but it's also exciting": Evidence of meta-affective learning in science, Cognit. Instr. **37**, 73 (2019).

[12] T. O. B. Odden and R. S. Russ, Vexing questions that sustain sensemaking, Int. J. Sci. Educ. **41**, 1 (2019).

[13] F. F. Zhao and A. Schuchardt, Development of the sci-math sensemaking framework: Categorizing sensemaking of mathematical equations in science, Int. J. STEM Educ. **8**, 10 (2021).

[14] A. M. L. Phillips, J. Watkins, and D. Hammer, Beyond "asking questions": Problematizing as a disciplinary activity., J. Res. Sci. Teach. **55**, 982 (2018).

[15] A. Sirnoorkar, J. T. Laverty, and P. D. O. Bergeron, Sensemaking and scientific modeling: Intertwined processes analyzed in the context of physics problem solving, arXiv:2207.03939.

[16] T. O. B. Odden, How conceptual blends support sensemaking: A case study from introductory physics, Sci. Educ. **105**, 989 (2021).

[17] B. A. Danielak, A. Gupta, and A. Elby, Marginalized identities of sense-makers: Reframing engineering student retention, J. Eng. Educ. **103**, 8 (2014).

[18] K. H. Hunter, J.-M. G. Rodriguez, and N. M. Becker, Making sense of sensemaking: Using the sensemaking epistemic game to investigate student discourse during a collaborative gas law activity, Chem. Educ. Res. Pract. **22**, 328 (2021).

[19] T. O. B. Odden and R. S. Russ, Sensemaking epistemic game: A model of student sensemaking processes in introductory physics, Phys. Rev. Phys. Educ. Res. **14**, 020122 (2018).

[20] M. J. Ford, A dialogic account of sense-making in scientific argumentation and reasoning, Cognit. Instr. **30**, 207 (2012).

[21] S. Kapon, Unpacking sensemaking, Sci. Educ. **101**, 165 (2017).

[22] R. S. Russ, R. E. Scherr, D. Hammer, and J. Mikeska, Recognizing mechanistic reasoning in student scientific inquiry: A framework for discourse analysis developed from philosophy of science, Sci. Educ. **92**, 499 (2008).

[23] G. G. M. Selen and T. Stelzer, *Classical Mechanics* (FlipItPhysics for University Physics, Macmillan Learning, New York, NY, 2016).

[24] M.-L. M. Ko and M. J. Luna, The glue that makes it "hang together": A framework for identifying how metadiscourse facilitates uncertainty navigation during knowledge building discussions, J. Res. Sci. Teach. **61**, 457(2024).

[25] tufts-ml/AuSeM, GitHub repository, https://github.com/tufts-ml/AuSeM (2025).

[26] A. M. L. Phillips, J. Watkins, and D. Hammer, Problematizing as a scientific endeavor, Phys. Rev. Phys. Educ. Res. **13**, 020107 (2017).

[27] P. Machamer, L. Darden, and C. F. Craver, Thinking about mechanisms, Philos. Sci. **67**, 1 (2000).

[28] F. M. Watts, J. A. Schmidt-McCormack, C. A. Wilhelm, A. Karlin, A. Sattar, B. C. Thompson, A. R. Gere, and G. V. Shultz, What students write about when students write about mechanisms: Analysis of features present in students' written descriptions of an organic reaction mechanism, Chem. Educ. Res. Pract. **21**, 1148 (2020).

[29] F. M. Watts, A. J. Dood, and G. V. Shultz, Developing machine learning models for automated analysis of organic chemistry students' written descriptions of organic reaction mechanisms, in *Student Reasoning in Organic Chemistry* (The Royal Society of Chemistry, London, United Kingdom, 2022).

[30] J. M. Aiken, R. De Bin, H. J. Lewandowski, and M. D. Caballero, Framework for evaluating statistical models in physics education research, Phys. Rev. Phys. Educ. Res. **17**, 020104 (2021).

[31] T. D. Ullmann, Automated analysis of reflection in writing: Validating machine learning approaches, Int. J. Artif. Intell. Educ. **29**, 217 (2019).

[32] C. M. Nakamura, S. K. Murphy, M. G. Christel, S. M. Stevens, and D. A. Zollman, Automated analysis of short responses in an interactive synthetic tutoring system for introductory physics, Phys. Rev. Phys. Educ. Res. **12**, 010122 (2016).

[33] J. Wilson, B. Pollard, J. M. Aiken, M. D. Caballero, and H. J. Lewandowski, Classification of open-ended responses to a research-based assessment using natural language processing, Phys. Rev. Phys. Educ. Res. **18**, 010141 (2022).

[34] P. Tschisgale, P. Wulff, and M. Kubsch, Integrating artificial intelligence-based methods into qualitative research in physics education research: A case for computational grounded theory, Phys. Rev. Phys. Educ. Res. **19**, 020123 (2023).

[35] T. O. B. Odden, A. Marin, and M. D. Caballero, Thematic analysis of 18 years of physics education research conference proceedings using natural language processing, Phys. Rev. Phys. Educ. Res. **16**, 010142 (2020).

[36] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, Mistral 7B, arXiv:2310.06825.

[37] Y. Long, H. Luo, and Y. Zhang, Evaluating large language models in analysing classroom dialogue, npj Sci. Learn. **9**, 60 (2024).

[38] Q. Long, Y. Wu, W. Wang, and S. J. Pan, Does in-context learning really learn? Rethinking how large language models respond and solve tasks via in-context learning, in *First Conference on Language Modeling* (2024), https://openreview.net/forum?id=i2oJjC0ESQ.

[39] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, Rethinking the role of demonstrations: what makes in-context learning work?, in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, edited by Y. Goldberg, Z. Kozareva, and Y. Zhang (Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022), pp. 11048–11064, https://aclanthology.org/2022.emnlp-main.759/.

[40] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, How can we know what language models know?, Trans. Assoc. Comput. Linguist. **8**, 423 (2020).

[41] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, What makes good in-context examples for GPT-3?, in *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures Online*, E. Agirre, M. Apidianaki, and I. Vulić (Association for Computational Linguistics, Dublin, Ireland, 2022), pp. 100–114, https://aclanthology.org/2022.deelio-1.10/.

[42] P. P. Liang, C. Wu, L.-P. Morency, and R. Salakhutdinov, Towards understanding and mitigating social biases in language models, in *Proceedings of the 38th International Conference on Machine Learning* (PMLR, 2021), pp. 6565–6576, https://proceedings.mlr.press/v139/liang21a.html.

[43] W. A. Qader, M. M. Ameen, and B. I. Ahmed, An overview of bag of words; importance, implementation, applications, and challenges, in *Proceedings of the 2019 International Engineering Conference (IEC), Erbil, Iraq, 2019* (2019), pp. 200–204, 10.1109/IEC47844.2019.8950616.

[44] K. P. Murphy, *Probabilistic Machine Learning: An Introduction* (MIT Press, Cambridge, MA, 2022), https://probml.github.io/pml-book/book1.html.

[45] R. E. Turner, An introduction to transformers, arXiv:2304.10557.

[46] C. Sun, X. Qiu, Y. Xu, and X. Huang, How to fine-tune BERT for text classification?, arXiv:1905.05583.

[47] E. Gibaja and S. Ventura, Multi-label learning: A review of the state of the art and ongoing research, WIREs Data Min. Knowl. Discov. **4**, 411 (2014).

[48] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA*, edited by Y. Bengio and Y. LeCun (2015), arXiv:1412.6980.

[49] C. Chelba and A. Acero, Adaptation of maximum entropy capitalizer: Little data can help a lot, Comput. Speech Lang. **20**, 382 (2006).

[50] L. Xuhong, Y. Grandvalet, and F. Davoine, Explicit inductive bias for transfer learning with convolutional networks, in *International Conference on Machine Learning (ICML)* (Proceedings of Machine Learning Research (PMLR), Stockholm, Sweden, 2018).

[51] D. Masters and C. Luschi, Revisiting small batch training for deep neural networks, arXiv:1804.07612.

[52] B. Efron, Bootstrap methods: Another look at the jackknife, in *Breakthroughs in Statistics: Methodology and Distribution* (Springer New York, New York, NY, 1992), pp. 569–593, 10.1007/978-1-4612-4380-9_41.

[53] S. Raschka, Model evaluation, model selection, and algorithm selection in machine learning, arXiv:1811.12808.

[54] J. E. van Engelen and H. H. Hoos, A survey on semi-supervised learning, Mach. Learn. **109**, 373 (2020).

[55] E. Kuo, M. M. Hull, A. Elby, and A. Gupta, Assessing mathematical sensemaking in physics through calculation-concept crossover, Phys. Rev. Phys. Educ. Res. **16**, 020109 (2020).

[56] J. Cohen, A coefficient of agreement for nominal scales, Educ. Psychol. Meas. **20**, 37 (1960).

[57] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, On the dangers of stochastic parrots: Can language models be too big?, in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT* (Association for Computing Machinery, New York, NY, USA, 2021), pp. 610–623, 10.1145/3442188.3445922.

[58] W. X. Zhao *et al.*, A survey of large language models, arXiv:2303.18223.

[59] T. B. Brown *et al.*, Language models are few-shot learners, arXiv:2005.14165.

[60] M. B. McDermott, H. Zhang, L. H. Hansen, G. Angelotti, and J. Gallifant, A closer look at AUROC and AUPRC under class imbalance, in *Proceedings of the 38th Annual Conference on Neural Information Processing Systems* (2024), https://openreview.net/forum?id=S3HvA808gk.