

TABLE I: CROHME 2019 DATA SETS

Tasks	Training	Validation	Test
Formulae (1, 2)	Train 2014 + Test 2013 + Test 2012 9993 expr	Test 2014 986 expr	Test 2019 1199 expr
Symbols (1a, 2a)	Train 2014 + Test 2013 + Test 2012 180440 symbols + junks	Test 2014 18435 symbols + junks	Test 2016 15483 symbols + junks
Structure (1b, 2b)	Train 2014 + Test 2013 + Test 2012 9993 expr	Test 2014 986 expr	Test 2016 1147 expr
Formula Detection (3)	36 rendered PDFs (600dpi): 569 pages 26395 formula regions Character BBs and labels	n/a	10 rendered PDFs (600dpi): 236 pages 11885 formula regions Character BBs <i>without</i> labels

to render images. Participants then convert these handwritten formula images to a Symbol Layout Tree. For evaluation, the same evaluation tools are used as for Task 1. Again, participants are ranked by the expression rate of their system.

- **Task 2a (symbols):** subtask where participants recognize isolated symbols, including ‘junk’ (invalid symbols). Ranked by symbol recognition rate.
- **Task 2b (parsing from provided symbols):** subtask where participants parse formulas from provided symbols (bounding boxes + labels). Ranked by expression rate.

**Task 3. Detection of Formulas in Document Pages.** Given a document page along with the bounding boxes of characters on that page (as are available for born-digital PDF files), participating systems identify formulas using bounding boxes. Evaluation is performed using Intersection-over-Union (IoU, or equivalently the Jaccard similarity coefficient), and systems are ranked based on their F-measure after matching output formula boxes to ground truth formula regions.

### III. DATASETS AND FORMULA ENCODINGS

In this Section we describe data used in the competition, how it was collected, and the encodings used. Table I summarizes the datasets used for the competition.

**Handwritten Formulas: Input Data.** For Task 1, we use online data in the same InkML and Label Graph (LG) file formats from previous CROHMEs. Strokes are defined by lists of (x,y) coordinates, representing sampled points as a stroke is written. Groupings of strokes into symbols, symbol labels, and formula structure are provided in both the InkML and LG formats. In InkML structure is represented using Presentation MathML (an XML-based representation), while in LG a simpler CSV-based representation is used. In both cases, formula structure is represented by a Symbol Layout Tree, as seen in Figure 1(b). Roughly speaking, this format represents the appearance of a formula by the placement of symbols on the different writing lines of the expression. Spatial relationships between symbols (e.g., ‘R’ for adjacent-at-right) are indicated using edge labels.

For Task 2, the offline formula data is provided as greyscale images. These were rendered automatically from the online data with  $1000 \times 1000$  pixels with 5 pixels of padding. This format is used for the main task (Task 2) and Task 2b. For

the isolated symbol sub-task, isolated symbols are rendered at  $28 \times 28$  pixels with the same amount of padding (5 pixels). The resolution of the inputs files is fixed for Task 2, but participants were welcome to resize the original images using pre-processing methods of their choice.

**Symbol Layout Graph (symLG) Formula Representation.** The stroke-based LG files used in previous CROHMEs allow all segmentation, classification, and structural errors to be identified unambiguously, even when segmentations disagree [1]. However, with the success of encoder-decoder-based systems that generate L<sup>A</sup>T<sub>E</sub>X output, a new representation is needed - these systems do not output information about stroke segmentation or the location of symbols in the input, instead producing Symbol Layout Trees directly.

Figure 1 shows two graph representations for the same expression ‘ $2+3^c$ ’. In the Stroke Label Graph (LG), there are 5 nodes (one per stroke), and edges represent segmentation and spatial relationships between pairs of strokes (including ‘no relationship’). For the Stroke Label Graph, node identifiers are for individual strokes. In the Symbol Label Graph (symLG), there are 4 nodes (one per symbol) and edges represent relationships between symbols (no segmentation information is provided). For symLG, node identifiers are constructed from the sequence of relation labels on the path from root to the symbol. For example, ‘c’ in Figure 1 has the identifier ‘oRRSup’ (origin/root, Right, Right, Superscript).

To compute the similarity of two Symbol Layout Trees in our symLG representation, we use an adjacency matrix. Labels on the diagonal define symbol labels, while off-diagonal elements represent spatial relationships between parent and child symbols. Using this representation, we can determine how formulas in an SLT representation differ in structure and symbol labels, but not the correspondence between symbols and relationships in a symLG file and the input data (i.e., strokes/images). Still, the symLG representation allows existing metrics and tools designed for evaluation of stroke-level LG files at the symbol level to be used directly. We note that symLG is closely related to the tree-based symbolic representation from earlier CROHME competitions [1], but permits more detailed error analysis.

**Formula Data Collection.** We used the labeled handwritten formulae from previous CROHMEs that are publicly available