# Dual Input Stream Transformer for Vertical Drift Correction in Eye-Tracking Reading Data

Thomas M. Mercier, Marcin Budka, Martin R. Vasilev, Julie A. Kirkby, Bernhard Angele, and Timothy J. Slattery

***Abstract*—We introduce a novel Dual Input Stream Transformer (DIST) for the challenging problem of assigning fixation points from eye-tracking data collected during passage reading to the line of text that the reader was actually focused on. This post-processing step is crucial for analysis of the reading data due to the presence of noise in the form of vertical drift. We evaluate DIST against eleven classical approaches on a comprehensive suite of nine diverse datasets. We demonstrate that combining multiple instances of the DIST model in an ensemble achieves high accuracy across all datasets. Further combining the DIST ensemble with the best classical approach yields an average accuracy of 98.17%. Our approach presents a significant step towards addressing the bottleneck of manual line assignment in reading research. Through extensive analysis and ablation studies, we identify key factors that contribute to DIST's success, including the incorporation of line overlap features and the use of a second input stream. Via rigorous evaluation, we demonstrate that DIST is robust to various experimental setups, making it a safe first choice for practitioners in the field.**

***Index Terms*—Artificial intelligence, computer vision, machine learning, pattern recognition, psychology.**

## I. Introduction

THE ability to read is an indispensable skill in modern society, making reading a prominent subject in the psychological and cognitive sciences. Eye-tracking technology has emerged as a valuable tool for uncovering the cognitive processes involved in reading, offering unique insights into the reading patterns of individuals. It involves measuring the gaze position on a computer screen over time with a typical sampling frequency of 1000 Hz (although higher and lower sampling rates such as 500 Hz and 2000 Hz are not unusual) by analyzing the relative position of the pupil and corneal reflection centers [1], [2]. This can provide critical information about where and for how long an individual's eyes are fixating while they navigate text, revealing essential clues about the mental processes at play [3]. From the gaze position measurements, each gaze point can be considered to either be part of a fixation, a state of oculomotor control where the gaze position is held fairly constant while information is extracted from the image on the retina, or part of a saccade, which is a very fast, ballistic eye movement that moves the center of the gaze to a different part of the visual field. Depending on the properties of the visual stimulus and the task, the mean fixation duration can vary between 180 and 330 ms (with typical reading fixations having lengths between 200 and 250 ms). In some tasks, fixations as short as 50 ms and as long as 600 ms have been observed. The duration of a saccade is directly related to the distance it covers in the visual field. In reading, saccade duration usually does not exceed 30–50 ms [3].

Eye movement data have proven invaluable in the study of reading behavior and in the diagnosis and treatment of specific disorders such as dyslexia and schizophrenia [2], [4], [5]. By examining eye movements during reading tasks, researchers can better understand the cognitive mechanisms engaged in the comprehension of written material and potentially improve interventions for those who struggle with reading due to neurological differences or other factors. Eye-tracking experiments can also offer insight into accessibility related issues of visual stimuli such as websites [6].

While the technological advances in recent decades have enabled the recording of gaze position during reading with high accuracy, eye-tracking data from such recordings consists of the raw gaze position samples and thus still requires post-processing to identify which gaze positions are part of the fixations and which are part of a saccade. Furthermore, these fixations need to be assigned to an area of interest in the reading stimulus, which can be a particular character or word, depending on the design of the experiment.

Modern eye-trackers can record gaze position with high precision, but they can only ever be as good as their calibration. Even small, involuntary participant movements can affect the precision of the calibration, which can result in dynamically changing offset in both the horizontal and the vertical coordinate. Due to the way printed text is organized in lines, an offset in the horizontal coordinate will lead to a fixation being assigned to a different letter, or at worst an adjacent word on the same line. Because of this, horizontal drift is usually not corrected.

Manuscript received 31 July 2023; revised 9 May 2024; accepted 31 May 2024. Date of publication 10 June 2024; date of current version 5 November 2024. This work was supported in part by a research grant from the Microsoft Corporation to T.J.S. which was marched by Bournemouth University to fund PhD studentships. Recommended for acceptance by V. Demberg. *(Corresponding author: Thomas M. Mercier.)*

Thomas M. Mercier, Marcin Budka, Julie A. Kirkby, and Timothy J. Slattery are with the Faculty of Science & Technology, Bournemouth University, BH12 5BB Poole, U.K. (e-mail: tmercier@bournemouth.ac.uk).

Martin R. Vasilev is with the Faculty of Brain Sciences, University College London, WC1E 6BT London, U.K.

Bernhard Angele is with the Nebrija Research Centre in Cognition (Centro de Investigación Nebrija en Cognición, CINC), Universidad Antonio de Nebrija, 28043 Madrid, Spain, and also with Bournemouth University, BH12 5BB Poole, U.K.

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3411938, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3411938

However, a vertical offset can lead to a fixation appearing to be on an incorrect line, possibly many words away from the correct fixation location [7]. This offset can be so pronounced as to make the recorded fixation appear to lie on one or several lines above or below the line that the reader was actually focusing on. An example for this phenomenon is shown in Fig. S1. The adjustment of the vertical coordinate of a fixation point to correct for this vertical drift is referred to as line assignment since the vertical coordinate is set to the center of the line that the fixation is assigned to.

Single line experiments avoid the issues associated with line assignment as assigning fixations is trivial when there is only one line, however, real-world readers do not typically read single lines on an otherwise blank page or screen. In order to get a more naturalistic sample of real-world reading, many experiments involve full passages of text that require each fixation to be assigned to a line of text to carry out further analysis. Since simply assigning a fixation point to whichever line it is closest to can lead to incorrect assignments, more careful approaches to line assignment have to be utilized. This can present a significant hurdle to carrying out large number of trials for studies involving multi-line passages of text as the process is often carried out manually. See for example [8], [9], [10].

Line assignment of fixations is made significantly more difficult due to noise in the tracking data, as can arise from loss of calibration of the eye-tracker, subtle head or body movements or even pupil dilation during the experiment [7]. Such noise can take the form of dynamically changing vertical drift of the recorded fixations, which causes fixations to be recorded as falling on lines above or below of the actually fixated line. There have been several attempts to create algorithms to automate the line assignment process and therefore enable researchers to carry out larger studies of multi-line reading. See Carr et al. [7] for a review of several available algorithms.

Such techniques, however, have not been evaluated on multiple datasets and are commonly not easily accessible to researchers in the field of psychology. This may lead researchers to have concerns about their accuracy and reliability as evidenced by the fact that line assignment to correct vertical drift is still commonly done manually [8], [9], [10] despite the availability of line assignment algorithms. Hence, manual correction is considered the gold standard for addressing noise in eye-tracking fixation data [7]. Unfortunately, any form of manual assignment depends on the individual carrying out the task and can yield inconsistent results based on the individual carrying out the task [11]. Nevertheless, if the available algorithms prove insufficient, they are left with no choice but to resort to a manual approach. Since this is both time-consuming and increases subjectivity in the resulting assignments, it is desirable for this process to be reliably automated, an idea initially put forward by Cohen [12].

In this paper we present a novel way of tackling the line assignment problem by utilizing a deep learning (DL) architecture trained using a rank-consistent ordinal regression loss function. Our proposed architecture uses two input streams to incorporate information about both the eye-tracking fixations and the stimulus text used in the trial. We call our model Dual Input Stream Transformer (DIST).[1] We show that combining an ensemble of our proposed model with classical algorithms in a "Wisdom of the Crowds" (WOC) approach outperforms the best classical algorithms (including a WOC of all classicals) on all datasets. The idea of combining the algorithms this way is inspired by [13]. We acknowledge that our model is unlikely to perform well for data with very different fixation patterns than those of the datasets used to train the model, such as experiments using different fonts or font sizes, for example. The datasets considered in this study all used either the Consolas or the Courier New fonts with font sizes ranging from 11 to 22. Nevertheless, the diversity of these datasets and the data normalization schemes employed should allow the model to generalize to many unseen datasets without the user having to carry out any manual line assignment or fine-tuning of the model. Furthermore, the combination with the classical models via the WOC approach should further increase resilience to data that is very unlike the training data. Note that all datasets are based on text being read left-to-right, so the model is unlikely to perform well on languages read in a different way. We further acknowledge that due to the increased complexity of our method how the results came to be can not be easily traced. However, in practice, this does not present a significant drawback since, regardless of the algorithm used, some visual inspection of the results will be necessary. Cases where the correction algorithm made particularly large adjustments or where there is strong disagreement between different correction methods should be inspected in particular. In practice, this will only apply to a small subset of trials.

To the best of our knowledge this paper presents the first comprehensive comparison of multiple line assignment algorithms across diverse datasets in the domain of eye-tracking research. For practitioners in the field of psychology the best model will likely be the one that can be robustly applied to data from various studies. Ultimately, widespread adoption of algorithmic tools could lead to increased efficiency and consistency in the analysis of such data, fostering more reliable and accurate scientific insights.

Our contributions are as follows: 1) introduction of a novel transformer-based architecture for the problem of line assignment of fixation coordinates, 2) comparative evaluation of the newly introduced approaches on diverse data from multiple studies, 3) increased robustness and accuracy of line assignment to enable reading researchers to carry out larger studies involving multi-line reading experiments.

## II. Related Work

Carr et al. [7] recently reviewed and evaluated several classical algorithms for correcting vertical drift in eye-tracking data. They summarized the approaches into ten categories and implemented each approach to enable direct comparison. They reported high accuracies for most algorithms for their small dataset of manually corrected experimental data as well as synthetic data

[1]Our code and the user interface can be found here: https://github.com/Gittingthehubbing/DIST-Dual_Input_Stream_Transformer

from simulated experiments. By implementing all algorithms with a similar interface, the authors greatly improved usability and comparability of the published algorithms. We adapt their implementations to compare our model to classical work. Note that while it is not discussed in their publication, Carr et al. nevertheless offers an implementation of the *slice* (described below) algorithm. This brings the total number of compared classical algorithms to eleven. As will be shown in Table II of Section VII, the compared algorithms showed large variability for the different dataset. When the accuracies across all trials and datasets are averaged, the *cluster*[7], [14], *merge* and *regress* [7], [15], [16] algorithms show the best performance among the individual classical methods.

Carr's *cluster* [7], [14] implementation applies k-means clustering in order to assign all fixations to one of m clusters, where m is equal to the number of lines in the passage. The mean value of the y-coordinate of all fixations in one cluster is used to find the line number that a particular cluster corresponds to.

The *merge* algorithm as implemented by Carr [7] uses Spakov et al.'s [17] post hoc correction approach with modifications. It generates progressive sequences of consecutive fixations and merges them into larger sequences until there's one sequence per line of text. A y threshold defines proximity, and a regression-based merge process ensures similar gradients and low errors between merged sequences. The algorithm follows Spakov et al.'s [17] four phases for relaxing criteria, while the number of sequences is reduced to match text lines in positional order.

The implementation by Carr et al. [7] of the *regress* algorithm is based on the R package *FixAlign* [12]. It fits a number of regression lines, meaning lines going through a set of points that are found by minimizing the y-distance of all points to the line, to the unordered fixation points and subsequently assigns each fixation to its highest likelihood regression line. This identifies outliers and assigns each fixation to their most appropriate line of text.

The recently published *slice* algorithm works in three main steps [18]. Firstly, it finds fixation sequences that are likely to belong to the same line. Secondly, it goes through the sequences, starting from the longest and finds which of the sequences belong to the same or an adjacent line. Lastly, it ensures that the number of detected sequences is equal to the number of lines in the text. They report 95.3% accuracy for a subset of the MECO dataset.

To the best of our knowledge this paper is the first DL-based solution to the line assignment problem. Eye-tracking data have been utilized to train machine learning (ML) models to diagnose reading difficulties. While a review of the many ML related eye-tracking applications is beyond the scope of this paper, we would like to briefly discuss some examples. In an early study by Rello and Ballesteros [19] the authors used the eye-tracking data of 97 participants, 48 of which had a dyslexia diagnosis, to extract features that were deemed relevant to the problem of reading difficulty diagnosis. These features were fed into a polynomial Support Vector Machine (SVM) to classify each participant as either dyslexic or not. Using this approach the authors report a 10-fold cross-validation accuracy of 80.18%.

In a more recent study, Vajs et al. [20] used a DL approach based on a convolutional neural network (CNN) to perform dyslexia classification for a dataset of 30 subjects, half of which had a diagnosis of dyslexia. They use the gaze coordinates without extracting fixations or other features, however, they do clean the data by removing eye blinks and data points they consider invalid due to the gaze not being registered by the measurement device. After splitting each trial data into a number of sequences, the authors visualize the gaze trace with the distance between subsequent points being encoded as color. This visualization serves as the input to the CNN with each visualization being assigned the label of belonging to a dyslexic subject or not. They report an accuracy of 87%. These reports highlight the importance of eye-tracking data for diagnosis of widespread pathologies.

Additionally, modeling of eye-tracking data has also been used to assess second language proficiency by scoring how similar the gaze patterns, as represented by eye-movement features, of the subjects to those of native speakers and predicting scores of other language tests via a regression model [21]. Furthermore, DL-based models using only features extracted from eye-movements or models that combine them with linguistic features of the text have been applied to the task of estimating reading comprehension [22], [23]. ML models have also been used to directly predict the fixations using an approach that incorporates both the sequence of fixations and the sequence of words making up the stimulus text [24].

## III. Datasets

We make use of data from nine studies (see Table I)[2]. For all these studies the authors carried out manual line assignment of all fixations. In total these datasets contain 15,446 trials, although the two largest datasets, ChA and TexF, only have two lines of text in their stimulus material. The low number of lines makes the correction task for these datasets much easier than for typical paragraph data and therefore pushes up the average accuracies for all compared approaches. Here, one trial is considered to be the result of a participant reading one screen of text and yielding one fixation sequence. For each of the fixations in such a sequence the ground truth consists of the index of the line to which the fixation is assigned, with this assignment being based on the gold standard human-corrected manual approach. Note that since none of the classical algorithms can discard fixations we only consider those fixations that had been assigned to a line by the human labeler, hence fixations that the human labeler discarded are not considered. Across datasets this was the case for an average of 2.8% (ranging from 0.15 to 8.2%) of fixations. Since the stimulus texts varied in length and line counts, this results in a differing number of target classes for each dataset. Please see Section IV for how this is handled in the model design.

Table I shows the dataset characteristics which varied in a variety of aspects. This provided the model a broad basis for learning how to assign fixations to lines under varying experimental conditions. On average, the text stimuli varied between 1

[2]The preprocessed datasets and links to the raw data, where available, can be found here: https://osf.io/zt9gn.

TABLE I
CHARACTERISTICS AND ASSOCIATED REFERENCES FOR ALL DATASETS

| Dataset | Abbr [25] | ChA [8] | OffN [26] | TexF [10] | CD [27] | OZ [28] | Harry [29] | MECOde [30] | Carr [7] |
|---|---|---|---|---|---|---|---|---|---|
| **Mean no. fixations** | 143.11 | 26.01 | 114.16 | 19.04 | 99.45 | 90.19 | 90.39 | 247.09 | 208.13 |
| **Mean trial time** | 39.49 | 10.40 | 31.90 | 4.74 | 26.84 | 22.42 | 27.79 | 59.42 | 49.43 |
| **Min lines** | 9 | 2 | 1 | 2 | 6 | 8 | 10 | 10 | 10 |
| **Max lines** | 12 | 2 | 12 | 2 | 8 | 10 | 10 | 14 | 13 |
| **Min words** | 95 | 15 | 10 | 13 | 84 | 82 | 133 | 144 | 104 |
| **Max words** | 132 | 25 | 155 | 34 | 116 | 142 | 153 | 225 | 168 |
| **No. trials** | 1599 | 2682 | 1506 | 6396 | 1150 | 1116 | 300 | 648 | 48 |
| **Mean fixation duration (ms)** | 237.15 | 376.24 | 208.52 | 206.72 | 220.63 | 209.97 | 234.26 | 203.83 | 223.07 |
| **Mean no. words per line** | 10.97 | 8.78 | 11.39 | 10.10 | 13.02 | 10.88 | 12.54 | 14.36 | 11.45 |
| **Mean line width (pixels)** | 826.50 | 671.85 | 964.51 | 759.32 | 821.42 | 704.83 | 1035.12 | 1557.36 | 1058.32 |
| **Min line width (pixels)** | 65 | 288 | 56 | 372 | 44 | 48 | 238 | 270 | 104 |
| **Max line width (pixels)** | 975 | 800 | 1316 | 1376 | 913 | 1080 | 1106 | 1770 | 1176 |
| **Is raw data public?** | Yes | No | No | Yes | Yes | Yes | No | No | Yes |
| **Language** | English | English | English | English | English | English | English | German | Italian |
| **Age group** | 18-45 | 7-10 | 18-64 | 18-26 | 18-40 | 19-63 | 18-31 | 18-39 | Children + Adults |

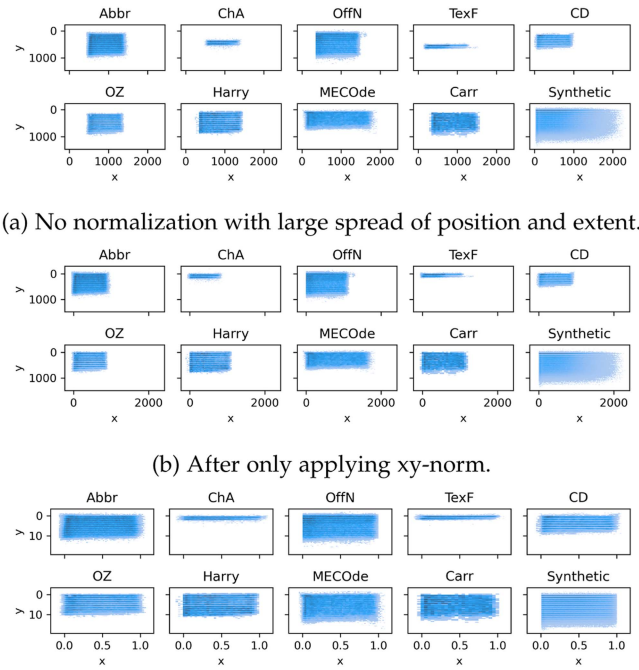
Mean no. of fixations is how many fixations appear in a trial on average. Mean trial time is how much time in seconds passes between start and end of a trial, on average. Min and max no. of lines gives what range of number of lines in a trial exists for a dataset, thereby giving an indication of paragraph length for each dataset. Max/min no. of words gives the largest/smallest number of words appearing in the stimuli of the dataset. No. of trials is the total number of trials in the dataset. Mean fixation duration gives the duration of a fixation in ms. Mean no. of words per line gives the number of words per line for the trials in a dataset, on average. Mean line width gives the width of an average line in pixels. Min/max line width gives the smallest/largest line width in pixels for the trials in a dataset. Age group gives the age range for the study.

and 14 lines yielding on average 19 to 247 fixations per trial. The average fixation duration is fairly consistent except for the ChA dataset having nearly twice the average duration. All datasets except MECOde and Carr were collected at Bournemouth University in the U.K. The MECOde dataset was kindly shared by the creators of the MECO dataset [18], [30]. The Carr dataset is publicly available and associated with Carr et al. [7].

In addition to the datasets available from eye-tracking studies, we produce a synthetic dataset of fixation sequences with different kinds of noise added to them. To produce the synthetic sequences we adapt code from [7]. Based on the *wikitext* dataset [31] we generate passages consisting of 8 to 14 lines of English text with each line being up to 130 characters long and the line height varying between 49 and 79 pixels. The choice for each parameter for each passage was based on a uniform random distribution. For each passage a sequence of up to 500 fixations is generated. To simulate the effects of loss of equipment calibration and other disturbances that can result in both random and systematic errors in the recorded fixation positions the y-coordinate is determined by $y = \mathcal{N}(l_y, d_{\text{noise}}) + l_y d_{\text{shift}}$, where $\mathcal{N}$ is the normal distribution, $l_y$ is the y-center of the line, $d_{\text{noise}}$ is the standard deviation of noise and $d_{\text{shift}}$ is the y-shift in pixels. To increase how realistic the synthetic fixation patterns are, within-line and between-line regressions are added. Regressions refer to the phenomenon of readers fixating parts of the text that lie before the current position. Please see Section S3 and [7] for details on how this is implemented.

While there are DL-based methods to produce fixation sequences [24], we focus on Carr's approach due to its ability to produce a large set of sequences with the ability to control both the type and severity of noise as well as the corresponding line correction for each fixation.

Fig. 1(a) shows the raw fixation point distributions for all datasets highlighting the data diversity across studies. Notably, the fixation distributions show a pattern of darker blue forming lines where the fixations cluster on the lines of text of the different stimuli but with significant variation around them.




(a) No normalization with large spread of position and extent.

(b) After only applying xy-norm.

(c) After applying both xy-norm and lw-norm.

Fig. 1. Fixation density plot illustrating the distribution of the fixation point coordinates for all datasets before and after normalizing the fixation points by subtracting the minimum character bounding box coordinates in each trial (xy-norm) and dividing by the line width and line height (lw-norm). Darker blue colors indicate a higher concentration of fixation points across the trials. The x- and y-axes in Subfigures a) and b) give the coordinates in pixels while c) is fully normalized and thus does not have any units. Please see Section S5 for an enlarged version of this figure.

This clustering is visible due to the fact that within a specific dataset the stimulus materials for the different trials always have the lines of text in the same position. The differences in the fixation distributions can be attributed to dissimilarities in the stimulus materials, namely the length, formatting and visual presentation of the passages read by participants, as well as the

equipment employed to present the stimuli in each study and the experimental setup. This diversity presents a challenge since the model will likely only be able to correctly assign fixations to lines if the coordinates have a similar scale and distribution to what the model is trained on. As is shown in Fig. 1 and will be expanded on in Section VIII, we address this by normalizing the fixation data using information from the stimuli of each trial of the various studies.

Before feeding the fixation coordinates into the model, the coordinates are normalized by first subtracting the minimum character bounding box coordinates (outer edge of bounding box) found in the trial. We dub this scheme xy-norm. The effect of this normalization step is illustrated in Fig. 1(b). Additionally, the fixation coordinates are further normalized by dividing the $y$ coordinate by the minimum line height and the $x$ coordinate by the maximum line width found in the trial. We dub this scheme lw-norm. This is done because the datasets had been recorded under different experimental conditions and the start position of each block of text is different for each dataset, as can be seen in Fig. 1(a). The effect of the full normalization is illustrated in Fig. 1(c).

Since the different model inputs varied in their scale all values were further normalized by subtracting the mean and dividing by the standard deviation of all training data.

To evaluate the model's ability to generalize beyond its training data, we shall employ a 9-fold cross-validation scheme, where one of the nine datasets is in turn withheld from the training process, ensuring that the model is evaluated on separate, previously unseen data.

## IV. Framework

Conceptually, our dual input approach consists of a sequence of fixation-related features and an image that contains information about both the fixations and the stimulus material being fed into a deep sequence model that uses this information to classify each fixation according to which line of text it belongs to. The sequence of fixation-related information makes up the first input stream and the image makes up the second input stream. The problem at hand can be described as a mapping $g : x_1, x_2 \mapsto y$ that jointly assigns each fixation in a sequence of length $s$ to a line index $y \in \{1, 2, \ldots, K\}$ with $K$ being the maximum line index in a trial, using fixation related features $x_1 \in \mathbb{R}^{f \times s}$ and associated trial related features $x_2 \in \mathbb{R}^{H \times W \times C}$. Here $f$ is the number of fixation related features, $s$ is the number of fixations, $H, W$ and $C$ are the height, width and number of channels in the input image associated with the information fed into the second input stream.

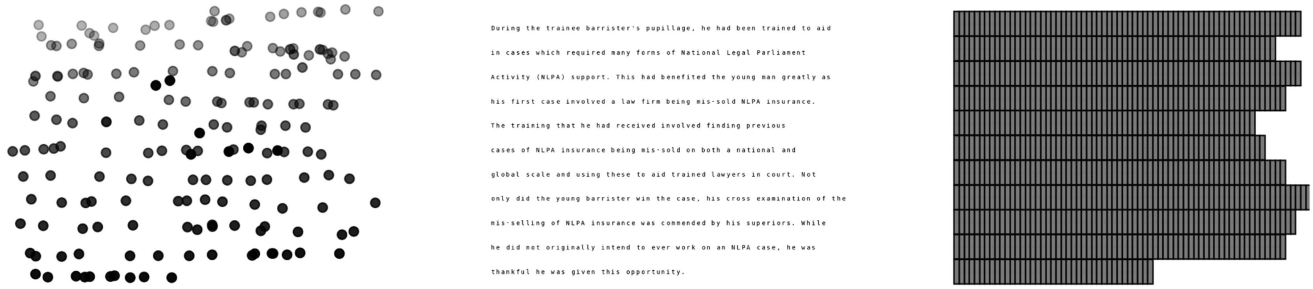
We leverage a bidirectional encoder-only Transformer model, while configured to be a smaller model, it largely follows the original Bidirectional Encoder Representations from Transformers (BERT) architecture [32], as our main encoder model. To distinguish this encoder from the pretrained model introduced in the original publication, we refer to this part of DIST simply as the main encoder. The self-attention-based Transformer architecture [33] has proven widely successful and has largely displaced the previous recurrent network architectures due to their more direct information flow across the sequence and improved training speed and stability [34]. These architectures largely overcome the vanishing or exploding gradient problem that recurrent neural networks suffered from when processing long sequences [35]. BERT-based approaches are able to take into account context from both directions from a specific part of the sequence [32]. This makes such encoders well suited to handle the long fixation sequences.

Our model incorporates two input streams: one consisting of normalized fixation features and one containing the rendered page consisting of the characters, their bounding boxes and coarsely depicted fixation point information. The fixation features consist of the *x*-*y* coordinates of each fixation point and the line number with which the fixation point overlaps with -1 being used when a point does not overlap with any line. A fixation point is considered to overlap with a line if its y-coordinate is within the y-coordinate-range of the character bounding boxes making up the line. No gaps exist between the bounding boxes of adjacent lines within a paragraph for all datasets except CD, which has a ten pixel gap between adjacent lines. We use an ImageNet pretrained CoAtNet [36], [37], [38] as a feature extractor for the second input stream. Its weights are unfrozen, so it is allowed to train with the rest of the architecture. We choose CoAtNet due to its ability to combine the desirable inductive biases found in CNNs with the attention mechanism.

The second input stream information is fed into the model by first grayscale rendering separate images for the stimulus text of the trial, the filled character bounding boxes and a scatter plot of the x-y-coordinates of each fixation point in the associated sequence with the fixation start time (scaled to between 0.25 and 1.0 for each trial) encoded as the gray-level of the scatter plot markers. See Fig. 2 for a depiction of the grayscale images used. These three single channel images are concatenated in the channel dimension to get an image with three channels as the pretrained CoAtNet expects. The CoAtNet then encodes this information into a single vector for each sequence. This vector is then projected to half the hidden dimension of the main encoder and repeated along the sequence dimension (repeat block in Fig. 3) to match the shape of the projected fixation features. The tensors resulting from both input streams are concatenated along the feature dimension and fed into the main encoder architecture. A simple linear head network then turns the encoder output into line prediction outputs for each fixation in the sequence. Fig. 3 shows an overview of the data flow through the architecture.

The main encoder model uses several blocks consisting of a multi-headed self-attention layer, a layer norm and a Multi Layer Perceptron (MLP). These blocks are stacked to create a deep encoder-only transformer network. The head of the network is a simple fully connected layer with no activation function. A Conditional Ordinal Regression for Neural Networks (CORN) [39] loss is used to train the network. It takes in the target line assignments and outputs of the last layer of the architecture, also referred to as the logits. In contrast to a simple cross-entropy loss function CORN takes the order of the line indices into account.

To feed the fixation features to the model in mini batches, each sequence of fixations has to be padded to match the longest sequence length. For the calculation of the loss and accuracy




(a) Fixations. (b) Characters. (c) Bounding Boxes.

Fig. 2. Example of the single channel images used as second input stream to the main encoder.

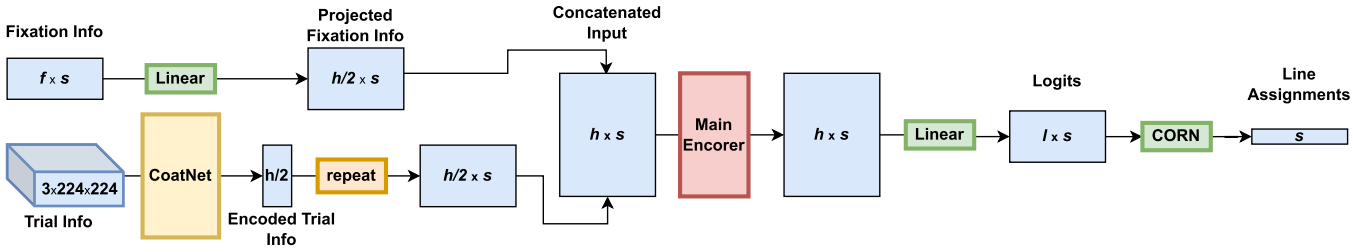



Fig. 3. Model flow with the top half showing the fixation information input stream and the bottom half showing the page information stream. $s$ is the length that all sequences are padded to. $f$ is the number of fixation related features. $h$ is the hidden dimension of the main encoder network. $l$ is the maximum number of lines in the datasets and therefore the highest possible line number DIST can assign to a fixation.

measures this padding is taken into account by masking out the padded parts of each sequence.

Since the stimulus texts and their formatting varied across datasets, the number of possible target lines can differ for each trial. This is addressed by using the largest line count present across all datasets as the output dimension of the final layer. At inference time the prediction is restricted by clipping it to the maximum line index in the trial. It should be noted that this does not present a significant limitation of our approach since that information is always available for data from passage reading studies.

As will be shown in Section VIII, some datasets show very different performance depending on what kind of normalization scheme is used for the fixation data. Since it is desirable to have a single model work well for all datasets, we explore an ensemble-based approach. This is implemented by training the same DIST configuration multiple times on the same training data, randomly reinitializing all non-pretrained weights each time. This is done for each normalization scheme (normalized fixation points are illustrated in Fig. 1(b) and (c)). The weights of the chosen models are then frozen and combined in an ensemble model which takes in the data with the different normalization schemes applied to it and feeds it to the list of models appropriate for each scheme. The resulting logit tensors are then averaged and fed into the CORN inference function (shown in (2)). This requires no further model training. We refer to an ensemble of DIST models as E-DIST. Inspired by Carr's work [13] on combining multiple classical algorithms, we further enhance the E-DIST approach by combining it with the WOC approach, which applies multiple algorithms and uses a majority decision voting system for each fixation assignment and works as follows. For a given trial, first a set line assignments is computed for each algorithm that is to be included in the WOC approach. For each fixation in that sequence, the WOC algorithm counts how many of the algorithms assign the fixation to a line number and the line number with the highest number of votes is chosen as the result. One can give additional weight to the vote of an algorithm by adding multiple copies of its line assignments to the voting pool. We refer to the approach that includes E-DIST corrections in the voting pool as E-WOC and the approach that only uses classically produced corrections as C-WOC. We will show in Section VII that E-WOC achieves the highest accuracy of all approaches when averaged across datasets.

## V. Implementation

Our approach has been implemented in *PyTorch* [40]. We have configured the main encoder to have a depth of four, an internal representation dimension of 512 and eight attention heads, this is equivalent to the "Small" configuration described in [41]. The page encoding computer vision (CV) model is configured to take in images with size of 224 × 224 pixels and has its weights initialized from a model trained to classify images from the ImageNet dataset. For the CoAtNet we chose a model with an embedding dimension of 512. To implement the WOC approach we modify Carr's implementation [42]. All training was carried out on a machine equipped with an Nvidia RTX 3090 with 24 GB

TABLE II
COMPARISON OF MEAN ACCURACY FOR ALL CLASSICAL ALGORITHMS, C-WOC, DIST, E-DIST AND E-WOC

| Method | Abbr | CD | Carr | ChA | Harry | MECOde | OZ | OffN | TexF | Mean |
|---|---|---|---|---|---|---|---|---|---|---|
| **E-WOC** | 97.82 | **98.43** | 98.13 | **99.11** | 98.56 | **96.33** | **98.41** | **97.00** | **99.72** | **98.17** |
| **E-DIST** | **98.16** | 98.25 | **99.63** | **99.14** | **98.87** | 95.87 | 98.14 | 92.41 | 99.61 | 97.79 |
| **C-WOC** | 94.98 | 98.22 | 96.27 | 99.00 | 94.97 | 94.83 | 98.39 | 94.42 | 99.71 | 96.75 |
| **DIST** | 97.71 | 96.77 | 99.16 | 99.09 | 98.44 | 95.80 | 98.02 | 80.46 | 99.51 | 96.11 |
| **cluster** | 94.86 | 93.88 | 94.12 | 98.06 | 98.37 | 90.10 | 96.65 | 94.61 | 99.61 | 95.59 |
| **merge** | 94.62 | 94.85 | 94.20 | 97.22 | 97.83 | 81.16 | 90.32 | 93.97 | 98.84 | 93.67 |
| **regress** | 86.83 | 95.33 | 92.33 | 98.13 | 94.18 | 85.88 | 96.90 | 90.05 | 99.49 | 93.24 |
| **slice** | 94.15 | 93.15 | 95.70 | 98.10 | 97.21 | 94.33 | 95.22 | 71.57 | 99.59 | 93.23 |
| **warp** | 89.71 | 88.83 | 96.68 | 96.25 | 88.52 | 87.69 | 94.28 | 91.90 | 95.50 | 92.15 |
| **chain** | 81.92 | 96.47 | 90.51 | 96.45 | 81.90 | 82.72 | 96.77 | 85.50 | 97.56 | 89.98 |
| **stretch** | 78.74 | 90.51 | 93.97 | 97.59 | 88.30 | 82.78 | 94.67 | 83.11 | 98.35 | 89.78 |
| **attach** | 80.58 | 95.48 | 87.38 | 96.16 | 81.37 | 80.29 | 96.23 | 84.61 | 97.38 | 88.83 |
| **split** | 76.02 | 90.89 | 89.96 | 94.91 | 79.18 | 81.21 | 89.09 | 81.98 | 95.68 | 86.55 |
| **segment** | 78.82 | 79.93 | 80.19 | 95.42 | 79.75 | 67.34 | 71.65 | 79.53 | 94.41 | 80.78 |
| **compare** | 52.61 | 50.35 | 63.58 | 92.71 | 44.01 | 59.88 | 46.01 | 80.18 | 85.45 | 63.86 |

Bold numbers highlight the best performing approach overall and underlined values indicate the best classical approach for each dataset. DIST accuracies are based on using input that had both xy-norm and lw-norm applied. E-DIST accuracy is based on using three instances of the DIST model for each normalization scheme, so six instances in total. As shown in S4 , the effect of adding more instances to E-DIST on accuracy differs across datasets while increasing the computational burden, thus the chosen number is a compromise of these trends. The reported accuracy for E-WOC is based on allocating three votes to E-DIST in the voting pool. This choice is a compromise of the trends shown in Fig. 5.

of memory and an Intel i7-7700 K CPU with 64 GB of memory. The *PyTorch* implementation of our model is publicly available.

## VI. TRAINING SETTING AND EVALUATION METRIC

As touched upon in Section IV for model supervision and evaluation during training we use the CORN loss function shown in (1) [39].

$$L(\mathbf{Z},\mathbf{y}) = -\frac{1}{\sum_{j=1}^{K-1}|S_j|}\sum_{j=1}^{K-1}\sum_{i=1}^{|S_j|}\Big[\log(\sigma(\mathbf{z}^{[i]}))\cdot\mathbb{1}\left\{y^{[i]} > r_j\right\} + (\log(\sigma(\mathbf{z}^{[i]})) - \mathbf{z}^{[i]})\cdot\mathbb{1}\left\{y^{[i]} \le r_j\right\}\Big] \tag{1}$$

where $\mathbf{Z}$ are the outputs of the last layer of the model, referred to as logits in Fig. 3. $\mathbf{y}$ is the set of ground truth line indices $y^{[i]}$ and $i$ the index for the model output $z^{[i]}$. $|S_j|$ denotes the size of the subset of the training data denoted by $j$, that matches the condition of the rank being no higher than $r_j$ with $r_j$ being able to take on values between 1 and $K-1$ for $K$ possible line indices. $\sigma$ is the Sigmoid function. $\mathbb{1}$ is the indicator function giving 1 if the condition is met and 0 if it is not, with the condition being that the ground truth label is above or below rank $r_j$. This loss function takes into account that the line index is ranked and the difference between line numbers is meaningful by creating conditional training sets for each rank, in this case line number. For models trained using the CORN loss (2) computes the predicted line indices $q^{[i]}$:

$$q^{[i]} = 1 + \sum_{j=1}^{K-1}\mathbb{1}\left(\hat{P}\left(y^{[i]} > r_j\right) > 0.5\right) \tag{2}$$

where $\hat{P}$ is the predicted probabilities. CoAtNet is initialized from pretraining on ImageNet, while weights of the rest of the architecture are randomly initialized (Kaiming initialization [43]). All training uses the Adam optimizer and is carried out using the same parameters. An exponential warm-up of 3000 steps with a peak learning rate of $4.5e-4$ which is reduced by half every 3000 training steps until the training ends at 25000 steps.

To compare the performance of our model to that of the classical algorithms we use relative accuracy $\alpha_r$, which is the difference between our model's accuracy $\alpha_m$ and the best classical algorithm's accuracy $\alpha_c$:

$$\alpha_r = \frac{\alpha_m - \alpha_c}{\alpha_c}. \tag{3}$$

Therefore, $\alpha_r > 0$ indicates that the model outperforms the best classical approach.

## VII. RESULTS AND DISCUSSION

To provide a fair performance benchmark for our proposed approaches, we evaluate eleven classical line assignment algorithms and C-WOC on all datasets. We keep the evaluation dataset out of the training data entirely and evaluate the model's performance on that dataset. In this manner, we perform a full cross-validation for all datasets.

In Table II we show the average accuracy for eleven different classical algorithms as well as their combination via C-WOC and our DIST, E-DIST and E-WOC approaches. The accuracy metric is calculated for each trial in the evaluation dataset and then averaged across all trials in that dataset. For each dataset the accuracy of the best performing classical approach is underlined, these accuracy values were used to calculate the relative accuracies shown in Fig. 4 and used for the ablation studies in Section VIII. The relative accuracy is the difference between our model accuracy and the accuracy of the best performing classical model (see (3)). This presents the most challenging comparison and could be considered somewhat unrealistic as a practitioner looking for the best way to automatically correct their fixation data would have no way of knowing which algorithm performs best for their data since they would have no ground truth data and can thus not compare the different algorithms. We nevertheless chose this measure in order to demonstrate that our proposed approach is likely to be the best default choice in most

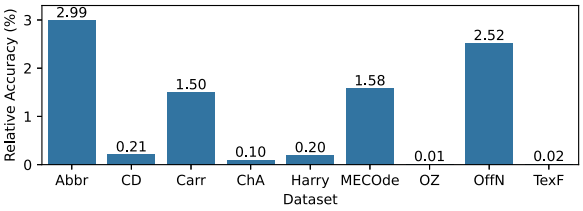



Fig. 4. Cross-validation using E-WOC evaluated using average accuracies relative to the accuracy of the best classical algorithm for each dataset. The reported accuracy is based on allocating three votes to E-DIST in the voting pool.

scenarios. As it can be seen, E-WOC outperforms all classical algorithms (including C-WOC) on all datasets. Note that for the TexF and ChA, the absolute accuracy achieved by C-WOC is already 99% or higher, since these datasets are the result of experiments using only two lines of text in their stimuli, there is little room for improvement over the classical approach. The strongest relative improvements achieved by E-WOC are seen for the Abbr, MECOde and OffN datasets. It is notable that many of the classical algorithms show good performance for at least some of the datasets in question with the overall best performing classical approach being C-WOC, closely followed by *cluster* and *merge*. Furthermore, it can be seen that E-DIST achieves the highest accuracy on three datasets, emerging as the second-best method.

As is shown in Fig. S4, the performance dependence on the number of DIST instances in E-DIST is different for each dataset. The accuracy reported in Table II is based on using three DIST instances for each normalization scheme, therefore six instances in total. Note that the use of six instances does not present a significant computational burden since the inference time for each instance is small and the majority of users in the field are unlikely to want to correct more than a few thousand trials at a time. Therefore, a small gain in accuracy is likely well worth the extra computation. See Table S1 for a comparison. A single DIST model using the fully normalized fixation features as well as the second stream information as input outperforms the classical approaches on five datasets with OffN showing the lowest accuracy. OffN is the only dataset containing paragraph breaks and is thus quite different from the other datasets.

In Fig. 4 we show the relative accuracy for each dataset achieved by E-WOC. As it can be seen E-WOC outperforms the classical approaches on all datasets with Abbr, OffN and MECOde showing the largest gains.

In Fig. 5 we show how E-WOC performance depends on how many votes are allocated to E-DIST. Note the number of votes allocated to all other algorithms is kept at one (See Fig. S5 for different vote allocations). The performance of E-WOC strongly depends on this vote allocation, with four datasets showing diminishing performance gains and the rest of the datasets showing decreasing performance as the number of votes is increased. As can be expected for datasets where E-DIST achieves lower accuracy than C-WOC, the performance decreases as the classical algorithms loose relative importance in the voting pool.

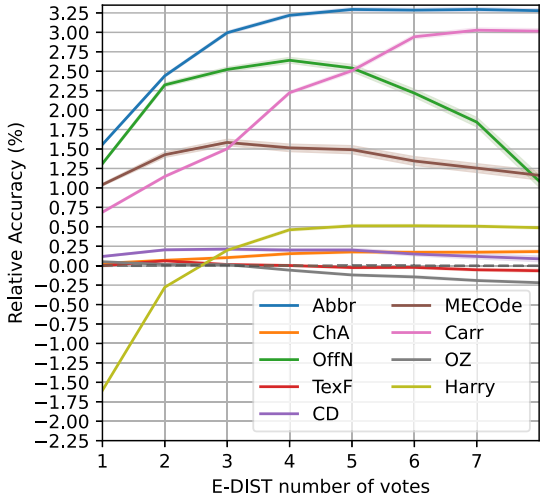



Fig. 5. Relative accuracy for E-WOC depending on the number of votes allocated to E-DIST model, which uses six DIST instances. To take into account the differences in achieved accuracy caused by which DIST instances are used in E-DIST, the data shown is the result of running the experiment 15 times for each dataset with the included instances being chosen at random (uniform probability) for each repetition.

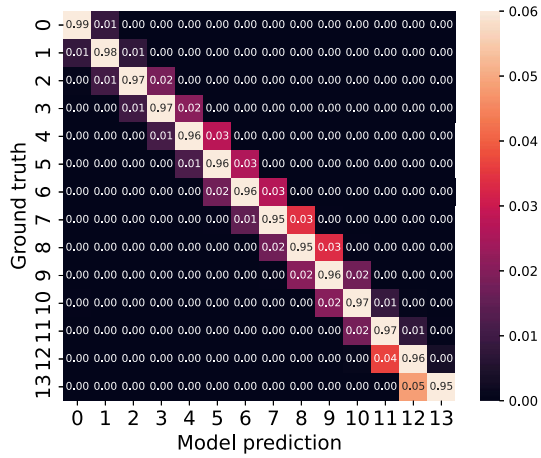



Fig. 6. Confusion matrix for all datapoints in all datasets combined. The results are based on an E-WOC with three votes allocated to E-DIST which uses six DIST instances. Note that the values are normalized for each row of the confusion matrix and the colormap is restricted to between 0 and 0.06 to better illustrate the mistakes.

We show the confusion matrix for E-WOC for all datasets in Fig. 6. As the confusion matrix shows, virtually all mistakes are misassignments by a single line. As expected from the high accuracies achieved on all datasets, the vast majority of fixation points get assigned to the correct lines.

## VIII. Ablation Studies

To further analyze the model's performance, we carry out a range of ablation studies. To better understand how each investigated factor of our model architecture and data pipeline affects the performance on the various datasets a full cross-validation is carried out for every ablation study. Each configuration is run at least ten times with all non-pretrained weights of the model being

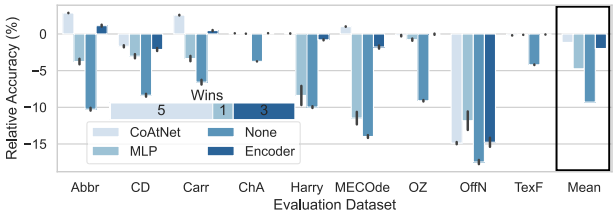



Fig. 7. Cross-validation for DIST model without the second input stream (labeled None), and for the different ways of encoding the trial related information, labeled Encoder, MLP and CoAtNet, respectively. Bar height gives average relative accuracy (relative to the best classical model for each dataset) of all experiments with a given configuration. Gray line indicates the standard error. The horizontal stacked bar chart labeled *Wins* indicates for how many datasets each configuration gives the highest relative accuracy.

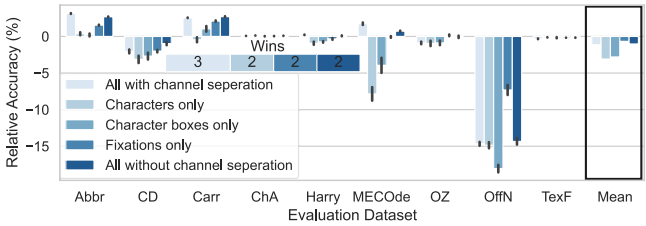



Fig. 8. DIST performance for using different input images as the second input stream as input to the CoAtNet.

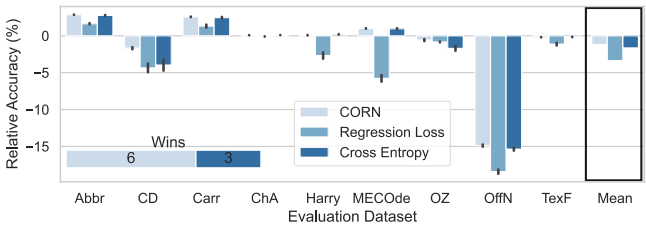


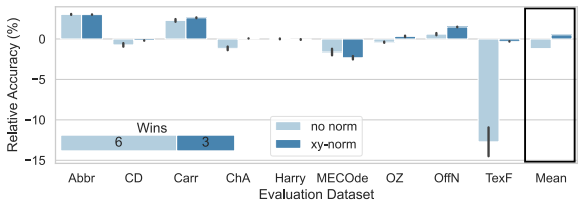

Fig. 9. Cross-validation for DIST with training done by using different loss functions to supervise.

randomly initialized each time (Kaiming initialization [43]). We present the average accuracy relative to the best performing classical approach and the standard error associated with the repeated experiments. The ablation studies are carried out for a single DIST model for each experiment, rather than an ensemble of models. This is done to show the variance in the performance for the DIST model and to more clearly show the effects of each investigated factor.

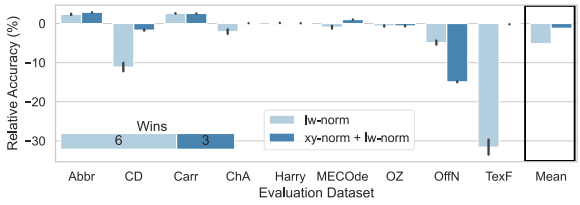
First, the effect of utilizing the dual input stream is investigated by training the model to predict line assignments based on the fixation information alone as well as with different methods of adding additional information to the main model input. In addition to the CoAtNet approach described in Section IV, we experiment with creating a sequence of bounding box coordinates of all characters on the page used as stimulus during the trial. Since this sequence length is not the same as the number of fixations in the trial, two approaches of preparing the bounding box information are explored. The first approach is to use a simple MLP consisting of two fully connected layers to first project the bounding box *x*-*y* coordinates into a single value for each step in the sequence of characters and then passing the result to a second linear layer that projects the sequence dimension to half the embedding dimension of the main encoder. The second approach is to use a separate encoder-only transformer model to encode the projected bounding box coordinates. As with the CoAtNet-based approach, this results in a single encoding vector for one sequence of fixations, which then gets repeated for every step in the fixation sequence. For all methods of including the second input stream the resulting tensor is concatenated with the projected fixation related information from the first input stream and fed into the main encoder. In Fig. 7 we show that the model greatly benefits from using a dual input stream approach with the accuracy dropping strongly for all datasets when only the fixation related input stream is used. DIST is outperformed by the classical algorithms for all datasets when no 2nd input stream is utilized. The MLP encoding approach only outperforms the classical approaches for one dataset. The encoder character bounding boxes encoding method outperforms the classical algorithm for three. Overall, the CV-based approach greatly outperforms all other approaches, achieving the highest accuracy on five datasets. The requirement of a second input stream does not present a restriction of the model since most classical models also rely on page related information to correct fixations and this information is typically recorded during the eye-tracking experiment and thus readily available.

In Fig. 8 we show how the DIST model performs when different kinds of page and fixation related information are fed into the model as a second input stream. Since the pretrained CoAtNet model that is used to encode this information expects an input with three channels, the single channel input rendering is duplicated three times in the channel dimension before being fed into the model. As it can be seen, using a scatter plot of the fixation positions without any text or bounding box information shows much better performance than using only the characters or character bounding box images for the OffN dataset. The magnitude of this difference pushes up the average performance for this input type. However, for most of the datasets the combination of all three types of information results in the best performance. The concatenation of all three renderings in the channel dimension and the non-concatenated color rendering give roughly equal average accuracy but the former achieving the highest accuracy on more datasets. OffN is a particularly challenging dataset, as it is the only one with paragraph gaps in the stimulus material, resulting in a very different pattern of fixation sequences.

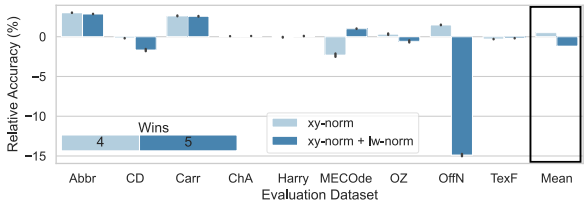
Fig. 9 shows how the model performs depending on which loss function is used to supervise the model during the training process. The rank consistent CORN loss produces the best overall performance as judged by the number of datasets on which it leads to the highest accuracy, closely followed by a standard cross-entropy loss, which does not take the ordering information into account. Both the CORN and the cross-entropy loss functions achieve similar average accuracies across the datasets. The regression loss, which is a simple mean squared error for treating the line assignment as a continuous number between zero and the maximum line in all datasets, does poorly on most datasets, showing the lowest average accuracy and no wins.




(a) Comparison of performance for using only xy-norm.




(b) Effect of using only lw-norm compared to using both xy-norm and lw-norm.




(c) Effect of using only xy-norm compared to using both xy-norm and lw-norm.

Fig. 10. Effect on relative accuracy of using different normalization schemes for the fixation points of the first input stream. Note that when both schemes are applied, the xy-norm is always applied first.

Fig. 10 shows the effect of using different normalization schemes for the fixation coordinates on the DIST model performance on the various datasets. As it can be seen, the chosen normalization scheme has a major effect on model performance for some datasets with the CD, OffN and MECOde seeing the biggest effect. Fig. 10(a) shows that if lw-norm is not used, the effect of using xy-norm is small for five datasets, while retaining an overall positive effect on the mean of all relative accuracies.

Fig. 10(b) shows the effect of using lw-norm with and without xy-norm. It can be seen that most datasets benefit from this combination while the OffN dataset sees a strong drop in relative accuracy.

Fig. 10(c) shows the effect of using xy-norm with and without lw-norm. As it can be seen, the difference in performance is very pronounced for the MECOde, CD and OffN datasets. The MECOde dataset performs very poorly without this normalization step while both the OffN and CD datasets see a strong boost when the line height and width normalization is not used.

The above analysis shows that none of the schemes or its combinations is a clear best choice for all datasets. This motivates the utilization of an ensemble-based approach which can make use of the benefits of both approaches and give good performance on all datasets when used in E-WOC as can be seen in Fig. 4. See Section IV for an explanation of how the different normalization schemes are used in the E-DIST model and how this is used as part of E-WOC.

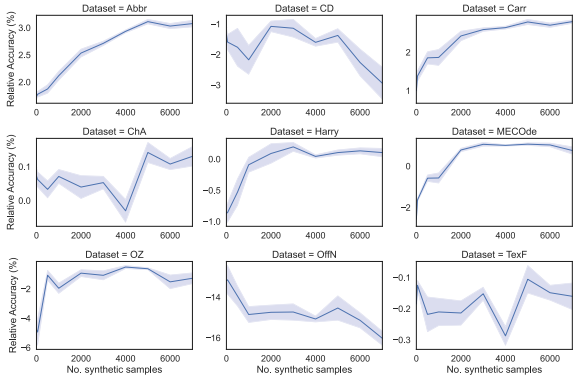



Fig. 11. Effect of using different amounts of synthetic data during training on the relative accuracy. The thick blue line indicates the mean of repeated experiments while the shaded area indicates the standard error. Note, the scale of the relative accuracy axis is not kept the same across the subplots to show the trend for each dataset.

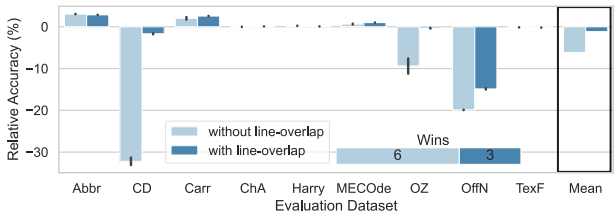



Fig. 12. Effect of adding the line-overlap feature as additional fixation related information to the first input stream.

In Fig. 11 shows how the relative accuracy develops depending on the number of synthetically created trials in the training data. Please see Section III and Section S3 for an explanation on how these trials can be generated. As it can be seen, the Abbr, Carr, Harry and MECOde datasets benefit significantly from adding at least 2000 synthetic trials to the training data. OffN is the only dataset where adding any synthetic trials to the training data results in a significant drop in relative accuracy while the accuracy of the remaining datasets is largely unaffected. As it can be seen in Table I in Section III MECOde is the only dataset with 14 line long stimulus texts, therefore adding synthetically created trials with the same number of lines will likely help the model learn such assignments.

Fig. 12 shows how DIST performs with and without the line-overlap feature in the first input stream. While for most datasets the addition of the line overlap feature does not change the relative accuracy by much, the CD, OZ and OffN see a large increase in performance. This results in an overall positive effect of adding the overlap feature. Since the overall amount of training data is limited, the model clearly benefits from being given this additional information.

## IX. Conclusion

We have introduced DIST, a dual input stream architecture using encoder-only transformer model to assign fixations from eye-tracking data collected during passage reading to their most appropriate line of text. The dynamically changing vertical drift of the recorded gaze positions makes this line assignment a crucial processing step. Our architecture utilizes fixation related information as well as trial related information to map each fixation

to a line index. We evaluate our model as well as eleven classical approaches found in literature on data from nine eye-tracking studies. It is demonstrated that an ensemble of DIST models using fixation data normalized in two different ways combined with the classical algorithms in a WOC approach outperforms all classical approaches on all datasets. The high performance of the ensemble approach is attributed to the finding that for some datasets a single DIST model performs very differently depending on how the fixation coordinates are normalized. In addition, we show that while many classical algorithms show impressive performance, the achieved accuracy varies greatly, depending on the dataset. Our combined approach, in contrast shows robust performance on all datasets and is hence a safe choice for any fixation alignment dataset. The addition of a line overlap feature for each fixation point emerges as crucial for achieving high accuracy for some of the datasets. It is also demonstrated that the use of a second input stream in addition to the fixation related features greatly benefits the model performance with a CV-based approach emerging as particularly effective. We furthermore show that the inclusion of synthetic data in the training dataset is beneficial for some datasets. Overall, our approach presents a considerable improvement over previously published classical approaches in terms of accuracy and robustness to differences in stimulus settings. This makes the presented method a safe first choice for practitioners, enabling them to carry out and analyze eye-tracking studies with larger amounts of text without being limited by the bottleneck of human corrected line assignments.

## REFERENCES


[1] S. B. Hutton, "Eye tracking methodology," in *Eye Movement Research: An Introduction to Its Scientific Foundations and Applications*, C. Klein and U. Ettinger, Eds., Cham, Switzerland: Springer, 2019, pp. 277–308.

[2] P. Raatikainen, J. Hautala, O. Loberg, T. Kärkkäinen, P. Leppänen, and P. Nieminen, "Detection of developmental dyslexia with machine learning using eye movement data," *Array*, vol. 12, Dec. 2021, Art. no. 100087.

[3] K. Rayner, "The 35th sir frederick bartlett lecture: Eye movements and attention in reading, scene perception, and visual search," *Quart. J. Exp. Psychol.*, vol. 62, no. 8, pp. 1457–1506, Aug. 2009.

[4] M. M. Janković, "Biomarker-based approaches for dyslexia screening: A review," in *Proc. IEEE Zooming Innov. Consum. Technol. Conf.*, 2022, pp. 28–33.

[5] E. C. Dias et al., "Neurophysiological, oculomotor, and computational modeling of impaired reading ability in schizophrenia," *Schizophrenia Bull.*, vol. 47, no. 1, pp. 97–107, Jan. 2021.

[6] S. Eraslan, V. Yaneva, Y. Yesilada, and S. Harper, "Web users with autism: Eye tracking evidence for differences," *Behav. Inf. Technol.*, vol. 38, no. 7, pp. 678–700, Jul. 2019.

[7] J. W. Carr, V. N. Pescuma, M. Furlan, M. Ktori, and D. Crepaldi, "Algorithms for the automated correction of vertical drift in eye-tracking data," *Behav Res*, vol. 54, no. 1, pp. 287–310, Feb. 2022.

[8] V. I. Adedeji, J. A. Kirkby, M. R. Vasilev, and T. J. Slattery, "Children's reading of sublexical units in years three to five: A combined analysis of eye-movements and voice recording," *Sci. Stud. Reading*, vol. 28, no. 2, pp. 1–20, 2023.

[9] K. C. Scherr, S. J. Agauas, and J. Ashby, "The text matters: Eye movements reflect the cognitive processing of interrogation rights," *Appl. Cogn. Psychol.*, vol. 30, no. 2, pp. 234–241, 2016.

[10] M. R. Vasilev, V. I. Adedeji, C. Laursen, M. Budka, and T. J. Slattery, "Do readers use character information when programming return-sweep saccades?," *Vis. Res.*, vol. 183, pp. 30–40, Jun. 2021.

[11] I. T. C. Hooge, D. C. Niehorster, M. Nyström, R. Andersson, and R. S. Hessels, "Is human classification by experienced untrained observers a gold standard in fixation detection?," *Behav. Res.*, vol. 50, no. 5, pp. 1864–1881, Oct. 2018.

[12] A. L. Cohen, "Software for the automatic correction of recorded eye fixation locations in reading experiments," *Behav. Res.*, vol. 45, no. 3, pp. 679–683, Sep. 2013.

[13] J. W. Carr, V. N. Pescuma, and D. Crepaldi, "Algorithms for assigning fixations to lines of text in multiline passage reading," *21st Eur. Conf. Eye Movements*, Leicester, England, U.K., Aug. 2022. [Online]. Available: https://joncarr.net/d/carr_pescuma_crepaldi-2022-ecem-slides.pdf

[14] S. Schroeder, "popEye - An R package to analyse eye movement data from reading experiments," *GitHub repository*, 2019. [Online]. Available: https://github.com/sascha2schroeder/popEye

[15] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[16] T. K. Vintsyuk, "Speech discrimination by dynamic programming," *Cybern Syst Anal*, vol. 4, no. 1, pp. 52–57, Jan. 1968.

[17] O. Špakov, H. Istance, A. Hyrskykari, H. Siirtola, and K.-J. Räihä, "Improving the performance of eye trackers with limited spatial accuracy and low sampling rates for reading analysis by heuristic fixation-to-word mapping," *Behav. Res.*, vol. 51, no. 6, pp. 2661–2687, Dec. 2019.

[18] D. Glandorf and S. Schroeder, "Slice: An algorithm to assign fixations in multi-line texts," *Procedia Comput. Sci.*, vol. 192, pp. 2971–2979, 2021.

[19] L. Rello and M. Ballesteros, "Detecting readers with dyslexia using machine learning with eye tracking measures," in *Proc. 12th Int. Web All Conf.*, Florence Italy: ACM, 2015, pp. 1–8.

[20] I. Vajs, V. Ković, T. Papić, A. M. Savić, and M. M. Janković, "Dyslexia detection in children using eye tracking data based on VGG16 network," in *Proc. 30th Eur. Signal Process. Conf.*, 2022, pp. 1601–1605.

[21] Y. Berzak, B. Katz, and R. Levy, "Assessing language proficiency from eye movements in reading," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, Volume 1, M. Walker, H. Ji, and A. Stent, Eds., New Orleans, Louisiana, USA, Jun. 2018, pp. 1986–1996, doi: 10.18653/v1/N18-1180.

[22] S. Ahn, C. Kelton, A. Balasubramanian, and G. Zelinsky, "Towards predicting reading comprehension from gaze behavior," in *Proc. ACM Symp. Eye Tracking Res. Appl.*, New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–5.

[23] D. R. Reich, P. Prasse, C. Tschirner, P. Haller, F. Goldhammer, and L. A. Jäger, "Inferring native and non-native human reading comprehension and subjective text difficulty from scanpaths in reading," in *Proc. Symp. Eye Tracking Res. Appl.*, New York, NY, USA: Association for Computing Machinery, 2022, pp. 1–8.

[24] S. Deng, D. R. Reich, P. Prasse, P. Haller, T. Scheffer, and L. A. Jäger, "Eyettention: An attention-based dual-sequence model for predicting human scanpaths during reading," in *Proc. ACM Hum.-Comput. Interact.*, vol. 7, no. ETRA, pp. 162:1–162:24, May 2023.

[25] V. I. Adedeji, M. R. Vasilev, J. A. Kirkby, and T. J. Slattery, "Return-sweep saccades in oral reading," *Psychol Res*, vol. 86, no. 6, pp. 1804–1815, 2022.

[26] V. Goldenberg, "Measured and perceived impact of noise during reading. an eye tracking study," MSc. Thesis, Department of Psychology, Bournemouth Univ., Poole, U.K., Aug. 2022.

[27] M. R. Vasilev, S. P. Liversedge, D. Rowan, J. A. Kirkby, and B. Angele, "Reading is disrupted by intelligible background speech: Evidence from eye-tracking," *J. Exp. Psychol.: Hum. Percep. Perform.*, vol. 45, no. 11, pp. 1484–1512, 2019.

[28] T. J. Slattery and M. R. Vasilev, "An eye-movement exploration into return-sweep targeting during reading," *Atten Percept Psychophys*, vol. 81, no. 5, pp. 1197–1203, Jul. 2019.

[29] H. S. Seymour, "Using eye-tracking technology to measure the effect of political ideology on comprehension of online news articles," MSc. Thesis, Bournemouth Univ., Poole, U.K,, Aug. 2022.

[30] N. Siegelman et al., "Expanding horizons of cross-linguistic research on reading: The multilingual eye-movement corpus (MECO)," *Behav. Res*, vol. 54, pp. 2843–2863, Feb. 2022.

[31] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," Sep. 2016, *arXiv:1609.07843*.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, Volume 1, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota, USA, Jun. 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

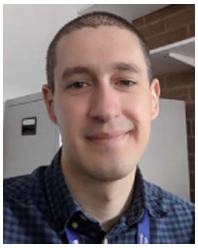
[33] A. Vaswani et al., "Attention is all you need," Dec. 2017, *arXiv: 1706.03762*.

[34] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and LSTM encoder decoder models for ASR," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 8–15.

[35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning, Ser. Adaptive Computation and Machine Learning*. Cambridge, MA, USA: The MIT Press, 2016.
[36] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "CoAtNet: Marrying convolution and attention for all data sizes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 3965–3977. Accessed: Jul. 3, 2024. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/20568692db622456cc42a2e853ca21f8-Abstract.html
[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[38] R. Wightman, "PyTorch image models," *GitHub repository*, 2019, doi: 10.48550/arXiv.2111.08851.
[39] X. Shi, W. Cao, and S. Raschka, "Deep neural networks for rank-consistent ordinal regression based on conditional probabilities," 2022, *arXiv: 2111.08851*.
[40] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv: 1912.01703*.
[41] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, "Well-read students learn better: On the importance of pre-training compact models," 2019, *arXiv: 1908.08962*.
[42] "Eyekit," 2019.[Online]. Available: https://github.com/jwcarr/eyekit
[43] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," 2015, *arXiv: 1502.01852*.

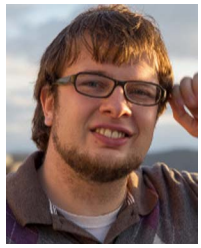
**Thomas M. Mercier** received the BSc degree in nanoegineering from the University of Duisburg-Essen in Germany, and the MSc and PhD degrees in nanoelectronics from the University of Southampton, U.K. He is currently employed as a postdoctoral researcher working on modeling of eye-movements during reading using Machine Learning techniques.

**Marcin Budka** received the dual MSc/BSc degrees in finance from the Katowice University of Economics, in 2003, the BSc degree in computer science from the University of Silesia, in 2005, and the PhD degree in computational intelligence from Bournemouth University, 2010. Between 2003 and 2007 he was working as an engineer, project manager and team leader in a smart-metering start-up, before pursuing an academic career. In the years 2011-2012 he was appointed as a visiting research fellow with the Wrocław University of Technology. Between 2015 and 2017 he was the head of research with the Department of Computing and Informatics, Bournemouth University.

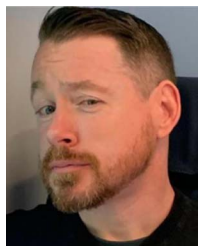


**Martin R. Vasilev** received the BA degree in psychology from the University of Sofia, Bulgaria, in 2013, the MSc degree in clinical linguistics from the University of Potsdam, Germany, in 2015, and the PhD degree from Bournemouth University, in 2018, focusing on eye-movement control and auditory distraction. He has been working as a lecturer since 2020.

**Julie A. Kirkby** received the PhD degree in cognitive psychology under the supervision of Prof. Simon P Liversedge from the University of Southampton examining dyslexia. Her research interests fall within the field of cognitive psychology, in particular eye movements, reading and visual cognition. The primary goal of her research is to increase understanding of the causes and outcomes of developmental dyslexia. Since 2010, she has worked at Bournemouth University and established the Reading Research group.



**Bernhard Angele** received the MA degree, in 2009 and the PhD degree on parafoveal processing in reading from the University of California San Diego, in 2013. His research interests primarily focus on eye movements during skilled adult reading and language processing. Specifically, he has been studying the effect of parafoveal preview on processing and reading performance.



**Timothy J. Slattery** received the BSc degree in psychology with honors from Buffalo University, in 1996, and the PhD degree in cognitive psychology from the University of Massachusetts, Amherst, in 2001, with minors in Quantitative Analysis. Post-PhD, he undertook a postdoc position with UC San Diego with Keith Rayner before becoming an assistant professor with the University of South Alabama, in 2011, where he established a new Psycholinguistics Lab. He joined Bournemouth University's Psychology Department in 2015 as part of their cognitive and neuroscience research team.