# Towards Cross-Modal Error Detection with Tables and Images

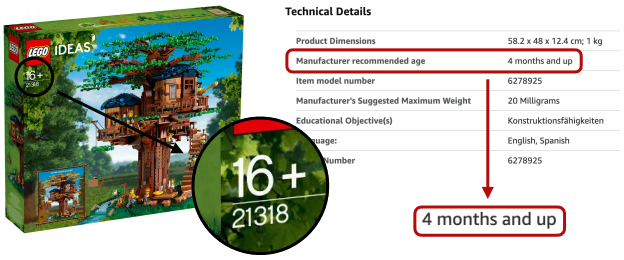
**Olga Ovcharenko** [1] **Sebastian Schelter** [1]

## Abstract

Ensuring data quality at scale remains a persistent challenge for large organizations. Despite recent advances, maintaining accurate and consistent data is still complex, especially when dealing with multiple data modalities. Traditional error detection and correction methods tend to focus on a single modality, typically a table, and often miss cross-modal errors that are common in domains like e-Commerce and healthcare, where image, tabular, and text data co-exist. To address this gap, we take an initial step towards cross-modal error detection in tabular data, by benchmarking several methods. Our evaluation spans four datasets and five baseline approaches. Among them, Cleanlab, a label error detection framework, and DataScope, a data valuation method, perform the best when paired with a strong AutoML framework, achieving the highest F1 scores. Our findings indicate that current methods remain limited, particularly when applied to heavy-tailed real-world data, motivating further research in this area.

## 1. Introduction

Maintaining high-quality data is a challenging task for large organizations and enterprises (Stonebraker et al., 2018; Oala et al., 2023; Abedjan et al., 2016; Singh et al., 2025), especially when a high level of automation is required (Mahdavi et al., 2019; Siddiqi et al., 2023; Yan et al., 2024). Erroneous data can lead to devastating economic, societal, and scientific consequences, especially in combination with machine learning (ML) methods (Sambasivan et al., 2021; McGregor, 2021; Holstein et al., 2019; Northcutt et al., 2021b; Birhane et al., 2021). Consequently, significant resources have been invested into automating data error detection and correction processes, e.g., via data validation systems such as TensorFlow Data Validation (TFDV) (Polyzotis et al., 2019) (deployed at Google) or Amazon Deequ (Schelter et al., 2018) which is used in several AWS services (Nigenda et al., 2022).




https://www.amazon.de/LEGO%C2%AE-Ideas-21318-Baumhaus-Konstruktionsspielzeug/dp/B07PX3WW5N/

*Figure 1.* A real-world cross-modal error from an e-Commerce catalog where a toy with a choking hazard is advertised to children of a wrong age group. The tabular product data erroneously states that the LEGO toy is suitable for four month old babies, even though the product image indicates that it is meant for teenagers with 16+ years of age (accessed May 18, 2025).

**Cross-modal error detection**. However, existing data validation systems often focus on a singular modality only (e.g., relational data) and do not cover scenarios where errors may occur across different modalities. Such cross-modal data errors entail inconsistent information across modalities, while each modality alone appears correct. For instance, cross-modal errors occur in online platforms for e-Commerce, traveling, or real estate, as well as self-driving vehicles and electronic health records, where multi-modal data combines tabular data, text, images, and videos.

*Real-world example*. To showcase a cross-modal data error from the e-Commerce domain, we refer to Figure 1, where a LEGO toy on Amazon is assigned to children of the wrong age group. The tabular product data erroneously states that the LEGO toy is suitable for four-month-old babies, even though the product image indicates that it is meant for 16+ years old teenagers. This data error is potentially dangerous since a toy with such small parts can pose a choking hazard to young children. We point to more cross-modal data errors from Amazon in Figure 2, and would like to highlight that it took us only a couple of minutes to manually find such cases, even though Amazon deploys highly sophisticated methods to manage product attributes (Lin et al., 2021).

[1]BIFOLD & TU Berlin. Correspondence to: Olga Ovcharenko <ovcharenko@tu-berlin.de>, Sebastian Schelter <schelter@tu-berlin.de>.

*Workshop on Unifying Data Curation Frameworks Across Domains (DataWorld)* at ICML'25. 


*Figure 2.* Additional real-world examples of cross-modal errors in Amazon's tabular product metadata, which are obvious from the corresponding product images: 1 A television is misleadingly labeled as 4K ready when the product image shows that its resolution is too low for 4K; 2 A pink toilet seat is listed as having color "white"; 3 A movie is marked as not rated, even though its cover clearly indicates that it is meant for adult audiences only; (product pages accessed on May 10, 2025).

**Practical challenges**. To expand our understanding of this area, we interviewed the content quality team of a large e-Commerce platform. Their product catalog contains tens of millions of products with thousands of distinct attributes. The majority of products originate from several thousand external sellers who provide images and tabular metadata of varying quality and quantity. Moreover, the data from external sellers is combined with additional data sourced from third-party data providers (Yang et al., 2022; Hou et al., 2024). The company's quality team designs custom large language model (LLM)-based solutions for cross-modal error detection and correction of a few selected attributes. However, the high customization requirements and training/inference cost of the attribute-tuned models make their current solution expensive and difficult to scale to more specialized attributes and products.

**Are existing error detection approaches sufficient?** The detection and repair of errors across diverse data modalities represent an emerging direction in data-centric AI research, and general techniques that work across multiple domains are required. Several existing approaches can be applied to multi-modal error detection, yet it is an open question whether they provide sufficient performance since the majority of them have not been explicitly designed for the cross-modal setting.

Single-modal tabular error detection approaches (Mahdavi et al., 2019; Heidari et al., 2019; Krishnan et al., 2016) identify inconsistencies between columns in a table but have no direct means to incorporate cross-modality information. Label error detection methods (Northcutt et al., 2021a;b) focus primarily on label information in predictive settings and would be costly to apply in our scenario since training a specialized model per table attribute is required. LLM-based approaches to error detection (Hua et al., 2024; Singh et al., 2025) show promising results for text-image pairs, but it is unclear how effective they are in handling tabular data combined with visual information. Additionally, LLM approaches suffer from high computational costs and typically require multiple expensive calls to identify erroneous data, which limits their practical application in large-scale data cleaning scenarios. We discuss the mentioned approaches in more detail in Section 2.2.

**Overview and contributions**. The goal of this paper is to introduce the problem of cross-modal error detection in tabular data and to motivate its high practical importance. Our detailed contributions include:

- We motivate and introduce the problem of cross-modal error detection in tabular data (Sections 1 & 2).
- We design a preliminary benchmark with four e-Commerce datasets to evaluate five state-of-the-art error detection methods. According to our findings, approaches designed for label error detection, such as Cleanlab (Northcutt et al., 2021a) and DataScope (Karlaš et al., 2023) (combined with the AutoML framework AutoGluon (Tang et al., 2024)), perform the best. Crucially, in the majority of cases, leveraging both tabular and image data is key to uncovering cross-modal errors. Nonetheless, current methods remain limited, particularly when applied to heavy-tailed real-world data. (Section 3).
- The benchmark containing our data and code is available at: `https://github.com/OlgaOvcharenko/find_errors`

## 2. Problem Statement

In this section, we formalize the problem of cross-modal error detection in tables and review related work.

### 2.1. Cross-Modal Error Detection in Tables

Given aligned multi-modal data $(D, I)$, where $D$ is a relational table and $I$ is a set of corresponding images, the goal is to identify erroneous entries in the relational data $D$. Following Heidari et al. (2019), we denote $A = A_1, A_2, \ldots, A_N$ the attributes of $D$. We consider $D$ to be a set of tuples, where each tuple $t \in D$ consists of cells $C_t = \{t[A_1], t[A_2], \ldots, t[A_N]\}$. Moreover, $t[A_k]$ denotes the value of attribute $A_k$ for tuple $t$, and the corresponding image for $t$ is $i_t$. We assume that errors appear due to inaccurate cell assignments in $D$. More formally, for a cell $c \in C_t$ we denote by $v_c^*$ its unknown true value and by

$v_c$ its observed value. We define an erroneous tuple $t \in D$ as a tuple with at least one cell $c \in C_t$ where $v_c \neq v_c^*$.

We define cross-modal error detection as deciding whether a tuple $t \in D$ is erroneous, based on its tabular data $C_t$ and corresponding image $i_t$. Importantly, we assume a setup where no labeled examples of erroneous records are available as training data, akin to novelty detection (Pimentel et al., 2014).

### 2.2. Related Work

**Tabular error detection**. Detecting errors in tabular data is a long-standing research problem in the data management community (Chu et al., 2013; Abedjan et al., 2016). HoloDetect (Rekatsinas et al., 2017) uses data augmentation and few-shot learning to detect errors, while HoloClean (Rekatsinas et al., 2017) uses probabilistic inference to find the best repair. Raha (Mahdavi et al., 2019) and Baran (Mahdavi & Abedjan, 2020) leverage an ensemble of existing detectors, rules, and constraints for error detection and correction, respectively. ActiveClean (Krishnan et al., 2016) applies active learning to iteratively repair the data while preserving monotone convergence guarantees.

Tabular error detection methods are designed to catch inconsistencies between columns but do not incorporate cross-modality information and may therefore struggle to detect multi-modal errors.

**Label error detection**. Our problem can also be treated as label error detection, where one or more modalities are used to predict errors in a "label column" derived from the table. For example, given the characteristics of e-Commerce products (from a table) and respective product images, we can use the images as input and one of the columns in the table as a label. A popular approach to label error detection is Cleanlab (Northcutt et al., 2021a;b). Cleanlab leverages confident learning to improve existing models and estimate dataset problems such as erroneous labels, (near) duplicates, and non-IID data. Second, Jäger & Biessmann (2024) proposed to apply conformal learning, a method to quantify and calibrate the uncertainty of ML models, for data cleaning. Third, Data Shapley values (Ghorbani & Zou, 2019; Karlaš et al., 2023; Wang & Jia, 2023) are a data valuation metric to quantify the impact of each training point on a model's predictions, which has been shown to work well for label error detection. Fourth, the LEMoN (Zhang et al., 2024) framework leverages contrastive learning and a CLIP (Radford et al., 2021) model. LEMoN finds the nearest neighbors of a sample on the image manifold and compares them to the neighbors on the textual manifold.

However, label error detection methods are designed for predictive problems and concentrate on label errors, not covering, for instance, multi-column errors where several cells in a tuple contain correlated errors. Moreover, it is costly to treat individual columns as labels as this often requires training a specialized model per column.

**Error detection with large language models**. Recently, LLMs emerged as a powerful tool that can be prompted to detect errors in text, images, and structural data. FineMatch (Hua et al., 2024) introduces a benchmark focusing on mismatch detection and correction in text-image pairs. FineMatch shows the proficiency of visual language models (VLMs), e.g., LLaVA (Liu et al., 2023) and GPT-4V (OpenAI, 2023), in detecting and fixing errors in multi-modal inputs. Versatile Data Cleanser (VDC) (Zhu et al., 2024) is another LLM-powered label error detection framework that consists of three parts: Question generation, answering, and evaluation. Given an image and a textual label, VDC creates LLM-generated label-specific questions that are later answered by the multi-modal LLM based on the image. The visual question-answering and original labels are used to evaluate the correctness of labels. Another LLM-based solution is DataVinci (Singh et al., 2025), which targets detecting and correcting sub-string errors.

While prior work has shown the effectiveness of LLMs and VLMs for text and image cleaning, it is still, to the best of our knowledge, unclear how VLMs handle tabular data and inter-row dependencies combined with visual data. Furthermore, VLMs maybe prohibitively expensive in settings with millions of input samples and hundreds of columns, especially since existing methods require several calls per sample to find erroneous data.

## 3. Preliminary Experimental Results

Next, we conduct a set of preliminary experiments. We aim to show that multiple modalities indeed help to detect cross-modal errors and that existing baseline techniques do not sufficiently address our problem.

### 3.1. Data and Error Generation

**Datasets**. We experiment with four datasets to analyze and demonstrate the difficulty of cross-modal error detection. All datasets contain tabular data and an image for each row in the table. *Fashion* (Iuhaniwal, 2024) and *Fashion 44K* (Aggarwal, 2019) are two similar Kaggle e-Commerce datasets with images and tabular data of clothing products, where *Fashion 44K* is a larger version. The other two datasets, *Baby* and *Sports*, originate from subcategories of the Kaggle e-Commerce image dataset (Calik & Büyükpancar, 2024). They originally contain only images, and, therefore, we leverage a VLM (LLaVA 1.5-7b) (Liu et al., 2023) to generate corresponding tabular data (see prompts in Appendix A.1), which we manually post-process and refine to obtain a ground truth dataset. All datasets con-

tain an e-Commerce product title, category, type, and color attributes. There are also dataset-specific columns, e.g., sport type for the *Sports* dataset. Table A1 and Table A2 in the Appendix highlight the properties of the datasets.

**Error injection**. Inspired by our interview with practitioners, we inject synthetic errors into the test splits of the tabular data for our datasets to simulate cross-modal errors that are hard to detect from one modality alone. Note that none of the training splits contain errors. First, we manually curate the data to remove pre-existing errors by inspecting the data and running Cleanlab with the original non-corrupted tabular data to find and fix inconsistencies. Next, we randomly select 50% of the rows of each test split. For each sampled row of the test data, we select a random column to introduce the error into. To inject an error, we replace the selected cell with a random existing value from the set of column unique values different from the cell's current value. In addition, we modify all cells in the selected row which contain the original cell value. For instance, if we change the color attribute of a product, we also replace the name of the color in the product title if contained. By that, we ensure that the error must be detected from the image and that the original value is not leaked from other cell values. For correlated columns, we replace the original values only with already observed pairs, e.g., to avoid creating non-sensical products with the category "footwear" and subcategory "dress".

### 3.2. Baseline Error Detection Performance

The goal of our first experiment is two-fold: we investigate whether the image modality helps with finding cross-modal errors in tabular data, and we assess the performance of several baselines on our benchmark data.

**Experimental setup**. We evaluate the error detection performance of several baseline approaches from Section 2.2 with different modalities (table only, image only, table + image). For *Fashion* and *Fashion 44K*, we use a random sample of 30% of the tuples as test set, for *Baby* and *Sports*, we use the union of their existing validation and test splits as test set. As discussed, we only introduce synthetic errors into the test data. We measure the performance with precision ($\mathcal{P}$), recall ($\mathcal{R}$), and F1 score ($\mathcal{F}1$).

**Methods**. We evaluate the following methods in our benchmark and refer to Appendix A.2 for details on used prompts.

- **Raha** – a state-of-the-art single-column error detection framework for tabular data (Mahdavi et al., 2019). We use the original benchmarking scripts and provide the framework with clean and dirty data samples. Raha does not support images, therefore, we evaluate Raha only with tabular data.
- **AutoGluon + Cleanlab** – we combine the state-of-the-art AutoML library AutoGluon (Tang et al., 2024), with a state-of-the-art method for label error detection and correction Cleanlab (Northcutt et al., 2021a;b). We repeatedly train AutoGluon models with each column as a target and use the resulting classifiers as input for Cleanlab's error detection. We vary the input modalities for the AutoGluon models from table-only to image-only and table combined with image.
- **AutoGluon + DataScope** – we combine AutoGluon with the DataScope library (Karlaš et al., 2023) to identify erroneous tuples via negative data importance scores. Concretely, we compute Data Shapley values (Ghorbani & Zou, 2019) for the potentially dirty test data, using the clean training data for validation. We configure DataScope to calculate exact Data Shapley values for a kNN proxy model (Jia et al., 2019) with $k = 1$, repeatedly train AutoGluon models with each column as a target and use the resulting feature representations as input for DataScope. We apply DataScope to the erroneous test set and score on clean train data. All instances with negative importance are considered erroreneous.
- **LLaVA** – we prompt the vision language models LLaVA-1.5 7b (Liu et al., 2023) and LLaVA-Next-Interleave 7b (Li et al., 2024) to detect cross-modal errors. For the zero-shot variant, we do not provide any examples, while we add up to ten concrete dataset- and modality-specific examples (image and label/table row) to the prompt for the few-shot variant. We use LLaVA-Next-Interleave for the few-shot setting since LLaVA-1.5 does not support multi-image inference.
- **LEMoN** – to evaluate the potential of contrastive learning for cross-modal error detection, we leverage LEMoN (Zhang et al., 2024) that can handle textual labels only. We create labels from tabular data by serializing each row into a string. The final label is a combination of column names and values, e.g., for *Fashion* data an example label text is *ProductTitle - Nike Men Air Zoom Shoes, Gender - Men, Category - Footwear...*. LEMoN requires hyperparameters like noise type and noise level; we use random noise with a level of 0.4, analogous to the settings in the original paper.

### 3.3. Results and Discussion

**Baseline performance**. We show the performance scores for the baselines in Table 1, together with an indication of the modalities used for the predictions. First, as expected, we observe that the cross-modal errors are hard to detect from the tabular data alone, indicated by the low scores of Raha, a state-of-the-art error detector for tabular data. Second, we see sub-par results for the other methods as well, when they only have access to the tabular data. Third, the image modality helps with error detection, and the $\mathcal{F}1$

| Method | Table used? | Image used? | *Fashion* $\mathcal{P}$ | *Fashion* $\mathcal{R}$ | *Fashion* $\mathcal{F}1$ | *Baby* $\mathcal{P}$ | *Baby* $\mathcal{R}$ | *Baby* $\mathcal{F}1$ | *Sports* $\mathcal{P}$ | *Sports* $\mathcal{R}$ | *Sports* $\mathcal{F}1$ | *Fashion 44K* $\mathcal{P}$ | *Fashion 44K* $\mathcal{R}$ | *Fashion 44K* $\mathcal{F}1$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Raha | ✓ | ✗ | 0.16 | 0.07 | 0.09 | 0.34 | 0.15 | 0.20 | 0.17 | 0.08 | 0.10 | 0.41 | 0.29 | 0.34 |
| AutoGluon + Cleanlab | ✓ | ✗ | 0.62 | 0.41 | 0.49 | 0.48 | 0.48 | 0.48 | 0.46 | 0.57 | 0.51 | 0.68 | 0.29 | 0.41 |
| AutoGluon + DataScope | ✓ | ✗ | 0.37 | 0.78 | 0.50 | 0.55 | 0.74 | 0.63 | 0.42 | 0.66 | 0.52 | 0.37 | 0.81 | 0.51 |
| LLaVA (zero-shot) | ✓ | ✗ | 1.00 | 0.001 | 0.003 | 0.71 | 0.14 | 0.23 | 0.84 | 0.37 | 0.52 | 0.94 | 0.00 | 0.01 |
| LLaVA (few-shot) | ✓ | ✗ | 0.71 | 0.34 | 0.46 | (0.50) | (1.00) | (0.67) | (0.50) | (1.00) | (0.67) | 0.50 | 0.71 | 0.59 |
| AutoGluon + Cleanlab | ✗ | ✓ | 0.71 | 0.94 | 0.81 | 0.65 | 0.66 | 0.66 | 0.61 | 0.70 | 0.65 | 0.56 | 0.92 | 0.70 |
| AutoGluon + DataScope | ✗ | ✓ | 0.82 | 0.96 | **0.89** | 0.81 | 0.96 | 0.88 | 0.67 | 0.74 | 0.70 | 0.53 | 0.96 | 0.68 |
| LLaVA (zero-shot) | ✗ | ✓ | 0.03 | 0.03 | 0.03 | 0.07 | 0.06 | 0.06 | 050 | 0.005 | 0.01 | 0.008 | 0.00 | 0.00 |
| LLaVA-I. (few-shot) | ✗ | ✓ | 0.15 | 0.98 | 0.27 | 0.17 | 0.64 | 0.27 | 0.26 | 0.94 | 0.41 | 0.11 | 0.99 | 0.21 |
| AutoGluon + Cleanlab | ✓ | ✓ | 0.87 | 0.78 | 0.83 | 0.73 | 0.67 | 0.70 | 0.62 | 0.70 | 0.66 | 0.80 | 0.71 | **0.75** |
| AutoGluon + DataScope | ✓ | ✓ | 0.83 | 0.82 | 0.83 | 0.84 | 0.93 | **0.89** | 0.72 | 0.75 | **0.73** | 0.75 | 0.47 | 0.58 |
| LLaVA (zero-shot) | ✓ | ✓ | 0.00 | 0.00 | 0.00 | 1.00 | 0.006 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| LLaVA-I. (few-shot) | ✓ | ✓ | (0.50) | (1.00) | (0.67) | 0.51 | 0.95 | 0.62 | (0.50) | (1.00) | (0.67) | (0.49) | (0.99) | (0.66) |
| LEMoN | ✓ | ✓ | 0.72 | 0.41 | 0.52 | 0.66 | 0.37 | 0.48 | 0.51 | 0.36 | 0.42 | 0.50 | 0.40 | 0.44 |

*Table 1.* Error detection performance of the baseline approaches with varying modalities on three e-commerce different methods. Best result per dataset are highlighted in bold, the second-best underlined. We report some of the metrics for LLaVA-Interleave (LLaVA-I.) (few-shot) in brackets, where it only outputs a single class prediction (marking everything as erroneous). The best-performing methods are AutoGluon + DataScope and AutoGluon + Cleanlab with access to both image and tabular data. Runtimes are included in Table A11.

scores of AutoGluon + Cleanlab and AutoGluon + DataScope in the image-only setup are significantly higher than in the table-only setup. Furthermore, in three out of four datasets, joint access to both modalities results in the best performance, with an improvement of up to 5% in $\mathcal{F}1$ score compared to the image-only setup. However, for *Fashion*, the performance of AutoGluon + DataScope degrades when images are combined with tabular data.

The results confirm our hypothesis that the chosen problem is difficult, even for VLMs. LLaVA in both modes, zero- and few-shot, produces low or unreliable $\mathcal{F}1$ scores and, in several cases, even marks every input tuple as erroneous (shown in brackets). This is surprising since we use LLaVA for the tabular data generation of the *Baby* and *Sports* datasets.

Even though AutoGluon + DataScope scores the highest in our benchmark, it only reaches high $\mathcal{F}1$ for the two small datasets and provides subpar performance on the larger *Fashion44K* dataset, missing up to half of the errors in some cases (as indicated by the recall scores). The inconsistent performance across modalities and datasets raises the question of when tabular data combined with images becomes helpful for DataScope, given that the multi-modal approach has proven effective for other methods. Furthermore, due to the use of a kNN proxy model, DataScope's performance relies heavily on AutoGluon's input embedding quality, with $\mathcal{F}1$ scores decreasing by 10% when using insufficiently trained representations (not shown in the table). Furthermore, DataScope requires a large clean validation set. While our benchmark uses clean training data for scoring, this requirement is problematic in real-world applications where such clean datasets may not be available. AutoGluon + Cleanlab scores the second highest in our benchmark but only reaches $\mathcal{F}1$ scores of around 70%-80% percent, missing 22% to 33% percent of errors (according to the recall scores). Interestingly, we see a consistent positive impact of having joint access to both modalities (table + image) for this baseline, with improvements of up to 5% in F1 scores.

| Erroneous Tuple | | Image |
|---|---|---|
| Category | *Clothing* |  |
| ProductType | *Pants* | |
| Color | Blue pink | |
| Material | Plastic | |
| Category | *Volleyball* |  |
| ProductType | *Volleyball nets* | |
| Color | Black | |
| Sport type | Camping | |

*Table 2.* Examples of two multi-modal errors that are only detected by jointly looking at image and tabular data.

We interpret our results as confirmation that further research is needed for cross-modal error detection where one of the modalities is a table.

*Example*. To give a concrete example of cross-modal errors that are found only by jointly looking at the image and table, we point to Table 2, which describes two examples

| | Dataset | Column | Method | $\mathcal{F}1$ Table | $\mathcal{F}1$ Image | $\mathcal{F}1$ Table + Image | Cardinality | Frequency Distribution |
|---|---|---|---|---|---|---|---|---|
| **Easy** | *Fashion* | Category | Cleanlab | 0.48 | 0.94 | **1.00** | 2 |  |
| | | | DataScope | 0.43 | **1.00** | **1.00** | | |
| | *Fashion* | SubCategory | Cleanlab | 0.61 | **0.94** | **0.94** | 9 |  |
| | | | DataScope | 0.47 | 0.87 | 0.91 | | |
| | *Baby* | PackageMaterial | Cleanlab | 0.70 | 0.89 | 0.90 | 9 |  |
| | | | DataScope | 0.70 | 0.94 | **0.95** | | |
| **Hard** | *Fashion* | Color | Cleanlab | 0.58 | 0.66 | 0.69 | 38 |  |
| | | | DataScope | 0.51 | **0.77** | 0.71 | | |
| | *Sports* | SportType | Cleanlab | 0.59 | 0.59 | 0.62 | 65 |  |
| | | | DataScope | 0.47 | **0.70** | 0.68 | | |
| | *Baby* | ProductType | Cleanlab | 0.43 | 0.42 | 0.51 | 132 |  |
| | | | DataScope | 0.70 | **0.88** | **0.88** | | |

*Table 3.* Selection of easy and hard columns for cross-modal error detection using AutoGluon + Cleanlab and AutoGluon + DataScope. Hard columns have higher number of distinct values and a more skewed frequency distribution.

from the *Baby* and *Sports* datasets. The baby wipes have the wrong product type and category, and are described as baby pants in the table. The camping chair is wrongly marked as a volleyball net. In both cases, the table-only or image-only methods fail to detect these inconsistencies. Both errors are nontrivial. While the "wipes" error happens even during data generation, where LLaVA confuses baby wipes/onesies/diapers, the "chair" error is detected only when given both image and table which we attribute to the fact that the sport type contradicts the category and helps to detect the inconsistency.

**Column-wise performance**. To deeper understand the results of the best-performing methods AutoGluon + DataScope and AutoGluon + Cleanlab, we analyze their performance for errors in different columns. Table 3 shows three easy (high $\mathcal{F}1$ score) and the three hard (low $\mathcal{F}1$ score) cases. Overall, the analysis indicates that it is more difficult to detect errors in high-cardinality columns (many distinct values) with a skewed frequency distribution (many rare values), e.g., the product type column in *Baby* is hard to detect because of a long-tailed distribution with only a few common values. Importantly, real-world data often exhibit skewed/long-tailed distributions (Yi et al., 2025). On the other hand, Table 3 indicates that errors are the easiest to detect in columns with few distinct values and balanced frequencies. An key observation, confirmed by the column-wise evaluation, is that joint access to image and table data improves the $\mathcal{F}1$ scores. However, DataScope struggles to leverage tabular data with images in challenging cases. The optimal approach for combining modalities remains unclear for DataScope, while other methods successfully benefit from multi-modal error detection.

### 3.4. Automated Repair

While this paper focuses on error detection, an important next step is the automated repair of the detected errors. For that, we evaluate AutoGluon + Cleanlab's ability to correct the errors by leveraging the correct "label" as suggested via confident learning. Table 4 shows the error detection and correction accuracy for the selected columns in all three datasets, denoting the fraction of injected errors that could be successfully detected and repaired. Full results are in Appendix, Table A7, Table A8, Table A9, and Table A10. Similar to detection, we observe that error correction benefits from joint access to tabular and image data and that repairs are more difficult for heavy-tailed data (e.g., in color and product type columns from the *Baby* and *Sports* datasets). There is room for improvement in error detection and repair.

| | Dataset | Column | Table | Image | Table+ Image |
|---|---|---|---|---|---|
| **Easy** | *Fashion* | Sub | 0.01 | 0.25 | **0.83** |
| | *Baby* | PackageMat. | 0.79 | 0.18 | **0.81** |
| | *Fashion* | SubCategory | 0.07 | 0.66 | **0.94** |
| **Hard** | *Baby* | ProductType | 0.17 | 0.20 | **0.24** |
| | *Baby* | Color | **0.21** | 0.16 | **0.21** |
| | *Sports* | ProductType | 0.05 | 0.30 | **0.35** |

*Table 4.* A selection of easy and hard columns for error detection and repair using AutoGluon + Cleanlab. The table contains the error correction *accuracy* per column.

## 4. Conclusions

Our preliminary findings highlight a gap in existing methodologies: the absence of an effective approach that can directly handle multi-modal inputs for both error detection and correction. We found tabular error detection methods and VLMs to have subpar performance. While label error detection methods showed potential, they still incur a performance gap, and it is unclear how to select the best approach among them. Furthermore, label error detection methods are expensive, as one has to train a separate AutoML model for each column in every dataset.

To advance cross-modal error detection for tables, we plan to design a dedicated multi-modal model (potentially based on self-supervised contrastive learning (Radford et al., 2021)) and to develop a larger and more comprehensive benchmark that includes real-world data from domains beyond e-Commerce. Additionally, a broader range of baseline models should be considered in the evaluation, e.g., tabular foundation models like TabPFN (Hollmann et al., 2025) and CARTE (Kim et al., 2024), as well as novelty and anomaly detection methods.

## Acknowledgements

The authors thank Sebastian Baunsgaard and Bojan Karlaš for their insightful comments and constructive feedback on the manuscript.

## Impact Statement

This work introduces the problem of cross-modal error detection in tabular data and benchmarks current approaches to advance data preparation and validation. As enterprises increasingly leverage multi-modal datasets—combining structured tables with accompanying text and images—ensuring consistency across modalities becomes vital. However, existing methods focus mainly on single-modality or label errors, often missing cross-modal inconsistencies common in domains like e-Commerce and healthcare. Such inconsistencies can lead to serious real-world consequences, including safety risks for consumers, for example, a product image showing a adults-only product mislabeled as child-safe, or a violent movie tagged as family-friendly. Moreover, undetected errors can result in violations of legal and regulatory standards, such as incorrect specification of allergens, age restrictions, or chemical contents. Our aim is to lay the groundwork for modality-aware data validation pipelines that not only enhance technical robustness but also promote consumer protection and regulatory compliance.

## References

Abedjan, Z., Chu, X., Deng, D., Fernandez, R. C., Ilyas, I. F., Ouzzani, M., Papotti, P., Stonebraker, M., and Tang, N. Detecting data errors: Where are we and what needs to be done? *Proceedings of the VLDB Endowment*, 9(12): 993–1004, 2016.

Aggarwal, P. Fashion product images dataset, 2019. URL https://www.kaggle.com/ds/139630.

Birhane, A., Prabhu, V. U., and Kahembwe, E. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *arXiv preprint arXiv:2110.01963*, 2021.

Calik, F. and Büyükpancar, H. Ecommerce product images 18k, 2024. URL https://www.kaggle.com/datasets/fatihkgg/ecommerce-product-images-18k.

Chu, X., Ilyas, I. F., and Papotti, P. Discovering denial constraints. *Proceedings of the VLDB Endowment*, 6(13): 1498–1509, 2013.

Ghorbani, A. and Zou, J. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.

Heidari, A., McGrath, J., Ilyas, I. F., and Rekatsinas, T. Holodetect: Few-shot learning for error detection. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, pp. 829–846, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450356435. doi: 10.1145/3299869.3319888. URL https://doi.org/10.1145/3299869.3319888.

Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S. B., Schirrmeister, R. T., and Hutter, F. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, Jan 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6. URL https://doi.org/10.1038/s41586-024-08328-6.

Holstein, K., Wortman Vaughan, J., Daumé III, H., Dudik, M., and Wallach, H. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–16, 2019.

Hou, Y., Li, J., He, Z., Yan, A., Chen, X., and McAuley, J. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Hua, H., Shi, J., Kafle, K., Jenni, S., Zhang, D., Collomosse, J., Cohen, S., and Luo, J. Finematch: Aspect-based fine-grained image and text mismatch detection and correction. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part IX*, pp. 474–491, Berlin, Heidelberg,

2024. Springer-Verlag. ISBN 978-3-031-72672-9. doi: 10.1007/978-3-031-72673-6_26. URL https://doi.org/10.1007/978-3-031-72673-6_26.

Iuhaniwal, V. E-commerce product images, 2024. URL https://www.kaggle.com/datasets/vikashrajluhaniwal/fashion-images.

Jäger, S. and Biessmann, F. From data imputation to data cleaning — automated cleaning of tabular data improves downstream predictive performance. In Dasgupta, S., Mandt, S., and Li, Y. (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3394–3402. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/jager24a.html.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Gurel, N. M., Zhang, B. L. C., and Song, C. S. D. Efficient task-specific data valuation for nearest neighbor algorithms. *Proceedings of the VLDB Endowment*, 12(11), 2019.

Karlaš, B., Dao, D., Interlandi, M., Schelter, S., Wu, W., and Zhang, C. Data debugging with shapley importance over machine learning pipelines. In *The Twelfth International Conference on Learning Representations*, 2023.

Kim, M. J., Grinsztajn, L., and Varoquaux, G. Carte: pre-training and transfer for tabular learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Krishnan, S., Wang, J., Wu, E., Franklin, M. J., and Goldberg, K. Activeclean: Interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment*, 9 (12):948–959, 2016.

Li, F., Zhang, R., Zhang, H., Zhang, Y., Li, B., Li, W., Ma, Z., and Li, C. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models, 2024. URL https://arxiv.org/abs/2407.07895.

Lin, R., He, X., Feng, J., Zalmout, N., Liang, Y., Xiong, L., and Dong, X. L. Pam: Understanding product images in cross product category attribute extraction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pp. 3262–3270, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467164. URL https://doi.org/10.1145/3447548.3467164.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Mahdavi, M. and Abedjan, Z. Baran: Effective error correction via a unified context representation and transfer learning. *Proceedings of the VLDB Endowment (PVLDB)*, 13(11):1948–1961, 2020.

Mahdavi, M., Abedjan, Z., Castro Fernandez, R., Madden, S., Ouzzani, M., Stonebraker, M., and Tang, N. Raha: A configuration-free error detection system. In *Proceedings of the 2019 International Conference on Management of Data*, pp. 865–882, 2019.

McGregor, S. Preventing repeated real world ai failures by cataloging incidents: The ai incident database. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 15458–15463, 2021.

Nigenda, D., Karnin, Z., Zafar, M. B., Ramesha, R., Tan, A., Donini, M., and Kenthapadi, K. Amazon sagemaker model monitor: A system for real-time insights into deployed machine learning models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3671–3681, 2022.

Northcutt, C., Jiang, L., and Chuang, I. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021a.

Northcutt, C. G., Athalye, A., and Mueller, J. Pervasive label errors in test sets destabilize machine learning benchmarks. *NeurIPS*, 2021b.

Oala, L., Maskey, M., Bat-Leah, L., Parrish, A., Gürel, N. M., Kuo, T.-S., Liu, Y., Dror, R., Brajovic, D., Yao, X., et al. Dmlr: Data-centric machine learning research–past, present and future. *arXiv preprint arXiv:2311.13028*, 2023.

OpenAI. Gpt-4v(ision) system card, 2023. URL https://cdn.openai.com/papers/GPTV_System_Card.pdf.

Pimentel, M. A., Clifton, D. A., Clifton, L., and Tarassenko, L. A review of novelty detection. *Signal processing*, 99: 215–249, 2014.

Polyzotis, N., Zinkevich, M., Roy, S., Breck, E., and Whang, S. Data validation for machine learning. *Proceedings of machine learning and systems*, 1:334–347, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Rekatsinas, T., Chu, X., Ilyas, I. F., and Ré, C. Holoclean: Holistic data repairs with probabilistic inference. *VLDB*, 2017.

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., and Aroyo, L. M. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2021.

Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F., and Grafberger, A. Automating large-scale data quality verification. *Proceedings of the VLDB Endowment*, 11 (12):1781–1794, 2018.

Siddiqi, S., Kern, R., and Boehm, M. Saga: A scalable framework for optimizing data cleaning pipelines for machine learning applications. *Proc. ACM Manag. Data*, 1(3), November 2023. doi: 10.1145/3617338. URL https://doi.org/10.1145/3617338.

Singh, M., Cambronero, J., Gulwani, S., Le, V., Negreanu, C., Radhakrishna, A., and Verbruggen, G. Datavinci: Learning syntactic and semantic string repairs. *Proc. ACM Manag. Data*, 3(1), February 2025. doi: 10.1145/3709677. URL https://doi.org/10.1145/3709677.

Stonebraker, M., Ilyas, I. F., et al. Data integration: The current status and the way forward. *IEEE Data Eng. Bull.*, 41(2):3–9, 2018.

Tang, Z., Fang, H., Zhou, S., Yang, T., Zhong, Z., Hu, T., Kirchhoff, K., and Karypis, G. Autogluon-multimodal (automm): Supercharging multimodal automl with foundation models, 2024. URL https://arxiv.org/abs/2404.16233.

Wang, J. T. and Jia, R. Data banzhaf: A robust data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 6388–6421. PMLR, 2023.

Yan, M., Wang, Y., Wang, Y., Miao, X., and Li, J. Gidcl: A graph-enhanced interpretable data cleaning framework with large language models. *Proc. ACM Manag. Data*, 2(6), December 2024. doi: 10.1145/3698811. URL https://doi.org/10.1145/3698811.

Yang, L., Wang, Q., Yu, Z., Kulkarni, A., Sanghai, S., Shu, B., Elsas, J., and Kanagal, B. Mave: A product dataset for multi-source attribute value extraction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, pp. 1256–1265, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391320. doi: 10.1145/3488560.3498377. URL https://doi.org/10.1145/3488560.3498377.

Yi, L., Yao, J., Lyu, W., Ling, H., Douady, R., and Chen, C. Geometry of long-tailed representation learning: Rebalancing features for skewed distributions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GySIAKEwtZ.

Zhang, H., Balagopalan, A., Oufattole, N., Jeong, H., Wu, Y., Zhu, J., and Ghassemi, M. Lemon: Label error detection using multimodal neighbors, 2024. URL https://arxiv.org/abs/2407.18941.

Zhu, Z., Zhang, M., Wei, S., Wu, B., and Wu, B. VDC: Versatile data cleanser based on visual-linguistic inconsistency by multimodal large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ygxTuVz9eU.

# A. Appendix

## A.1. Tabular Data Generation

The prompt used to generate the tabular data for *Baby* is:

```
Given this image of baby product, fill these attributes of a table: {",".join(
category_fields)}. Use only basic colors. The Age Limit should be a number. Title should
be a combination of {",".join(category_fields_in_title))}. Return the result as a JSON
with the attributes.
```

The prompt used to generate the tabular data for *Sports* is:

```
Given this image of products for sport/outdoor activities, fill these attributes of a
table: {",".join(category_fields)}. Use only basic colors. Title should be a combination
of {",".join(category_fields_in_title))}. Return the result as a JSON with the attributes.
```

## A.2. LLM Prompting for Error Detection

To evaluate LLaVA 1.5-7b (Liu et al., 2023) with a single table, we prompt LLM with each row from the table:

```
Given a set of e-Commerce product properties, answer if there are errors in the product
properties.
Properties: {col_name-row_value-pairs}.
Please only answer with 'yes' if there are errors, or 'no' if there are no errors.
```

The prompts used to assess performance of the LLM with images and a single value from the label column:

```
Given am e-Commerce product image and property, answer if the property is erroneous,
especially comparing to the image.
Property: {col_name-row_name}.
Please only answer with 'yes' if there are errors, or 'no' if there are no errors.
```

The prompts used to assess performance of the LLM with images and tabular data:

```
Given an e-Commerce product image and set of product properties, answer if there are
inconsistencies between product properties and the image.
Properties: {col_name-row_name-pairs}.
Please only answer with 'yes' if there are errors, or 'no' if there are no errors.
```

For LLaVA-Next Interleave 7b (Li et al., 2024), we add additional examples with images and expected result to the prompts above. For example, for images with a single attribute as a target, we use:

```
<|im_start|>user \nGiven an e-Commerce product image and property, answer if the product
property contains errors, especially comparing to the image. Here are some examples:

Example 1: Image: <image>. Properties: Category - care. Assessment: yes. Category is wrong
. Category should be diapers.
Example 2: Image: <image>. Properties: Product type - wipes. Assessment: yes. Product type
 is wrong. Product type should be diapers.
Example 3: Image: <image>. Properties: Color - multi-colored. Assessment: no.
...

Please evaluate a product with image <image> and the following property: {col_name-
row_name}. Please only answer with 'yes' if there are errors, or 'no' if there are no
errors.|im_end|><|im_start|>assistant
```

### A.3. Supplementary Tables

| Name | #Rows | #Cols | Image size |
|---|---|---|---|
| *Fashion* | 2,907 | 6 | 1080×1440 |
| *Baby* | 1,299 | 8 | 224×224 |
| *Sports* | 1,368 | 7 | 224×224 |
| *Fashion 44K* | 44,442 | 9 | 1080×1440 |

*Table A1.* Basic statistics of the multi-modal e-Commerce datasets.

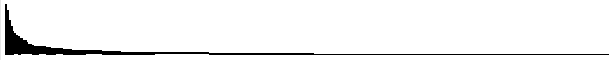
| Dataset | Column | #Distinct | Distinct Frequency Distribution |
|---|---|---|---|
| *Fashion* | Gender | 4 |  |
| | Category | 2 |  |
| | SubCategory | 9 |  |
| | ProductType | 30 |  |
| | Color | 38 |  |
| *Baby* | Category | 20 |  |
| | ProductType | 132 |  |
| | Color | 107 |  |
| | PackageMaterial | 9 |  |
| *Sports* | Category | 55 |  |
| | ProductType | 344 |  |
| | Color | 77 |  |
| | SportType | 65 |  |
| *Fashion 44K* | Gender | 5 |  |
| | Category | 7 |  |
| | SubCategory | 45 |  |
| | ProductType | 144 |  |
| | Color | 47 |  |
| | Season | 5 |  |

*Table A2.* Statistics of all columns and datasets, including value counts histogram of each column, sorted descending by counts.

| Table | Image | Measure | Gender | Category | SubCategory | ProductType | Color |
|---|---|---|---|---|---|---|---|
| ✓ | ✗ | $\mathcal{P}$ | 1.00 | 1.00 | 0.69 | 0.54 | 0.61 |
| | | $\mathcal{R}$ | 0.01 | 0.32 | 0.55 | 0.60 | 0.55 |
| | | $\mathcal{F}1$ | 0.02 | 0.48 | 0.61 | 0.57 | 0.58 |
| ✗ | ✓ | $\mathcal{P}$ | 0.79 | 1.00 | 0.91 | 0.66 | 0.52 |
| | | $\mathcal{R}$ | 0.95 | 0.88 | 0.98 | 0.93 | 0.93 |
| | | $\mathcal{F}1$ | **0.86** | 0.94 | **0.94** | 0.77 | 0.66 |
| ✓ | ✓ | $\mathcal{P}$ | 0.95 | 1.00 | 0.94 | 0.78 | 0.75 |
| | | $\mathcal{R}$ | 0.57 | 1.00 | 0.94 | 0.88 | 0.64 |
| | | $\mathcal{F}1$ | 0.71 | **1.00** | **0.94** | **0.83** | **0.69** |

*Table A3.* AutoGluon + Cleanlab performance per column in *Fashion*.

| Table | Image | Measure | Category | ProductType | Color | PackageMaterial |
|---|---|---|---|---|---|---|
| ✓ | ✗ | $\mathcal{P}$ | 0.70 | 0.54 | 0.26 | 0.61 |
| | | $\mathcal{R}$ | 0.34 | 0.36 | 0.39 | 0.81 |
| | | $\mathcal{F}1$ | 0.46 | 0.43 | 0.32 | 0.70 |
| ✗ | ✓ | $\mathcal{P}$ | 0.70 | 0.44 | 0.55 | 0.87 |
| | | $\mathcal{R}$ | 0.88 | 0.41 | 0.45 | 0.92 |
| | | $\mathcal{F}1$ | 0.78 | 0.42 | **0.50** | 0.89 |
| ✓ | ✓ | $\mathcal{P}$ | 0.80 | 0.61 | 0.57 | 0.86 |
| | | $\mathcal{R}$ | 0.88 | 0.44 | 0.45 | 0.94 |
| | | $\mathcal{F}1$ | **0.83** | **0.51** | **0.50** | **0.90** |

*Table A4.* AutoGluon + Cleanlab performance per column in *Baby*.

| Table | Image | Measure | Category | ProductType | Color | SportType |
|---|---|---|---|---|---|---|
| ✓ | ✗ | $\mathcal{P}$ | 0.59 | 0.51 | 0.39 | 0.49 |
| | | $\mathcal{R}$ | 0.33 | 0.56 | 0.61 | 0.75 |
| | | $\mathcal{F}1$ | 0.43 | 0.53 | 0.47 | 0.59 |
| ✗ | ✓ | $\mathcal{P}$ | 0.77 | 0.61 | 0.60 | 0.50 |
| | | $\mathcal{R}$ | 0.77 | 0.60 | 0.73 | 0.71 |
| | | $\mathcal{F}1$ | **0.77** | **0.61** | **0.66** | 0.59 |
| ✓ | ✓ | $\mathcal{P}$ | 0.74 | 0.62 | 0.60 | 0.55 |
| | | $\mathcal{R}$ | 0.74 | 0.60 | 0.64 | 0.71 |
| | | $\mathcal{F}1$ | 0.74 | **0.61** | 0.62 | **0.62** |

*Table A5.* AutoGluon + Cleanlab performance per column in *Sports*.

| Table | Image | Measure | Gender | Category | SubCategory | ProductType | Color | Season |
|---|---|---|---|---|---|---|---|---|
| ✓ | ✗ | $\mathcal{P}$ | 0.90 | 1.00 | 0.86 | 0.74 | 0.47 | 0.70 |
| | | $\mathcal{R}$ | 0.08 | 0.04 | 0.12 | 0.28 | 0.21 | 0.94 |
| | | $\mathcal{F}1$ | 0.15 | 0.09 | 0.20 | 0.41 | 0.29 | 0.80 |
| ✗ | ✓ | $\mathcal{P}$ | 0.74 | 0.96 | 0.78 | 0.55 | 0.35 | 0.44 |
| | | $\mathcal{R}$ | 0.92 | 0.99 | 0.97 | 0.93 | 0.95 | 0.77 |
| | | $\mathcal{F}1$ | **0.82** | **0.97** | **0.87** | 0.70 | **0.51** | 0.57 |
| ✓ | ✓ | $\mathcal{P}$ | 0.74 | 0.96 | 0.96 | 0.89 | 0.50 | 0.75 |
| | | $\mathcal{R}$ | 0.71 | 0.96 | 0.72 | 0.71 | 0.29 | 0.94 |
| | | $\mathcal{F}1$ | 0.73 | 0.96 | 0.82 | **0.79** | 0.37 | **0.84** |

*Table A6.* AutoGluon + Cleanlab performance per column in *Fashion 44K*.

| Measure | Gender | Category | SubCategory | ProductType | Color |
|---|---|---|---|---|---|
| Table | 0.01 | 0.44 | 0.07 | 0.45 | 0.06 |
| Image | 0.25 | **0.97** | 0.66 | 0.65 | 0.10 |
| Table & Image | **0.83** | 0.88 | **0.94** | **0.80** | **0.72** |

*Table A7.* Error detection and repair accuracy for *Fashion* columns (table + image) using Cleanlab.

| Measure | Category | ProductType | Color | PackageMaterial |
|---|---|---|---|---|
| Table | 0.18 | 0.17 | **0.21** | 0.79 |
| Image | **0.79** | 0.20 | 0.16 | 0.18 |
| Table & Image | 0.54 | **0.24** | **0.21** | **0.81** |

*Table A8.* Error detection and correction accuracy for *Baby* (table + image) using Cleanlab.

| Measure | Category | ProductType | Color | SportType |
|---|---|---|---|---|
| Table | 0.15 | 0.05 | 0.33 | 0.42 |
| Image | **0.64** | 0.30 | **0.47** | 0.40 |
| Table & Image | 0.46 | **0.35** | 0.31 | **0.51** |

*Table A9.* Error detection and correction accuracy for *Sports* (table + image) using Cleanlab.

| Measure | Gender | Category | SubCategory | ProductType | Color | Season |
|---|---|---|---|---|---|---|
| Table | 0.10 | 0.04 | 0.07 | 0.20 | 0.02 | 0.67 |
| Image | **0.88** | **0.98** | **0.94** | **0.84** | **0.69** | 0.64 |
| Table & Image | 0.13 | 0.73 | 0.54 | 0.58 | 0.04 | **0.86** |

*Table A10.* Error detection and repair accuracy for *Fashion 44K* columns (table + image) using Cleanlab.

| Method | **Table** used? | **Image** used? | *Fashion* Time (S) | *Fashion 44K* Time (S) |
|---|---|---|---|---|
| Raha | ✓ | ✗ | 12 | 85 |
| AutoGluon + DataScope | ✓ | ✗ | 655 + 7 | 1,782 + 4,157 |
| AutoGluon + Cleanlab | ✓ | ✗ | 722 + 17 | 1,913 + 149 |
| LLaVA (zero-shot) | ✓ | ✗ | 69 | 3,276 |
| LLaVA (few-shot) | ✓ | ✗ | 130 | 3,724 |
| AutoGluon + DataScope | ✗ | ✓ | 3,036 + 321 | 5,009 + 6,584 |
| AutoGluon + Cleanlab | ✗ | ✓ | 2,749 + 353 | 5,664 + 6,924 |
| LLaVA (zero-shot) | ✗ | ✓ | 1,251 | 19,725 |
| LLaVA-I. (few-shot) | ✗ | ✓ | 10,647 | 69,071 |
| AutoGluon + DataScope | ✓ | ✓ | 3,346 + 327 | 17,289 + 5,866 |
| AutoGluon + Cleanlab | ✓ | ✓ | 9,354 + 59 | 19,289 + 454 |
| LLaVA (zero-shot) | ✓ | ✓ | 193 | 4,366 |
| LLaVA-I. (few-shot) | ✓ | ✓ | 497 | 8,946 |
| LEMoN | ✓ | ✓ | 489 | 5,057 |

*Table A11.* Time elapsed for training each method *Fashion* and *Fashion 44K* datasets that have similar structure and contain 2,907 and 44,442 items respectively. All experiments are conducted on the same machine, using two Xeon Gold 6326 CPUs at 2.9GHz, 1 TB DDR4 memory, and an A100 80GB GPU.