Table 1. Comparison of crowd-sourced datasets for table structure recognition.

Dataset	Input Modality	# Tables	Cell Topology	Cell Content	Cell Location	Row & Column Location	Canonical Structure
TableBank [9]	Image	145K	✓				
SciTSR [3]	PDF*	15K	✓	✓			
PubTabNet [22,23]	Image	510K‡	✓	✓	✓†		
FinTabNet [22]	PDF*	113K	✓	✓	✓†		
PubTables-1M (ours)	**PDF***	**948K**	✓	✓	✓	✓	✓

*Multiple input modalities, such as image or text, can be derived from annotated PDF data.
‡The authors release annotations for 510K of the 568K total tables in their dataset.
†For these datasets, cell bounding boxes are given for non-blank cells only and exclude any non-text portion of a cell.

components or custom training procedures, and incorporate rules or other unlearned processing stages tailored to the TSR task, which brings in prior knowledge to lessen the burden placed on learning the task from data. Currently, no solution exists that uses a simple supervised learning approach with an off-the-shelf architecture, solves the TSR task completely, and achieves state-of-the-art performance.

3. PubTables-1M

In this section, we describe the process used to develop PubTables-1M. First, to obtain a large source of annotated tables, we choose the PMCOA corpus, which consists of millions of publicly available scientific articles. In the PMCOA corpus, each scientific article is given in two forms: as a PDF document, which visually presents the article, and as an XML document, which provides a semantic description and hierarchical organization of the document's elements. Each table's content and structure is specified using standard HTML tags.

However, because this data was not intended for use as ground truth for table extraction modeling, it does not explicitly label or guarantee multiple things that would be helpful for this purpose. For instance, although the same tables appear in both documents, no direct correspondence between them is given, nor the spatial location of each table. In terms of data quality, while tables are generally annotated reliably, it is not guaranteed that column headers are annotated completely or that text content as annotated exactly matches the text content as it appears in the PDF. Finally, some labels, such as the row header for each table, are not annotated at all.

The basic approach we take to overcome these issues is first we attempt to reliably infer as much missing annotation information as possible (for instance, the spatial location of each table) from the information that is present, then we verify that each annotation meets certain requirements for consistency. In some cases, we correct an annotation to attempt to make it more consistent, such as merging cells that are oversegmented. We consider certain requirements for tables to be strict and samples whose annotations violate these are removed. This provides a set of conditions for quality and consistency that the annotations are guaranteed to meet. In the rest of this section, we describe these conditions and the steps we take to derive ground truth that meets them.

Alignment Text in a PDF document has spatial location $[x_{\text{min}}, y_{\text{min}}, x_{\text{max}}, y_{\text{max}}]$, while text in an XML document appears inside semantically labeled tags. Because the correspondence between these is not given, the first step in creating PubTables-1M is to match the text content from both. We process the PDF document into a sequence of characters each with their associated bounding box and use the Needleman-Wunsch algorithm [10] to align this with the character sequence for the text extracted from each table HTML. This connects the text within each HTML tag to its spatial location with the PDF document. For each cell with text, we compute the union of the bounding boxes for each character of the cell's text, which we refer to as a *text cell* bounding box.

Completion Following alignment, we complete the spatial annotations to define bounding boxes for rows, columns, and the entire table. The bounding box for the table is defined simply as the union of all text cell bounding boxes. The x_{min} and x_{max} of the bounding box for each row are defined as the x_{min} and x_{max} of the table, giving every row the same horizontal length. The y_{min} and y_{max} of the bounding box for each row, m, are defined as the y_{min} and y_{max} of the union of the text cells for each cell whose starting row or ending row is m. Similarly, the y_{min} and y_{max} of the bounding box for each column are defined as the y_{min} and y_{max} of the table. The x_{min} and x_{max} of the bounding box for each column, n, are defined as the x_{min} and x_{max} of the union of the text cell for each cell whose starting column or ending column is n. From these definitions, the *grid cell* for each cell is defined as the union of the bounding boxes of the cell's rows intersected with the union of the bounding boxes for its columns. Unlike the text cell, the grid cell is defined even for blank cells.