Representing Online Handwriting for Recognition in Large Vision-Language Models

Anastasiia Fadeeva * 1 Philippe Schlattner * 1 Andrii Maksai § 1 Mark Collier 1 Efi Kokiopoulou 1 Jesse Berent 1 Claudiu Musat § 1

Abstract

The adoption of tablets with touchscreens and styluses is increasing, and a key feature is converting handwriting to text, enabling search, indexing, and AI assistance.

Meanwhile, vision-language models (VLMs) are now the go-to solution for image understanding, thanks to both their state-of-the-art performance across a variety of tasks and the simplicity of a unified approach to training, fine-tuning, and inference. While VLMs obtain high performance on image-based tasks, they perform poorly on handwriting recognition when applied naively, i.e., by rendering handwriting as an image and performing optical character recognition (OCR).

In this paper, we study online handwriting recognition with VLMs, going beyond naive OCR. We propose a novel tokenized representation of digital ink (online handwriting) that includes both a time-ordered sequence of strokes as text, and as image. We show that this representation yields results comparable to or better than state-of-the-art online handwriting recognizers. Wide applicability is shown through results with two different VLM families, on multiple public datasets. Our approach can be applied to off-the-shelf VLMs, does not require any changes in their architecture, and can be used in both fine-tuning and parameterefficient tuning. We perform a detailed ablation study to identify the key elements of the proposed representation.

1. Introduction

Digital alternatives to writing on paper are expanding. One of the key features users need is to seamlessly transition between modalities and turn their handwritten notes into printed text (Riche et al., 2017). This feature depends on the quality of the underlying handwriting recognition models.

Historical perspective. Approaches to handwriting recognition have evolved over time alongside similar problems in speech recognition and OCR, going from segment-and-decode models (Hu et al., 1996) to RNNs (Carbune et al., 2020; Graves et al., 2009) to Transformer-based approaches (Dhiaf et al., 2023). However, as in other modalities, it is still far from being solved, especially in more complex cases that involve whole-page note-taking, math expression recognition, and scripts with small amounts of training data. This is also visible through the variety of model architectures used to solve these problems (Xie et al., 2023).

Why use VLMs? LLMs and VLMs (Brown et al., 2020; Touvron et al., 2023; Chen et al., 2023a) have shown impressive performance in a variety of tasks and across different modalities and they can offer multiple benefits if they can be used for solving handwriting recognition problems (or any other target domain). The obvious one is a potential quality improvement coming from their scale and underlying language model. Their simple design allows fine-tuning a single model end-to-end using standard and widely available tools in contrast to standard multi-step recognition models, ex. (Carbune et al., 2020). Finally, they allow seamlessly mixing multiple handwriting tasks expanding the plethora of tasks they can already perform.

How to use VLMs? To be able to instill VLMs with hand-writing recognition, one needs a representation of digital ink that is suitable for use with VLMs. A straightforward approach is to simply provide a rendering of digital ink as an input image and perform OCR. However, for handwriting recognition, OCR-only performance is subpar to the quality of specialized online handwriting recognition models that operate on the digital ink as a time-ordered sequence of points (Xie et al., 2023).

Our work. The focus of our work is the representation of digital ink for VLMs that is applicable across different datasets and model families, and yields comparable performance to state-of-the-art task-specific models. To our

^{*}Equal contribution. \$Technical lead. ¹Google Research. Correspondence to: Anastasiia Fadeeva <fadeich@google.com>, Andrii Maksai <amaksai@google.com>.

knowledge, we are the first to explore stroke-based representations in VLMs for handwriting recognition.

In our search for a widely applicable representation of digital ink we explore two main options: based on images and time-ordered sequences of points. We look for the optimal way of rendering ink into an image and of discretizing the sequence of points into a sequence of text tokens that can be consumed by VLMs. We show how these representations should be combined together to achieve optimal performance.

We find that it is possible to obtain good recognition quality while representing digital ink as text. This is unlike some other modalities like audio (Rubenstein et al., 2023), where adding a new modality into an existing model requires extending the token dictionary with the tokens of the new modality as well as modifying the model architecture. This means that our approach does not require any changes to existing models and enables adding handwriting recognition capabilities to pre-trained VLMs by fine-tuning or even parameter-efficient tuning, which further preserves original capabilities of the model. Our findings generalize across two model families and several different handwriting recognition datasets.

To sum up, our main contributions are as follows:

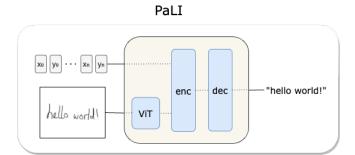
- We propose a representation for digital ink that combines images and time-ordered token sequences, which is suitable for use with VLMs;
- We show that such a dual representation is vital for achieving performance comparable to state-of-the-art task-specific handwriting recognition models; to our knowledge this is the first work to explore stroke-based representations for online handwriting recognition in VLMs;
- We show that our representation is suitable for finetuning or parameter-efficient tuning of a pre-trained VLM and does not require changing model architecture or vocabulary;
- We perform a detailed ablation study to find the best way of representing digital ink as images and as token sequences.

2. Background

This paper focuses on **online handwriting recognition**, meaning the input includes spatial and time information. Related work on the topic is summarized in Section 5. We denote a **stroke** s as a sequence of triplets (x, y, t) where x and y are coordinates on the screen and t is time information (Carbune et al., 2020). We denote an **ink** $I = [s_0, \ldots, s_n]$ as a sequence of written strokes (aka pen-down strokes). The

input of our model is an ink I and the **output** is the text written in the ink.

The VLM architectures that we use in this work are PaLI (Chen et al., 2023c;a) and PaLM-E (Driess et al., 2023), transformer-based models (Vaswani et al., 2017). PaLI is an encoder-decoder model that combines image and text representations in the encoder as shown in Fig. 1. PaLM-E is a decoder-only model that combines image and textual embeddings in its input. The main difference between the two is the presence of transformer encoder in PaLI. In our experiments with PaLM-E, we use non-causal mask on the the input as described in Wang et al. (2022) given that Raffel et al. (2020) has shown that this approach provides similar quality to encoder-decoder models.



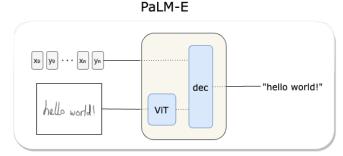


Figure 1. PaLI and PaLM-E architectures for ink recognition.

Text tokenization in VLMs plays an important role in the success of the training. Most models use the Byte-Pair Encoding (BPE) procedure (Kudo & Richardson, 2018), however there are different approaches to tokenizing numbers in text. It was shown in Liu & Low (2023) that tokenizing each digit separately helps with arithmetic operations, at the price of making the input sequence longer, as opposed to a BPE vocabulary that contains multi-digit tokens. We consider both options as PaLI utilizes standard BPE and PaLM-E uses the single digit tokenization strategy.

3. Method

In our work we focus on representation choices that make online handwriting recognition suitable for VLMs. In this section we describe a range of possibilities to represent online handwriting as a sequence of time-ordered tokens and as images. We provide the recipe that we used used in the experiments and the intuition behind it.

3.1. Sequence representation

In its original form online inks are represented as sequences of strokes with coordinates x,y and time information t, sampled. This could be encoded in text format as: "stroke $x_{0,0}$ $y_{0,0}$ $t_{0,0}$ $x_{0,1}$ $y_{0,1}$ $t_{0,1}$...". Even though this representation can be directly fed into a VLM, we found it crucial to consider time sampling, scale normalization, coordinate representation, discretization, and token dictionary. Below we outline each of these and Figure 2 depicts them.

Time sampling, scale normalization, and token dictionary choice have an immediate effect on the length of the sequence used to represent this ink and Table 1 shows how doing them allows to reduce the sequence length significantly. The effect of other choices, namely coordinate representation, codebook, and token dictionary, is studied in Section 4.5.2.

Time sampling. In order to normalize sampling frequency among different devices and reduce the sequence length, we resample points at regular time intervals within each stroke. It is critical to pick an appropriate time delta because with high values an ink can lose important writing details (see Appendix A for examples). After resampling, all points within each stroke have the same time delta between each other, so we can omit value t for the text representation (information about the duration of time between strokes is thus discarded). This allows us to significantly shorten the sequences due to lower number of points and an ability to omit value t from text, as seen in Tab. 1.

Table 1. Effect of normalization and tokenization on the sequence length. Median numbers with mT5 (Xue et al., 2021) tokenizer on MathWriting dataset (Gervais et al., 2024). The first line reports the length when representing the ink as the original sequence of x, y, and t coordinates as described in the beginning of Section 3.1 and each following line cumulatively adds some form of processing described in the section. After time sampling, we remove the t from representation as described in the appropriate paragraph, and after scale normalization, we round x and y to the nearest integer.

Representation	# Points	# Tokens
Original x,y,t	313	2692
+Time sampling	178	1954
+Scale normalization	178	381
+Extended token dictionary	178	367

Scale normalization. We scale and shift, preserving the aspect ratio, so that all points fit into the range between 0 and N (where N is the ViT encoder image size). That helps to account for potential differences in the input canvas size and to reduce the sequence length through smaller point

coordinates that have a more compact text representation. We use the information about the scaling range when doing discretization, as described below.

Coordinate representation. We represent points of the ink using the offsets of coordinates from one time-step to the next rather than the absolute coordinates: $(x_t^r, y_t^r) = (x_t, y_t) - (x_{t-1}, y_{t-1})$. In the ablation study, we compare the absolute and relative representations. See Fig. 2 for more details.

Discretization codebook. We use two values to represent each point (x, y) in the ink, by rounding the normalized x and y coordinate to the nearest integer. In the ablation study, we show that this approach leads to better accuracy than using a learned codebook of offsets, similar to Ribeiro et al. (2020), where each offset is represented by a single token.

Token dictionary. We represent each point (x,y) directly as text, as opposed to extending the token dictionary (Rubenstein et al., 2023). We use a separator expression to indicate the beginning of the new stroke, as "<stroke> 2 1 2 2 ... <stroke> 3 1 3 2 3 3 ...". In the ablation study, we compare this to the approach where the token dictionary of the model is extended with new tokens, which has similar performance but requires changes to the model.

Note that when the input is provided as text, we are effectively performing tokenization twice. First, inks are converted to a model-agnostic sequence of indices in text format, as shown above. Then, these indices are converted to a final sequence of tokens accepted by the VLM by applying the model-specific tokenizer, e.g. using Byte-Pair Encoding. As a result, each token of the initial tokenization may correspond to several text tokens. Therefore, we observe that bigger indices coming out of the first tokenization result in increased sequence length of the second tokenization.

3.2. Image representation

Rendering ink as an image enables conveying different information about the underlying digital ink, with the a simple approach being rendering black strokes on white background. Alternatively, (Kag, 2023) proposed to render time information in image channels.

An added complexity in representing images of handwriting is the varying size and dimensions. There are several solutions to account for these, including stretching the sample to square, rendering respecting the aspect ratio, writing the image in multiple lines (Bavishi et al., 2023), or using vision encoders that support arbitrary input image dimensions such as Pix2Struct (Lee et al., 2023) or NaViT (Dehghani et al., 2023).

In our work we use the vanilla ViT as the most popular and standard vision encoder solution. We encode speed

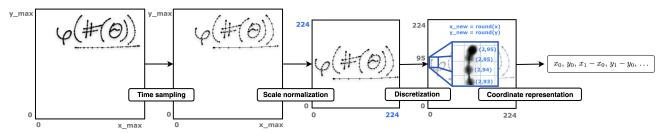


Figure 2. The full pipeline for the **sequence representation** in VLMs. This pipeline includes time sampling, scale normalization, discretization with uniform grid and representation of points with two coordinates in text.

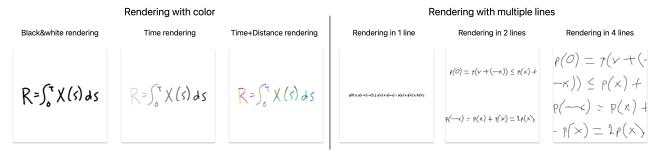


Figure 3. Examples of different rendering options. Rendering options for color – black&white, time from Eq. 1 and time+distance from Eq. 1. Examples of rendering in one, two or four lines.

information in input color channels, and render the input ink in several lines. We describe each of those below, provide examples of components of the rendering in Fig. 3 and validate this choice in Section 4.5.3.

Rendering the time and distance information. We can use the three available color channels to encode the writing direction and stroke order. To do so, we normalize the time information within the ink between 0 and 1. The example of such representation is shown in the left part of Fig. 3, and the appropriate expression is given below

$$c_{i,j}^{R} = \frac{t_{i,j} - t_{0,0}}{\max t_{m,n}} \ c_{i,j}^{G} = \frac{|dx_{i,j}|}{\max |dx_{m,n}|} \ c_{i,j}^{B} = \frac{|dy_{i,j}|}{\max |dy_{m,n}|}$$
(1)

where $t_{i,j}$ is the time value for the point j in stroke i and $dx_{i,j} = x_{i,j+1} - x_{i,j}$. Distance in x and y brings additional information about the size of the step and together with time information the model can estimate the speed. They are normalized between 0 and 1 similar to the time information. An example of this time and distance representation is shown in Fig. 3.

We show the importance of rendering both time and distance information in Section 4.5.3 where we compare it to rendering time information only in 3 color channels or using simple black-on-white rendering.

Rendering samples in multiple lines. Since handwritten samples often have very skewed aspect ratio, rendering

them in a single line on a fixed size image can often produce unreadable results. To account for this fact, they could be rendered on an image with aspect ratio 1:X, which is then split horizontally and merged vertically to produce a single square image with multiple lines, see Fig. 3, far right. In our experiments, we use X=2 and validate this choice in Section 4.5.3.

3.3. Target representation

There are multiple ways to represent the label of handwriting, that needs to be predicted (the target). Having Byte-Pair Encoding tokenization in VLMs allows for faster decoding, but can be suboptimal for recognition performance, where two similar inputs (ex. "hello" and "hallo") may correspond to two different sequences of tokens. This is especially true in case of non-vocabulary words recognition. This can be circumvented by training the model to predict the target label broken into visual blocks of interest - ex. letters in case of text, of LATEX symbols in case of math recognition.

To achieve this, in case of text recognition we use space-separated letters as the target label (ex. "h e l l o") to map visual elements (letters) to output tokens. For math expression recognition we don't use space separation as it allows a model to utilize knowledge about LATEX syntax from pretraining. We validate this decision in Section 4.5.4.

4. Experiments

We evaluate the proposed representation of online handwriting on two different VLMs and compare results to a series of baselines. In our experiments we find that the model can fall back to the image representation when the text representation of handwriting is too long. Through ablation studies we find the relative coordinate tokenizer, with text and speed rendering in images as the best-performing combination.

4.1. Models

As described in section 2, we use two main base model families, PaLI (Chen et al., 2023c) combining ViT vision encoder with mT5 text encoder-decoder, and PaLM-E (Driess et al., 2023), projecting ViT tokens into a PaLM (Anil et al., 2023) text decoder. For PaLI, we use ViT-B/16 vision encoder pretrained on JFT-300M (Sun et al., 2017) and mT5-base pretrained on Common Crawl-based dataset, totalling 700M params. For PaLM-E, we use ViT-B/16 vision encoder pretrained on CoCa (Yu et al., 2022) and PaLM 128M text decoder, totalling 500M params, pretrained on a mix of social media, webpages, books and Github (Anil et al., 2023).

Additionally, results for LoRA-tuning reported in Table 4 are based on a 5B parameter model of the same architecture as PaLI (Chen et al., 2023b). In spite of not being directly comparable to our smaller models, these results showcase the effectiveness of our method when applied through parameter-efficient tuning on larger models, where fine-tuning becomes computationally expensive. It is for this same reason that fine-tuning experiments of these larger model are omitted in this paper.

We compare our method to several baselines ranging from OCR models trained on private datasets to CTC transformer baseline which we train ourselves on the same public datasets.

State-of-the-art methods. First, we compare to a publicly available OCR API (Google Cloud, 2023). For online handwriting recognition, we compare to (Carbune et al., 2020) which uses a 5M parameters LSTM encoder with a CTC decoder (Graves et al., 2006), combined with a character-based language model. Additionally, for the VNOnDB dataset, we report the results from 1) the best-performing method from VOHTR-2018 online handwriting recognition challenge (Nguyen et al., 2018) which is Vietnamese-specific and 2) the OCR results of Le et al. (2019).

Trained baselines. We also compare the results to a Transformer encoder with a CTC decoder, similar to (Alwajih et al., 2022). We provide the architecture details in Appendix D.

4.2. Datasets

We train the models on the three public datasets – DeepWriting (Aksan et al., 2018), MathWriting (Gervais et al., 2024) and VNonDB (Nguyen et al., 2018). Dataset statistics are given in Table 2 and examples from each dataset are shown in Fig. 4.

Table 2. Dataset statistics. For MathWriting, 630k training samples includes 230k real and 400k synthetic samples.

Dataset	Language	Samples	Mean target tokens
DeepWriting	English	34k	13
MathWriting	LATEX	630k	15
VNOnDB	Vietnamese	67k	3

Figure 4. Examples from DeepWriting, MathWriting and VNonDB datasets.

4.3. Training and evaluation setup

PaLI and PaLM-E models described in Section 4.1 are fine-tuned for 200k steps with 128 and 64 batch size respectively. CTC transformer is trained for 100k steps with 256 batch size. We use a context length 1024 for PaLI and 1500 for PaLM-E covering 95th percentile of sequence lengths in all validation datasets. These numbers differ between the two models due to different text tokenizers used in each, see details in Section 2. We use an additional 256 tokens coming from the ViT encoder on 224px image in case of PaLI and 288px in case of PaLM-E. During inference, we use greedy decoding with at most 64 tokens.

For PaLI and PaLM-E, we train on a 80% MathWriting, 10% VNOnDB, 10% DeepWriting mixture of datasets (a distribution similar to proportion of the number of samples in each dataset) as our representation allows seamless mixing of tasks and it is beneficial for the model quality. We additionally report numbers when trained on all datasets separately in the AppendixB.

Similar to most literature on the topic, we report the standard Character Error Rate (CER) metric (Michael et al., 2019). We report the mean and variance over three runs for all the methods that we train. In the case of MathWriting

we calculate CER based on the dictionary of LaTeX tokens (see Appendix F). In cases where we use space-separated target representation, we remove the spaces before CER computation.

4.4. Results

Table 3 compares PaLI and PaLM-E models finetuned with our representation described in Section 3 (relative coordinates in text and speed rendering with two lines in image representation) to CTC transformer and state-of-the-art OCR and online handwriting recognition models. Our main conclusion from it is that **performance of both VLMs finetuned for handwriting recognition is comparable to or better than state-of-the-art results.** Furthermore, best results for each dataset are achieved with VLMs, with the exception of VNOnDB. For VNOnDB, the best performing model is, unlike all the other approaches, tailored to Vietnamese - it relies on specific handling of diacritics, Vietnamese-specific tokenizer, etc. We attribute the somewhat low performance of PaLM-E on Deepwriting to the small size of this dataset.

Our proposed representation is useful in both fine-tuning and parameter-efficient tuning. As visible in Table 4, fine-tuning with the 700M parameter model yields better results than LoRA-tuning 5B parameter model. However, LoRA-tuning with our representation performs better than image-based methods from Table 3.

4.5. Ablation study

In the following subsections we perform most of the ablation studies on the MathWriting dataset, which we chose due to its scale. We train PaLI for 100k steps in these experiments for shorter experiment cycles.

4.5.1. MULTIMODAL INPUT

Experiments in Table 5 show how sequence and image representations perform in recognition training with VLMs and when it is beneficial to use them in the representation. We show examples of mistakes in image-only recognition and show how they are addressed with an additional ink modality Fig. 5. We draw several conclusions: (1) **Combining** both image and ink input yields better performance than when using only one of the input modalities. This is particularly visible in the case of PaLM-E. Our hypothesis is that, due to the longer sequence length from the different tokenization of numbers in PaLM-E (see Section 2), training only on ink becomes a more challenging task than in PaLI. In this case addition of image tokens significantly improves the quality of the final model. (2) The input image helps if the sequence representation of the ink exceeds the context length. We calculated that for the inks that exceed the sequence length CER goes down from 17.38 to 11.17 with

an introduction of an image input for PaLI.

4.5.2. INK TOKENIZATION

In this ablation study we compare different approaches to tokenization from the Section 3.1. We compare them on the PaLI model with only sequence representation in order to evaluate the effect of tokenization without an influence from an image input. We draw the following main conclusions based on Table 6: (1) Representing ink with text or separate tokens yields similar performance to extendeding the vocabulary as long as the whole ink sequence fits into the model context and the dictionary is small. For the histogram-based tokenizer (similar to (Ribeiro et al., 2020), more details in the Appendix E), which uses a large dictionary, each index corresponds to several tokens when represented as text, resulting in worse performance; (2) Absolute and relative representation of the ink result in a similar recognition quality. This is also visible in Table 7. We have observed that this is dataset-specific, with English and Vietnamese recognition performing better with relative coordinates and Math recognition with absolute coordinates. We attribute this to the fact that relative representation is shift-invariant and is easier to learn, but for math expressions, which don't have a linear structure, absolute coordinates are more important.

4.5.3. IMAGE RENDERING

In the following ablation study we compare three different types of color rendering – black&white, time and time+distance; see examples in Fig. 3. We focus our attention specifically on time information as it is a distinguishing feature of online handwriting recognition task. From Table 8 we draw the conclusion that **time and distance information in color channels both contribute to a better performance**. In Appendix C we present an example of how writing order helps with recognition of an ambiguous ink.

We then explore in Table 8 the optimal number of lines for rendering. Rendering in multiple lines allows to control the size of writing, which can be small if an ink is rendered in one line (see Fig. 3) or in too many lines. We find that two lines is optimal for MathWriting dataset which has a median aspect ratio of 2.29. We conclude that the **number of lines closest to average aspect ratio in the dataset performs the best**.

4.5.4. TARGET REPRESENTATION

In Table 9 we show that space-separation of the target improves handwriting text recognition but performs poorly on mathematical expression recognition. The gap in performance on DeepWriting is especially big as this dataset contains many non-vocabulary words like "clearl" in Fig. 4.

Table 3. Comparison of our approaches to state-of-the-art methods on three public datasets. PaLI and PaLM-E are finetuned on relative coordinates in text and speed rendering in image representations. For VNonDB, DeepWriting we use space-separated target and for MathWriting targets are not space separated (see Sec. 4.5.4 for more details).

					CER↓	
Model	Training data	Language-specific	Input	MathWriting	VNOnDB	DeepWriting
(Google Cloud, 2023)	Private	-	Image	5.93	4.83	14.16
(Le et al., 2019)	Public	No	Image	-	4.10	-
(Carbune et al., 2020)	Private	No	Ink	-	4.13	6.14
MyScript(Nguyen et al., 2018)	Private	Yes	Ink	-	2.91	-
CTC transformer	Public	No	Ink	4.28 (0.06)	3.82 (0.24)	5.71 (0.2)
PaLI [ours]	Public	No	Ink+Image	4.47 (0.07)	3.04 (0.01)	4.39 (0.06)
PaLM-E [ours]	Fublic	INO	IIIK+IIIiage	4.19 (0.04)	3.27 (0.1)	6.89 (0.06)

Figure 5. PaLI recognition on four examples where prediction only on image or ink is different from the target. We compare PaLI results to the ground truth from the MathWriting dataset. Mistakes include mixing similar characters like "tau" and "T", "d" and "a". We show that those mistakes are addressed by including ink representation.

Table 4. Comparison between finetuning the 700M PaLI and LoRA-tuning PaLI-5B with proposed ink+image representation.

Train	# Train.	CER ↓		
setup	params	MathWriting	VNOnDB	DeepWriting
Full	700M	4.47 (0.07)	3.04 (0.01)	4.39 (0.06)
LoRA	27M	4.93 (0.08)	3.61 (0.16)	5.74 (0.17)

Table 5. Multimodality in PaLI and PaLM-E. In addition, we compare PaLI with 1024 and 512 tokens to see how image helps when ink is too long for context length. PaLM-E has a specific number tokenization, described in Section 2 and requires a longer sequence length than PaLI. Ink and image representations are the same as in Table 3.

ruote 5.					
			Seq	ı. len	CER↓
	Model	Input	Ink	Image	MathWriting
		Image	-	256	8.07 (0.14)
		T 1	1024	-	4.64 (0.07)
	PaLI	Ink	512	-	10.65 (0.34)
		Tala I Tarana	1024	256	4.55 (0.04)
		Ink + Image	512	256	5.89 (0.35)
		Image	-	256	4.87 (0.07)
	PaLM-E	Ink	1500	-	6.46 (0.02)
		Ink+Image	1500	256	4.22 (0.16)

In case of MathWriting we obtain much better results without space interleaving as it allows the model to utilize knowledge about LATEX syntax from pretraining.

Table 6. PaLI with different ink representations and no image input. Models with extended dictionaries are compared to text representation of inks.

	$\operatorname{CER}\downarrow$
Token type	MathWriting
Гехt	4.64 (0.07)
Extended	4.59 (0.08)
Гext	4.20 (0.02)
Extended	4.43 (0.04)
Гext	7.11 (0.08)
Extended	4.53 (0.04)
	Fext Extended Fext Extended Extended Fext

Table 7. Absolute coordinates vs relative offsets for text representation combined with an image in the PaLI model. Absolute representation performs 12% worse on DeepWriting and 4% on VNOnDB.

		CER↓	
Tokenization	MathWriting	VNOnDB	DeepWriting
Relative	4.55 (0.04)	4.55 (0.09)	4.68 (0.16)
Absolute	4.37 (0.04)	4.72 (0.0)	5.23 (0.1)

5. Related work

The **online handwriting recognition** has a long history, and ink representation plays an important role in any recognition model. In early works inks were represented as aggregations of geometric features like direction, distances, curvatures, etc described in (Jaeger et al., 2003). With more

Table 8. Different rendering options w.r.t. rendering with color and in multiple lines for PaLI model. Measured with only image input to estimate the effect of image representation without an influence of ink representation. Examples of all options that we evaluate are shown in Fig. 3.

0						
Color			CER ↓			
Time	Distance	# Lines	MathWriting			
No	No	2	13.93 (4.34)			
Yes	No	2	11.36 (1.49)			
Yes	Yes	2	8.07 (0.14)			
Yes	Yes	1	11.15 (1.69)			
Yes	Yes	2	8.07 (0.14)			
Yes	Yes	4	14.94 (1.81)			

Table 9. Comparison on space interleaving in target on PaLI relative ink+image. This target representation performs well on VNOnDB and DeepWriting datasets but decreases the quality on MathWriting where multiple characters in the target frequently represents one symbol in ink.

	CER↓		
Spaces	MathWriting	VNOnDB	DeepWriting
Yes	11.03 (0.36)	4.55 (0.09)	4.68 (0.16)
No	4.55 (0.04)	5.02 (0.26)	22.3 (1.79)

complex machine learning systems, manual feature engineering became unnecessary as models with sufficient data can learn relevant features from raw representations, for instance in computer vision (Krizhevsky et al., 2012) and in NLP (Mikolov et al., 2013).

One of the main challenges in online handwriting recognition is aligning the input ink with the target text as they have different lengths. Historically, Hidden Markov Models were used for this task (Hu et al., 1996). They were subsequently replaced by deep neural networks like LSTMs (Hochreiter & Schmidhuber, 1997) and Transformers (Vaswani et al., 2017) with **connectionist temporal classification** (CTC) loss. This solution was initially proposed for speech recognition in (Graves et al., 2006) and it trains the segmentation and classification together. Another approach is to use the encoder-decoder model (also known as seq2seq) where an encoder produces a sequence of states based on the input and an autoregressive decoder uses those states in crossattention. This method has been widely adopted in speech recognition in recent years (Prabhavalkar et al., 2017; Pundak et al., 2018; Chiu et al., 2017; Hannun et al., 2014). Similarly, decoder-only models like PaLM (Anil et al., 2023) can be adopted to speech recognition (Rubenstein et al., 2023). In offline handwriting recognition the input is a scanned image or a photo of handwriting and the output is digital text. Modern approaches to offline handwriting recognition are mainly based on deep neural nets according to a survey on handwritten OCR (Al Sayed et al., 2022).

The idea to **combine online and offline recognition** has been explored before in (Wang et al., 2019), which proposed

a special variation of encoder-decoder architecture to combine sequence data and image representation. A similar idea for sketch generation was investigated in (Pourreza et al., 2023) where the image representation was merged with an LLM using a cross-attention mechanism.

In recent years Large Language Models have demonstrated impressive capabilities in various tasks like question answering, math problem solving, summarization etc. (Anil et al., 2023; Brown et al., 2020; Xue et al., 2021). Recognizing the limitations of text-only models, researchers are now exploring how integrating different modalities like vision can unlock even greater capabilities for large language models. The main goal of this expansion is to have a universal model (Bommasani et al., 2022) that can be easily adapted to a variety of tasks like image captioning (Zhou et al., 2019), speech translation (Huang et al., 2023) and many more. The models that combine visual and language components (Large Visual Language Models) include Flamingo (Alayrac et al., 2022), PaLI (Chen et al., 2023c;a), PaLM-E (Driess et al., 2023), LLaVA (Liu et al., 2023). Those models vary in size and training datasets, but most notably in architectural approaches to combining image and text modalities. In Flamingo gated cross-attention was used to connect language and image modalities. PaLI expands the encoder state with image embeddings and utilizes crossattention in an encoder-decoder to process images and text together. PaLM-E and LLaVA models rely on self-attention as they fuse projected image embeddings into the sequence of text tokens.

Our work We argue that VLMs provide a natural framework to combine online and offline representations of digital inks. Unlike (Wang et al., 2019) we utilize already existing VLMs for easier processing of both ink sequences and images. Our method is focused on finding text and image representations that work best with VLMs.

6. Conclusion

In this work, we provide a representation of handwriting for VLMs as a text sequence and an image that enables tuning VLMs to a quality comparable to state of the art on three public datasets. Our representation includes relative discretized coordinates as text and rendered ink with time and distance information as image.

We show that (1) VLMs profit from multimodal inputs and that the image is particularly important in cases where the ink's text representation doesn't fit into context length (2) multiple handwriting tasks can be combined with this representation and (3) it is compatible with parameter-efficient tuning as well as fine-tuning. This points us to a future direction of exploring different handwriting task combinations in large VLMs.

7. Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

8. Acknowledgements

We would like to thank Philippe Gervais, Sean Kirmani, Henry Rowley, Blagoj Mitrevski, Arina Rak, Julian Schnitzler, Chengkun Li, Vincent Etter for their helpful suggestions.

References

- Kaggle Quick, Draw! competition. https://www.kaggle.com/competitions/quickdraw-doodle-recognition, [Online; accessed 10-Jan-2024].
- Aksan, E., Pece, F., and Hilliges, O. Deepwriting: Making digital ink editable via deep generative modeling. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pp. 1–14, 2018.
- Al Sayed, I., Mawlod, A., Alnajjar, A., and Gheni, H. Survey on handwritten recognition. pp. 273–281, 10 2022. doi: 10.1109/ISMSIT56059.2022.9932793.
- Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning, 2022.
- Alwajih, F., Badr, E., and Abdou, S. Transformer-based models for arabic online handwriting recognition. *International Journal of Advanced Computer Science and Applications*, 13(5), 2022.
- Anil, C., Wu, Y., Andreassen, A., Lewkowycz, A., Misra, V., Ramasesh, V., Slone, A., Gur-Ari, G., Dyer, E., and Neyshabur, B. Exploring length generalization in large language models, 2022.
- Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J. H., Shafey, L. E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G. H., Ahn, J., Austin, J., Barham, P., Botha, J., Bradbury, J., Brahma, S., Brooks, K., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C. A., Chowdhery, A., Crepy, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., Díaz, M., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., Garcia, X., Gehrmann, S., Gonzalez, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M., Li, W., Li, Y., Li, J., Lim, H., Lin, H., Liu, Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, A., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A. C., Roy, A., Saeta, B., Samuel, R., Shelby, R., Slone, A., Smilkov, D., So, D. R., Sohn, D., Tokumine, S., Valter,

- D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L., Yin, P., Yu, J., Zhang, Q., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S., and Wu, Y. Palm 2 technical report, 2023.
- Bavishi, R., Elsen, E., Hawthorne, C., Nye, M., Odena, A., Somani, A., and Taşırlar, S. Introducing our multimodal models, 2023. URL https://www.adept.ai/blog/fuyu-8b.
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. On the opportunities and risks of foundation models, 2022.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan,
 J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G.,
 Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G.,
 Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu,
 J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M.,
 Gray, S., Chess, B., Clark, J., Berner, C., McCandlish,
 S., Radford, A., Sutskever, I., and Amodei, D. Language
 models are few-shot learners, 2020.
- Carbune, V., Gonnet, P., Deselaers, T., Rowley, H., Daryin, A., Calvo, M., Wang, L.-L., Keysers, D., Feuz, S., and Gervais, P. Fast multi-language lstm-based online handwriting recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 2020.
- Chen, X., Djolonga, J., Padlewski, P., Mustafa, B., Changpinyo, S., Wu, J., Ruiz, C. R., Goodman, S., Wang, X., Tay, Y., Shakeri, S., Dehghani, M., Salz, D., Lucic, M.,

- Tschannen, M., Nagrani, A., Hu, H., Joshi, M., Pang, B., Montgomery, C., Pietrzyk, P., Ritter, M., Piergiovanni, A., Minderer, M., Pavetic, F., Waters, A., Li, G., Alabdulmohsin, I., Beyer, L., Amelot, J., Lee, K., Steiner, A. P., Li, Y., Keysers, D., Arnab, A., Xu, Y., Rong, K., Kolesnikov, A., Seyedhosseini, M., Angelova, A., Zhai, X., Houlsby, N., and Soricut, R. Pali-x: On scaling up a multilingual vision and language model, 2023a.
- Chen, X., Wang, X., Beyer, L., Kolesnikov, A., Wu, J., Voigtlaender, P., Mustafa, B., Goodman, S., Alabdulmohsin, I., Padlewski, P., et al. Pali-3 vision language models: Smaller, faster, stronger. arXiv preprint arXiv:2310.09199, 2023b.
- Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., Kolesnikov, A., Puigcerver, J., Ding, N., Rong, K., Akbari, H., Mishra, G., Xue, L., Thapliyal, A., Bradbury, J., Kuo, W., Seyedhosseini, M., Jia, C., Ayan, B. K., Riquelme, C., Steiner, A., Angelova, A., Zhai, X., Houlsby, N., and Soricut, R. Pali: A jointly-scaled multilingual language-image model, 2023c.
- Chiu, C.-C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonina, K., Jaitly, N., Li, B., Chorowski, J., and Bacchiani, M. State-of-the-art speech recognition with sequence-to-sequence models. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4774–4778, 2017. URL https://api.semanticscholar.org/CorpusID:206742954.
- Dehghani, M., Mustafa, B., Djolonga, J., Heek, J., Minderer, M., Caron, M., Steiner, A., Puigcerver, J., Geirhos, R., Alabdulmohsin, I., et al. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. *arXiv* preprint arXiv:2307.06304, 2023.
- Dhiaf, M., Rouhou, A. C., Kessentini, Y., and Salem, S. B. Msdoctr-lite: A lite transformer for full page multi-script handwriting recognition. *Pattern Recogn. Lett.*, 169(C):28–34, may 2023. ISSN 0167-8655. doi: 10.1016/j.patrec.2023.03.020. URL https://doi.org/10.1016/j.patrec.2023.03.020.
- Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery,
 A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu,
 T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth,
 D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint,
 M., Greff, K., Zeng, A., Mordatch, I., and Florence, P.
 Palm-e: An embodied multimodal language model, 2023.
- Gervais, P., Fadeeva, A., and Maksai, A. Mathwriting dataset, 2024. URL https://storage.

- googleapis.com/mathwriting_data/
 mathwriting-2024.tqz.
- Google Cloud. Detect handwriting in image, 2023. URL https://cloud.google.com/vision/docs/handwriting.
- Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 369–376, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143891. URL https://doi.org/10.1145/1143844.1143891.
- Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., and Schmidhuber, J. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:855–868, 2009. URL https://api.semanticscholar.org/CorpusID:14635907.
- Hannun, A. Y., Case, C., Casper, J., Catanzaro, B., Diamos, G. F., Elsen, E., Prenger, R. J., Satheesh, S., Sengupta, S., Coates, A., and Ng, A. Deep speech: Scaling up end-to-end speech recognition. *ArXiv*, abs/1412.5567, 2014. URL https://api.semanticscholar.org/CorpusID:16979536.
- Hochreiter, S. and Schmidhuber, J. Long short-term memory. 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10. 1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.
- Hu, J., Brown, M., and Turin, W. Hmm based online hand-writing recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, 1996. doi: 10.1109/34.541414.
- Huang, Z., Ye, R., Ko, T., Dong, Q., Cheng, S., Wang, M., and Li, H. Speech translation with large language models: An industrial practice, 2023.
- Jaeger, S., Liu, C.-L., and Nakagawa, M. The state of the art in japanese online handwriting recognition compared to techniques in western handwriting recognition. *International Journal on Document Analysis and Recognition*, 6:75–88, 10 2003. doi: 10.1007/s10032-003-0107-y.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.

- cc/paper_files/paper/2012/file/
 c399862d3b9d6b76c8436e924a68c45b-Paper.
 pdf.
- Kudo, T. and Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://api.semanticscholar. org/CorpusID:52051958.
- Le, A. D., Nguyen, H. T., and Nakagawa, M. End to end recognition system for recognizing offline unconstrained vietnamese handwriting. *arXiv preprint arXiv:1905.05381*, 2019.
- Lee, K., Joshi, M., Turc, I. R., Hu, H., Liu, F., Eisenschlos, J. M., Khandelwal, U., Shaw, P., Chang, M.-W., and Toutanova, K. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pp. 18893–18912. PMLR, 2023.
- Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning, 2023.
- Liu, T. and Low, B. K. H. Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks, 2023.
- Michael, J., Labahn, R., Grüning, T., and Zöllner, J. Evaluating sequence-to-sequence models for handwritten text recognition, 2019.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), Advances in Neural Information Processing Systems, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.
- Nguyen, H. T., Nguyen, C. T., and Nakagawa, M. Icfhr 2018–competition on vietnamese online handwritten text recognition using hands-vnondb (vohtr2018). In 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 494–499. IEEE, 2018.
- Pourreza, R., Bhattacharyya, A., Panchal, S., Lee, M., Madan, P., and Memisevic, R. Painter: Teaching autoregressive language models to draw sketches, 2023.
- Prabhavalkar, R., Rao, K., Sainath, T. N., Li, B., Johnson, L. M., and Jaitly, N. A comparison of sequence-to-sequence models for speech recognition. In *Interspeech*, 2017. URL https://api.semanticscholar.org/CorpusID:6028290.

- Pundak, G., Sainath, T. N., Prabhavalkar, R., Kannan, A., and Zhao, D. Deep context: End-to-end contextual speech recognition. 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 418–425, 2018. URL https://api.semanticscholar.org/CorpusID:51942169.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435.
- Ribeiro, L. S. F., Bui, T., Collomosse, J., and Ponti, M. Sketchformer: Transformer-based representation for sketched structure, 2020.
- Riche, Y., Henry Riche, N., Hinckley, K., Panabaker, S., Fuelling, S., and Williams, S. As we may ink?: Learning from everyday analog pen use to improve digital ink experiences. pp. 3241–3253, 05 2017. doi: 10.1145/3025453.3025716.
- Rubenstein, P. K., Asawaroengchai, C., Nguyen, D. D., Bapna, A., Borsos, Z., de Chaumont Quitry, F., Chen, P., Badawy, D. E., Han, W., Kharitonov, E., Muckenhirn, H., Padfield, D., Qin, J., Rozenberg, D., Sainath, T., Schalkwyk, J., Sharifi, M., Ramanovich, M. T., Tagliasacchi, M., Tudor, A., Velimirović, M., Vincent, D., Yu, J., Wang, Y., Zayats, V., Zeghidour, N., Zhang, Y., Zhang, Z., Zilka, L., and Frank, C. Audiopalm: A large language model that can speak and listen, 2023.
- Sun, C., Shrivastava, A., Singh, S., and Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.
- Wang, J., Du, J., Zhang, J., and Wang, Z.-R. Multi-modal attention network for handwritten mathematical expression recognition. In 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1181–1186, 2019. doi: 10.1109/ICDAR.2019.00191.
- Wang, T., Roberts, A., Hesslow, D., Scao, T., Chung, H., Beltagy, I., Launay, J., and Raffel, C. What language

- model architecture and pretraining objective work best for zero-shot generalization?, 04 2022.
- Xie, Y., Mouchère, H., Simistira Liwicki, F., Rakesh, S., Saini, R., Nakagawa, M., Nguyen, C. T., and Truong, T.-N. Icdar 2023 crohme: Competition on recognition of handwritten mathematical expressions. In Fink, G. A., Jain, R., Kise, K., and Zanibbi, R. (eds.), *Document Analysis and Recognition ICDAR 2023*, pp. 553–565, Cham, 2023. Springer Nature Switzerland.
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. mt5: A massively multilingual pre-trained text-to-text transformer, 2021.
- Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. Coca: Contrastive captioners are imagetext foundation models, 2022.
- Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J. J., and Gao, J. Unified vision-language pre-training for image captioning and vqa, 2019.

A. Time sampling

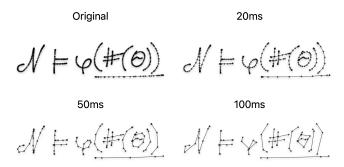


Figure 6. Different time sampling deltas. In our experiments we use 20ms for all datasets. Importantly, note that larger deltas result in a shorter representation.

In Fig. 6 we show how the time sampling delta affects the ink. With bigger deltas we get less points in the ink and important details of writing can be missed.

B. Training on the mix of datasets

Table 10. A comparison of fine-tuning separately on each dataset or on the mix of 80% MathWriting, 10% DeepWriting and 10% VNOnDB.

		CER ↓		
model	mix data	MathWriting	VNOnDB	DeepWriting
CTC transformer		4.28 (0.06)	3.82 (0.24)	5.71 (0.2)
PaLI	no	4.36 (0.04)	4.55 (0.09)	4.68 (0.16)
PaLM-E		4.22 (0.16)	3.42 (0.05)	7.98 (0.17)
CTC transformer		5.31 (0.03)	4.33 (0.04)	6.12 (0.24)
PaLI	yes	4.47 (0.07)	3.04 (0.01)	4.39 (0.06)
PaLM-E		4.19 (0.04)	3.27 (0.1)	6.89 (0.06)

In Table 10 we show that both PaLI and PaLM-E models benefit from training on a mix of datasets whereas CTC transformer doesn't. VNOnDB and DeepWriting character error rate decreases significantly if a mix of datasets is used with VLMs. We observe the decrease of PaLI MathWriting quality when trained on the mix of datasets which we attribute to fewer math tokens observed during training. We also show that training CTC transformer model on a mix of datasets doesn't lead to improvements on any of the datasets. Due to this fact we report CTC transformer that was trained on separate datasets in the Section 4.4.

C. Time and distance rendering in images

In Fig. 7 we provide an example of ambiguous writing where order of writing helps a model to recognize the ink correctly.

D. Baseline architecture

Table 11. CTC transformer architecture details.

	MathWriting	VNOnDB	DeepWriting
parameters	35M	15M	9M
layers	11	7	10
embedding size	512	384	192
attention heads	8	8	2
units per head	256	256	1024
activation	swish	gelu	gelu
dropout	0.15	0.1	0.2

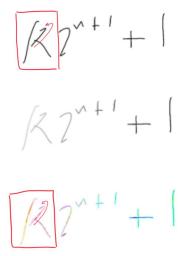


Figure 7. Example with an ambiguous first letter. Depending on the order of points it might be "K" or "R". With time and distance information in the image our model changes the wrong prediction "R" to correct "K".

Table 12. Coordinate vs histogram tokenizer.

	coordinate	histogram
resize	yes	no
vocabulary	900	12k
tokens per point	2	1

Our baseline architecture is a Transformer encoder combined with a CTC loss similar to (Alwajih et al., 2022). We provide the details of models for each dataset in Table. 11. The raw points from the ink are encoded into Bézier curves to account for factors such as input device sample rate and length of the resulting input features (Carbune et al., 2020), therefore removing the need for further preprocessing of the ink. The input features are then processed by a transformer encoder with multiple attention layers and a final logit layer of the size of the vocabulary. We train the resulting model end-to-end using a CTC loss. We don't combine a trained recognizer model with an additional external language model (ex. in (Carbune et al., 2020)) on top of transformer due to lack of public training data for this process.

E. Histogram tokenizer

We compare our method shown in Fig. 2 to a method where one point translates into one token in the sequence see Table 12. That is especially important to consider because bigger sequence length requires more resources and can lead to lower performance in LLMs (Anil et al., 2022). We can potentially enumerate all bins in coordinate tokenizer which would result in N^2 vocabulary size. However, for size N big enough to have good reconstruction (224 in our experiments) we will end up with at least 50k tokens. For this reason, we train a histogram tokenization method based on the polar form of offsets – log distance and angle. In this approach we don't use scale normalization to a fixed-size canvas as it is not required by the method. During training we split angles uniformly across 2π into 100 buckets and than we iteratively do a binary split of log distance intervals until they contain less than 0.1% of train data. This way we end up with much smaller buckets near 0 as they are more frequent.

F. MathWriting vocabulary

We use a special vocabulary for MathWriting dataset to account for the fact that some math symbols are represented in text with multiple characters, for instance θ would appear in target as "theta". This vocabulary consists of 142 special latex symbols and 87 English characters with punctuation.

"!", "&", "(", ")", "*", "+", ",", "-", ".", "l", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", ":", ";", ";", "=", "¿", "?", "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "[", "\#", "\%", "\&", "\Delta", "\Gamma", "\Lambda", "\Leftrightarrow", "\Omega", "\Phi", "\Pi", "\Psi", "\Rightarrow", "\Sigma", "\Theta", "\Upsilon", "\Vdash", "\Xi", "\", "\-", "\aleph", "\alephalpha", "\angle", "\approx", "\bigcirc", "\bigcirc", "\bigcirc", "\bigcirc", "\bigcipcirc", "\bigcipcirc "\bigwedge", "\bullet", "\cap", "\cdot", "\chi", "\circ", "\cong", "\cup", "\dagger", "\delta", "\div", "\dot", "\emptyset", "\endmatrix", "\epsilon", "\equiv", "\eta", "\exists", "\forall", "\frac", "\gamma", "\ge", "\gg", "\hat", "\hbar", "\hookrightarrow", "\iff", "\in", "\in", "\int", "\iota", "\kappa", "\lambda", "\langle", "\leel", "\lee", "\l "\leftarrow", "\leftrightarrow", "\lfloor", "\ll", "\longrightarrow", "\mapsto", "\mapsto", "\models", "\mp", "\mu", "\nabla", "\ne", "\neg", "\ni", "\noti", "\notin", "\odot", "\oint", "\omega", "\ominus", "\oplus", "\otimes", "\overline", "\partial", "\perp", "\phi", "\pi", "\prime", "\prod", "\proto", "\psi", "\rangle", "\recil", "\rfloor", "\rho", "\rightarrow", "\rightarrow", "\sigma", "\sim", "\simeq", "\sqrt", "\sqsubseteq", "\subseteq", "\subsetneq", "\supset", "\supseteq", "\tau", "\theta", "\tilde", "\times", "\top", "\triangle'", "\triangleleft", "\triangleq", "\underline", "\varphi", "\varphi", "\varpi", "\varsigma", "\vartheta", "\vdots", "\vec", "\vec", "\wedge", "\xi", "\zeta", "\{", "\—", "\}", "]", "^", "-", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z", "{", "|", "}"