Learning Beyond Labels: Self-Supervised Handwritten Text Recognition

Shree Mitra
IIIT Hyderabad
shree.mitra@research.iiit.ac.in

Ajoy Mondal
IIIT Hyderabad
ajoy.mondal@iiit.ac.in

C.V. Jawahar
IIIT Hyderabad
jawahar@iiit.ac.in

Abstract

This paper addresses a key challenge in Handwritten Text Recognition (HTR): the dependence on large volumes of labeled data. To overcome this, we propose a self-supervised learning (SSL) framework, ***LoGo-HTR****, that minimizes labeling requirements while achieving strong recognition performance. We introduce a large-scale dataset,* ***SSL-HWD*** *of 10 million word-level handwritten images from diverse scanned documents, partitioned into a small labeled subset and a much larger unlabeled subset.*

LoGo-HTR *combines a local contrastive loss for spatial consistency and a global decorrelation loss to enhance feature diversity. This dual objective enables robust, invariant, and spatially discriminative feature learning. After self-supervised pretraining, we fine-tune a transformer-based decoder using limited labeled data. Extensive experiments on standard HTR benchmarks, which include multilingual and historical data, demonstrate that, after SSL pretraining on our unlabeled dataset, our method consistently outperforms state-of-the-art approaches, even when fine-tuned using only 80% and 20% of the available labeled training data from the respective benchmarks. Ablation studies highlight the effectiveness of our dual loss design and demonstrate the potential of scalable, label-efficient handwritten text recognition. The SSL-HWD dataset and LoGo-HTR model with code are publicly available at* https://logo-ssl.github.io/.

1. Introduction

Handwritten Text Recognition (HTR) is an important task in document image analysis, essential for applications such as preserving historical manuscripts, automating administrative workflows, and enabling natural user interfaces. Despite advances in deep learning, state-of-the-art HTR models [5, 6, 17, 29, 33, 38, 39, 51] still rely on large-scale, manually annotated datasets for training. Creating these datasets is both time-consuming and resource-intensive, creating a significant bottleneck. As a result, current HTR systems struggle to scale and adapt to new domains, scripts, or variations in handwriting, limiting their broader applicability.

Recent advancements in deep learning have greatly improved the accuracy of HTR systems. This progress is particularly evident with the use of convolutional and recurrent neural networks, along with techniques like Connectionist Temporal Classification (CTC), such as [1, 15, 40, 47]. Additionally, attention-based encoder-decoder architectures [8, 16, 33, 34], have also contributed to this enhancement. These models can learn complex visual and sequential patterns found in handwritten scripts. However, their effectiveness often relies heavily on the availability of supervised data, which poses a challenge in low-resource or domain-shifted environments. To mitigate the dependency on extensive labeled datasets, synthetic handwritten data [33] has emerged as a promising alternative. By programmatically generating diverse handwriting samples using font-based rendering [31], style transfer techniques [29], or generative models, researchers can simulate realistic handwriting variations. These synthetic datasets serve as valuable pretraining resources, enabling models to learn generalizable features that transfer well to real handwritten data with minimal supervision. In contrast, diverse unlabeled handwritten notes present a valuable resource. Can we leverage these resources to develop a more generalized HTR model?

This work aims to reduce the heavy dependence on labeled data in HTR by proposing a self-supervised learning framework, **LoGo-HTR**, Local patch-based contrastive and Global decorrelation-based self-supervised learning for HTR. LoGo-HTR is designed to learn rich and meaningful representations from large-scale unlabeled handwritten text. To support our approach, we introduce a new dataset called **SSL-HWD** (Self-Supervised Learning–HandWritten Dataset), comprising over 10 million word-level handwritten samples. These samples are extracted from scanned documents across diverse domains such as Physics, Computer Science, Biology, and Mathematics, etc. contributed by 852 writers. The dataset is divided into two parts, (i) a small labeled subset used for supervised fine-tuning; and (ii) a large unlabeled subset used for self-supervised pretraining.

Figure 1. Diverse and complex handwritten text samples, from our dataset **SSL-HWD**, illustrating the challenges in real-world handwritten text recognition. The samples are categorized into five types: (a) Texts with different font colors, presenting variability in ink and pen usage; (b) Texts with a difficult background, where lines or noise interfere with text legibility; (c) Texts with distorted characters, including irregular strokes and structural inconsistencies; (d) Texts with blurring effects, where motion or focus issues hinder clarity; and (e) Texts with highlighted background, where background color or markings obscure the textual content. These examples underscore the need for robust models that can handle a wide range of visual degradations and inconsistencies in handwritten documents.

Our method follows a two-stage learning pipeline. Stage 1: Self-supervised pretraining. We use a DenseNet-based [28] visual-textual encoder, trained on the unlabeled subset with a contrastive learning objective. This helps the model learn meaningful handwriting features without relying on transcriptions. Stage 2: Supervised fine-tuning, using a standard sequence prediction loss, the pretrained model is fine-tuned on the small labeled subset. This hybrid training strategy significantly reduces the need for labeled data and improves the model's generalization ability across different handwriting styles and datasets. We validate the effectiveness of our approach through experiments on two widely used benchmarks for handwritten text recognition. Results show that pretraining on unlabeled data yields notable accuracy improvements, especially when labeled data is limited. Additional ablation studies further confirm the individual contributions of each component in our proposed framework.

In summary, our work offers the following key contributions:

- We introduce a large-scale handwritten text dataset, **SSL-HWD**, with 10M word-level images of 852 writers from diverse domains with 2.08M labeled and 7.92M unlabeled samples, providing a valuable resource for scalable self-supervised learning in HTR.
- We propose a novel method, **LoGo-HTR**, with two-stage training strategies for HTR. First, we employ contrastive self-supervised pretraining to learn robust representations, followed by supervised fine-tuning for task-specific adaptation.
- We achieve state-of-the-art results on four HTR benchmarks, **IAM**, **GNHK**, **RIMES**, and **LAM** under limited supervision, highlighting the effectiveness of the proposed approach and underscoring the promise of self-supervised learning in advancing handwritten text recognition beyond traditional label-dependent methods.

2. Related Works

2.1. Supervised Text Recognition Methods

Handwritten Text Recognition (HTR) has progressed through several stages. Early methods relied on handcrafted features with HMMs and SVMs [19, 46], effective for small vocabularies but brittle under style variation and noise. With deep learning, CNNs [20] combined with recurrent models such as LSTMs [27] and GRUs [10] became standard, while the CTC loss [21] enabled end-to-end training without alignment.

Attention-based encoder–decoders [3, 49] reframed HTR as sequence-to-sequence prediction, selectively focusing on relevant regions and improving robustness to irregular text layouts. Recently, transformer-based architectures [33, 34, 49] have replaced recurrent models, leveraging self-attention for long-range dependencies and achieving state-of-the-art accuracy with scalable, parallelizable training.

Scene text recognition (STR) has followed a similar trajectory: ABINet [18] integrates iterative vision–language interaction, VisionLAN [52] aligns semantics with visual features, and PARSeq [4] introduces parallel autoregressive decoding. Overall, supervised recognition has shifted from hand-crafted features to CNN–RNN–CTC pipelines, and now to fully transformer-based systems that dominate both HTR and STR benchmarks.

2.2. Semi-supervised Text Recognition Methods

Semi-supervised learning (SSL) mitigates the high cost of character-level annotation by combining limited labeled

data with abundant unlabeled samples. Techniques include pseudo-labeling, consistency regularization, and uncertainty modeling.

For STR, Qu *et al.* [45] proposed multi-view aggregation to stabilize pseudo-labels, while for HTR, SoftCTC [30] extended CTC with soft pseudo-labels that tolerate multiple plausible transcriptions. Seq-UPS [41] refined pseudo-labeling with sequence- and character-level uncertainty estimation, ensuring reliable unlabeled contributions. Beyond recognition, SemiETS [36] introduced a semi-supervised text spotting framework with hierarchical pseudo-labels and spatial–semantic consistency.

Together, recent SSL methods show that refining pseudo-labels, enforcing cross-view consistency, and gradually incorporating unlabeled samples are effective for both HTR and STR, offering strong recognition performance under low-resource conditions.

2.3. Self-supervised Learning for Text Recognition

Self-supervised learning (SSL) learns robust representations without transcriptions, making it attractive for HTR and STR. SeqCLR [2] aligns token features across augmented views; DiG [53] combines contrastive and masked image modeling; and CCD/CCDPlus [24, 25] enforce local character-level consistency, achieving state-of-the-art results. In STR, TextScanner [50] integrates character-order signals, while surveys [43] categorize SSL methods into global contrastive, generative, and local distillation approaches.

Despite these advances, HTR remains challenged by style variability, noisy images, and the scarcity of labeled corpora. Recent semi- and self-supervised approaches [45, 50] help alleviate these issues, pushing the field toward more label-efficient and generalizable recognition systems. Our **LoGo-HTR** extends this line by unifying local patch-wise contrastive alignment with global redundancy reduction [55], producing spatially discriminative and decorrelated embeddings that transfer well across handwriting domains.

3. SSL-HWD

3.1. Data Collection

The dataset, which is curated from publicly available web sources, comprises 81,280 pages of publicly available digitized manuscripts, selected for being fully or substantially handwritten. It spans a wide range of authors, styles, complexities, and over 20 domains, including literature, sciences, and mathematics. To capture natural handwriting variations, it includes diverse document types like personal diaries, academic notes, and historical correspondence. Each document was carefully reviewed for legible, meaningful text to ensure high quality and utility for robust handwritten text recognition[1].

3.2. Annotation

We follow a two-step process for annotation — (i) automatic annotation and (ii) manual verification.

Automatic Annotation: We employed Amazon Textract's[2] `AnalyzeDocument API` to automatically extract line- and word-level bounding boxes with textual transcriptions from 81,280 page images. By enabling the `"TEXT_DETECTION"` feature, the API returned structured JSON data containing both **Line** and **Word** blocks with their respective text, coordinates, and confidence scores. This process yielded over 2.9 million line-level samples and 10 million word-level instances. For quality control, all word images with a high confidence score ($\geq$0.99) were cropped for subsequent manual verification.

Manual Verification and Correction: Our language-expert annotation team manually checks each word image and its textual transcription. The verified 2.08M word-level images and textual transcriptions are used to create a labeled set. While all other 7.92M word-level images without textual transcriptions are used for creating the unlabeled set.

Dataset	#Pages	#Writers	#Words	#Unique
IAM [37, 60]	1.5K	657	115K	10.5K
GNHK [32]	687	–	39K	12.3K
IIIT-HW-English-Word [39]	20.8K	1,215	757K	174K
SSL-HWD (ours)	81.2K	852	10M	107K

Table 1. Comparison of standard handwritten text datasets with our **SSL-HWD** dataset.

3.3. Inherent Characteristic

The **SSL-HWD** dataset provides large-scale, diverse handwriting samples across multiple challenges (Fig. 1), including multicolored text, blur, background interference, distortions, and highlights, reflecting real-world degradations. Compared to existing datasets (Table 1), it is the largest, with over 10M word-level instances from 852 writers. Its vocabulary is notably richer (Table 2), covering alphabetic, numeric, and rare words for stronger generalization. **SSL-HWD** includes both labeled data for recognition and writer identification, and unlabeled data for self-supervised, semi-supervised, and domain adaptation tasks.

4. LoGo-HTR

We propose **LoGo-HTR**, a self-supervised framework learning transferable features from unlabeled handwritten

[1] *Refer to Appendix A of the supplementary material for more details about data collection.*

[2] https://aws.amazon.com/textract/

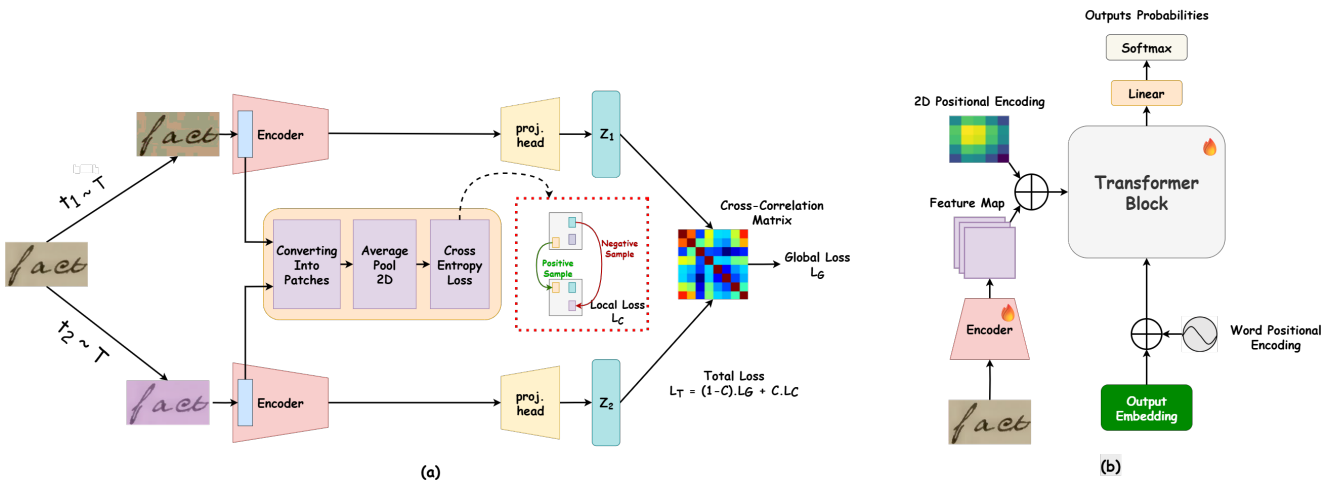

Figure 2. Overview of our proposed two-stage **LoGo-HTR** framework for handwritten text recognition. (a) During the self-supervised pretraining stage, two augmented views of the same input image are passed through a shared encoder. The representations are processed with a local patch-wise contrastive loss and a global decorrelation loss to learn spatially discriminative and semantically rich features. (b) In the downstream stage, the pretrained encoder is coupled with a transformer-based decoder to generate a word-level sequence using limited labeled data. This modular pipeline enables robust recognition with reduced annotation effort.

Category	IAM	GNHK	SSL-HWD (Ours)
Alphabetic Words	9,103	6,649	61,088
Numeric Words	116	250	4,981
Stop-words	140	141	457
Other Words	1,121	4,194	41,287
Total Unique Words	10,480	12,341	107,813

Table 2. Comparison of unique words across datasets.

data by combining local patch-based contrastive learning with a global decorrelation objective [55]. During pretraining, a contrastive loss aligns matching patches from two augmented views of an image, building robustness to handwriting variations. Unlike prior methods [26], our approach avoids memory queues by operating within each image. Concurrently, the decorrelation loss enhances feature diversity. The pretrained encoder is then attached to a Transformer decoder and fine-tuned with limited labeled data for recognition (Fig. 2), enabling a modular, label-efficient design.

4.1. Self-Supervised Pretraining

4.1.1. Local Contrastive Loss for Patch-wise Feature Alignment

To enhance local correspondence, we introduce a patch-level contrastive loss, $\mathcal{L}_{\text{Local}}$. Complementing global objectives [55], which can overlook local details, our loss aligns fine-grained spatial structures by enforcing similarity between corresponding patches across augmented views. This helps capture subtle local patterns crucial for handwritten text.

Let $x_{\text{aug}_1}, x_{\text{aug}_2} \in \mathbb{R}^{B\times C\times H\times W}$ be the feature maps from two augmentations of an image, with batch size B, channels C, and spatial resolution $H \times W$.

Patch Extraction: We divide each feature map into patches to compare corresponding spatial regions of two augmentations of the same image. The patch size is determined by:

$$P = \left\lfloor \frac{H}{K} \right\rfloor, \quad \text{and stride} \quad S = \lfloor (1 - \text{overlap}) \cdot P \rfloor,$$

where K defines the number of patches per spatial dimension, and the `overlap` parameter controls the amount of overlap between adjacent patches (ranging from 0 to less than 1). This flexible patching strategy allows the framework to adjust for varying levels of spatial granularity. Each patch in a batch is then converted to a fixed-size feature vector using average pooling:

$$z^1_{b,n} = \text{AvgPool}(x^{(b)}_{\text{aug}_1}[i : i + P, j : j + P]) \in \mathbb{R}^C,$$
$$z^2_{b,n} = \text{AvgPool}(x^{(b)}_{\text{aug}_2}[i : i + P, j : j + P]).$$

Here, $n \in \{1, \ldots, N\}$ indexes the extracted patches for each image in batch $b \in \{1, \ldots, B\}$, and (i, j) denotes the spatial coordinates corresponding to patch n.

Positive and Negative Pairs: To build contrastive supervision, we define **positive pairs** as patches extracted from the same spatial region across two augmentations of the same image. Their similarity is computed using normalized cosine similarity:

$$s_{\text{pos}}^{(b,n)} = \frac{z_{b,n}^1 \cdot z_{b,n}^2}{\|z_{b,n}^1\| \, \|z_{b,n}^2\|}.$$

For **negative pairs**, we do not rely on other images in the batch but instead exploit the intra-view diversity within the same image. Specifically, we compute patch-patch similarities within each augmented view:

$$S_b^1 = Z_b^1 (Z_b^1)^\top, \quad S_b^2 = Z_b^2 (Z_b^2)^\top,$$

where $Z_b^1, Z_b^2 \in \mathbb{R}^{N \times C}$ are the stacked patch representations for all N patches in a given image. To focus only on dissimilar regions, we subtract the diagonal terms which correspond to self-similarities:

$$\tilde{S}_b^1 = S_b^1 - \text{diag}(S_b^1), \quad \tilde{S}_b^2 = S_b^2 - \text{diag}(S_b^2).$$

These intra-view negatives act as hard negatives, representing other spatial regions within the same image that should be distinguishable from the true corresponding patch.

Loss Formulation: For each patch, we build a contrastive logit vector composed of the positive similarity (anchor to corresponding patch in the other view) and all negative similarities (anchor to other patches in both views). These logits are scaled by a temperature parameter τ, which controls the sharpness of the softmax distribution:

$$\ell_{b,n} = \left[\frac{s_{\text{pos}}^{(b,n)}}{\tau} \;\middle|\; \frac{\tilde{S}_{b,n,:}^1}{\tau} \;\middle|\; \frac{\tilde{S}_{b,n,:}^2}{\tau} \right] \in \mathbb{R}^{1+2(N-1)}.$$

Here, the logit corresponding to the positive pair is always placed at index 0, and the remaining elements correspond to intra-view negatives. The final patch-level contrastive loss is the cross-entropy over these logits:

$$\mathcal{L}_{\text{Local}} = \frac{1}{B \cdot N} \sum_{b=1}^{B} \sum_{n=1}^{N} \text{CE}(\ell_{b,n}, 0),$$

where CE denotes the standard cross-entropy loss. This encourages the model to assign higher similarity to corresponding patches across augmentations and suppress similarity between unrelated regions, thereby preserving and aligning local discriminative features across transformations.

4.1.2. Global Loss for Feature Decorrelation

To ensure global semantic consistency and reduce representational redundancy, we adopt the Barlow Twins loss [55] as the global learning objective. This self-supervised method aligns embeddings from two augmented views of the same input while explicitly discouraging correlation across feature dimensions.

Let $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^{B \times d}$ denote embeddings from two distorted views of a batch of inputs, where B is the batch size and d the feature dimensionality. The cross-correlation matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$ is defined as:

$$\mathbf{C}_{ij} = \frac{\sum_{b=1}^{B} z_1^{(b,i)} z_2^{(b,j)}}{\sqrt{\sum_{b=1}^{B} (z_1^{(b,i)})^2} \cdot \sqrt{\sum_{b=1}^{B} (z_2^{(b,j)})^2}}. \tag{1}$$

The loss is then formulated as:

$$\mathcal{L}_{\text{BT}} = \sum_i (1 - \mathbf{C}_{ii})^2 + \lambda \sum_i \sum_{j \neq i} \mathbf{C}_{ij}^2, \tag{2}$$

where the diagonal term enforces invariance by aligning corresponding features ($\mathbf{C}_{ii} \to 1$), while the off-diagonal term promotes decorrelation by minimizing feature redundancy ($\mathbf{C}_{ij} \to 0, i \neq j$). The balance between invariance and decorrelation is controlled by the hyperparameter λ.

This objective leads to embeddings that are both augmentation-invariant and information-rich, facilitating generalization in downstream recognition tasks. A more detailed analysis of the complementarity between global and local self-supervised objectives, supported by theoretical insights and prior work, is provided in *Appendix B*.

4.2. Supervised Training

In the downstream stage, we couple the self-supervised pretrained CNN encoder with a Transformer decoder for handwritten text recognition. The encoder, based on DenseNet [28], captures local patterns and long-range dependencies. Pretrained with patch-wise contrastive alignment and global decorrelation, it provides robust and transferable features with minimal annotation. Given an input $\mathbf{I} \in \mathbb{R}^{3 \times H_0 \times W_0}$, the encoder outputs a feature map $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$, which is reshaped into a sequence of $L = H \times W$ tokens $\{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$, each enriched with positional encodings $\mathbf{p}_i$, yielding $\mathbf{z}_i = \mathbf{x}_i + \mathbf{p}_i$. This sequence serves as memory for a Transformer decoder [49, 59] with an Attention Refinement Module (ARM [58]) to auto regressively generate word-level sequences. Depending on configuration, the encoder can be frozen or fine-tuned, ensuring stable adaptation to new domains. This modular design enables efficient recognition with low annotation cost and lightweight decoding.

5. Experiments

5.1. Implementation Details

We use a DenseNet-based encoder with 3 blocks of 16 bottleneck layers, growth rate k=24, dropout p=0.2, and compression factor θ=0.5. For self-supervised learning, each image is split into 5×5 patches (after the first convolution layer), followed by average pooling. A projection head ($1024 \rightarrow 512$) generates embeddings for the global decorrelation loss. Augmentations include scale jittering ([0.7, 1.4]) and color jittering (brightness/contrast 0.25, saturation 0.2, hue 0.2), each with 0.5 probability. The SSL loss combines local and global terms (weights: 0.6 and 0.4). For fine-tuning, we adopt the **CoMER** [58] decoder: 3 Transformer layers (dimension 256, FFN 1024, dropout 0.3, 8 heads) with coverage attention. All experiments were conducted on **two RTX A6000 GPUs** (48 GB VRAM each). (*Refer to Appendix C of the supplementary material for more details*).

5.2. Dataset

We conduct training and evaluation using five datasets: IAM, GNHK, RIMES, LAM and SSL-HWD (Ours). Among 2.08M labeled word-level images of the SSL-HWD dataset, 60% (1.25M samples) are used for training, and the remaining 40% (0.83M samples) are held out for testing. For IAM, we follow the standard Aachen partitions[3] to split the data into training and test sets. The GNHK dataset comprises 26K samples for training and 9.9K for evaluation. The RIMES dataset is used following its standard training and test partitions. The LAM dataset, which consists of historical handwritten documents, is considered more challenging due to its degraded quality and diverse writing styles; we use its official training and test splits for our experiments.

5.3. Evaluation Metrics

We evaluate the performance of our **LoGo-HTR** using four standard metrics, Word Recognition Rate (WRR), Character Recognition Rate (CRR), Character Error Rate (CER), and Word Error Rate (WER) [5, 6, 17, 29, 33, 38, 39, 51]. For word-level datasets, WRR matches $100-\text{WER}$. In contrast, for line-level datasets (e.g., LAM), this equivalence breaks since WER is computed via edit distance, where insertions and deletions shift word alignment. Likewise, CRR is not simply $100-\text{CER}$ because character-level insertions and deletions affect the metric. (*Details are available in Appendix D of the supplementary material*).

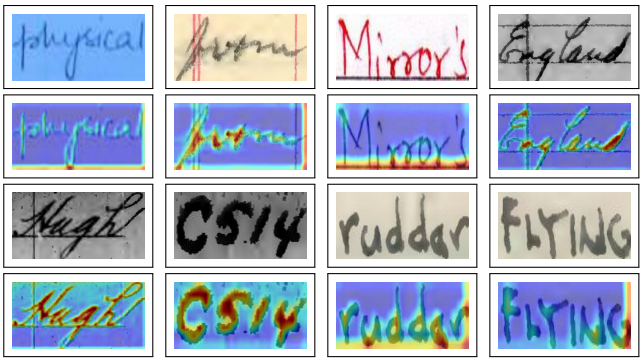

Figure 3. Grad-CAM visualizations showing model attention on cursive, blurred, long, and numeric words; rows 1 and 3 show inputs, rows 2 and 4 the heatmaps.

6. Result Analysis

6.1. Comparison with SOTA

To validate the effectiveness and generalization capacity of the proposed self-supervised framework, we conduct extensive evaluations on four benchmark datasets, IAM, GNHK, RIMES, and LAM. As shown in Table 3, we compare our method with current models including TrOCR [33] and Bhunia *et al.* [5] in IAM, Mondal *et al.* [39] in GNHK, Gu *et al.* [23] in RIMES, and HTR-VT [34] in LAM.

Unlike prior works relying on full supervision or synthetic augmentation, our approach leverages only unlabeled data for pretraining and is fine-tuned with 20–100% labeled target data. Despite using far fewer labels, our model consistently outperforms or remains highly competitive with fully supervised baselines.

On IAM, we achieve a WER of 11.93 and CER of 2.31 with 80% of the labeled data, outperforming the fully supervised TrOCR. On GNHK, our model reaches a WER of 19.69 and CER of 9.05 using 80% labeled data, surpassing the baseline despite no use of the supervised IIIT-HW-English-Word dataset. On RIMES, our method attains a WER of 7.20 and CER of 2.38 with 80% labeled data. Notably, although LAM is a line-level historical dataset and our pretraining is performed only on large-scale word-level data, finetuning on LAM still yields a WER of 6.3 and CER of 2.39 with full data, achieving state-of-the-art performance.

6.2. LoGo-HTR Performance on SSL-HWD

LoGo-HTR achieves strong recognition performance on the **SSL-HWD** test set, demonstrating the effectiveness of combining local and global self-supervised objectives. Fig. 3 shows a heatmap of the feature learned by **LoGo-HTR**. Self-supervised pretraining with local and global objectives learn more representative features for recognition.

[3] https://opensrl.org/56/

Dataset	Model	#Params	SSL Pretraining	Supervised Training/Pre-Training	Finetuning	#Training Data	WER / WRR	CER / CRR
IAM	$\text{TrOCR}_{\text{LARGE}}$ [33]	558M	–	IAM + Synthetic	–	100%	– / –	2.89 / –
	Bhunia *et al.* [5]	–	–	STR+HTR Datasets [5]	IAM	100%	– / 86.0	– / –
	Bluche *et al.* [6]	750k / 300k	–	IAM + Synthetic	–	100%	– / –	3.20 / –
	Michael *et al.* [38]	–	–	IAM	–	100%	– / –	4.87 / –
	Wang *et al.* [51]	–	–	IAM	–	100%	– / –	6.40 / –
	Kang *et al.* [29]	–	–	Synthetic [29]	IAM	100%	– / –	4.67 / –
	Diaz *et al.* [17]	5M - 30M(Best 22M)	–	IAM + Internal [17]	–	100%	– / –	2.75 / –
	HTR-VT [34]	53.5M	–	IAM	–	100%	14.9 / –	4.7 / –
	Gu *et al.* [23]	80K	–	IAM	–	100%	10.32 / –	3.36 / –
	LoGo-HTR	6.4M	Ours(Unlabeled)	–	IAM	20%	30.8 / 69.2	7.5 / 86.0
						40%	19.3 / 80.7	5.2 / 91.0
						60%	12.01 / **87.99**	3.36 / 95.42
						80%	11.93 / **88.07**	**2.31** / 96.33
						100%	**10.27** / **89.73**	**2.01** / **97.76**
GNHK	Lee *et al.* [32]	–	–	GNHK	–	100%	– / 50.2	– /86.1
	Mondal *et al.* [39]	–	–	IIIT-HW-English-Word [39]	GNHK	100%	– / 64.31	– / 83.44
	LoGo-HTR	6.4M	Ours(Unlabeled)	–	GNHK	20%	32.1 / 67.9	19.4 / 84.3
						40%	28.4 / **71.6**	16.2 / **86.4**
						60%	24.61 / **75.39**	13.33 / **88.37**
						80%	19.69 / **80.31**	9.05 / **91.43**
						100%	12.07 / **87.93**	7.20 / **92.58**
RIMES	Bhunia *et al.* [5]	–	–	STR+HTR Datasets [5]	RIMES	100%	– / 90.06	– / –
	SPAN [12]	19M	–	RIMES	–	100%	13.8 / –	3.81 / –
	Puigcerver *et al.* [44]	–	–	RIMES	–	100%	12.8 / –	3.3 / –
	Coquenet *et al.* [14]	–	–	RIMES	–	100%	8.32 / –	3.04 / –
	DAN [13]	–	–	RIMES	–	100%	6.78 / –	2.63 / –
	Gu *et al.* [23]	80K	–	RIMES	–	100%	6.63 / –	2.19 / –
	LoGo-HTR	6.4M	Ours(Unlabeled)	–	RIMES	20%	26.50 / 73.50	6.68 / 93.20
						40%	10.80 / 89.20	3.68 / 96.10
						60%	7.20 / **92.80**	2.38 / 97.45
						80%	**6.15** / **93.85**	**1.89** / 97.99
						100%	**5.50** / **94.50**	**1.78** / 98.05
LAM*	TrOCR [7, 33]**	385M	–	LAM	–	100%	11.6 / –	3.6 / –
	GFCN [7, 11]**	1.4M	–	LAM	–	100%	18.5 / –	5.2 / –
	OrigamiNet [7, 54]**	115.3M	–	LAM	–	100%	11.0 / –	3.0 / –
	HTR-VT [34]	53.5M	–	LAM	–	100%	7.4 / –	2.8 / –
	LoGo-HTR	6.4M	Ours(Unlabeled)	–	LAM	20%	24.86 / 74.8	7.6 / 92.9
						40%	14.9 / 84.6	4.38 / 95.80
						60%	9.6 / 89.2	3.43 / 96.76
						80%	**7.2** / 91.9	3.2 / 96.98
						100%	**6.3** / 93.1	**2.39** / 97.33

Table 3. Performance comparison with SOTA on four benchmarks IAM, GNHK, RIMES and LAM. (*) Line-level dataset. (**) Re-implementation by [7].

6.3. Cross Dataset Evaluation

Finetune	Test Datasets									
	IAM		GNHK		RIMES		LAM		OURS	
	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER
IAM	**10.27**	**2.01**	52.50	47.80	21.50	11.8	27.52	12.38	26.70	18.20
GNHK	34.20	22.90	12.07	7.20	36.00	23.50	25.00	12.00	19.40	12.60
RIMES	18.30	9.40	50.08	43.9	**5.50**	**1.78**	22.4	8.6	25.50	16.90
LAM	24.30	9.10	28.00	13.50	22.90	7.50	**6.9**	**2.39**	16.00	7.20
Ours	13.20	2.90	**10.10**	**6.80**	11.20	3.50	16.40	7.20	**5.20**	**1.20**

Table 4. Cross-datasets Evaluations.

We evaluate the generalization of our dataset through cross-dataset testing. The model is pretrained on large-scale unlabeled data using a self-supervised strategy and then fine-tuned on a small labeled subset. We test on IAM [37], GNHK [32], RIMES [22], and LAM [7], which differ in vocabulary, script, and writing conditions. As shown in Table 4, our dataset achieves consistent performance across domains: Ours→IAM (13.2/2.9), Ours→GNHK (10.1/6.8), Ours→RIMES (11.2/3.5), and Ours→LAM (16.4/7.2). In contrast, IAM→OURS rises sharply to 26.7/18.2, GNHK→IAM reaches 34.2/22.9, RIMES→LAM degrades to 22.4/8.6, while LAM→OURS still remains higher at 16.0/7.0. These comparisons indicate that models trained on individual datasets overfit to their domain, whereas ours transfers robustly across all. Overall, the strong Ours→Others results highlight the potential of our dataset as a reliable source for pretraining and fine-tuning in generalized handwritten text recognition.

6.4. Comparison with SSL-Based Methods

Table 5 compares LoGo-HTR with prior SSL approaches. While methods such as SimCLR(Text-DIAE), Seq-CLR, and CMT-Co achieve competitive results, their performance drops on challenging handwriting styles. LoGo-HTR attains **89.73%** WRR on IAM and **94.5%** on RIMES, outperforming the best baselines by a clear margin. This high-

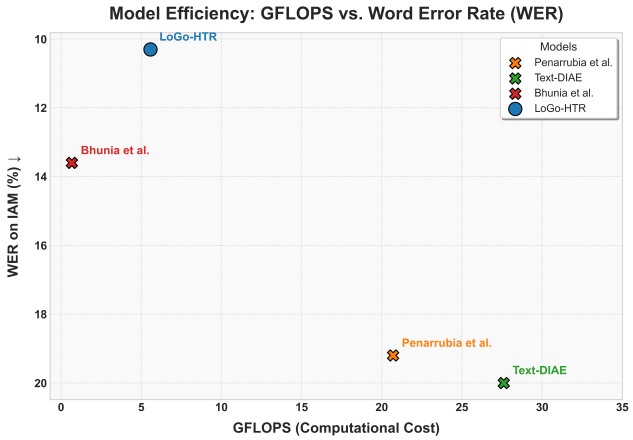

Figure 4. Model efficiency analysis. Word Error Rate (WER) on the IAM benchmark is plotted against computational cost in GFLOPS. The proposed LoGo-HTR model achieves a superior balance of low WER and computational efficiency compared to other methods, including Penarrubia et al. [42], Text-DIAE [48], and Bhunia et al. [5]. (*See Appendix G for further details.)*

lights the advantage of combining local patch-wise contrastive learning with global decorrelation for robust and transferable representations.

Methods	IAM	RIMES
SimCLR [9, 48]	70.7	–
Seq-CLR [2]	79.9	92.4
Text-DIAE [48]	80.0	–
PerSec [35, 48]	81.8	–
Chaco [56]	81.4	90.6
Penarrubia et al. [42]	80.8	92.0
CMT-Co [57]	81.9	91.2
DiG [53]	87.0	–
Ours	**89.73**	**94.5**

Table 5. Comparison of self-supervised methods on IAM and RIMES datasets (measured in Word Recognition Rate, %). Our method achieves the best performance across both benchmarks.

7. Ablation Study

We conduct an ablation study to evaluate self-supervised pretraining strategies: (i) using only the global decorrelation loss, and (ii) combining it with our proposed local contrastive loss. As shown in Table 6, the **Global+Local** setup consistently outperforms the **Global**-only baseline across all datasets. For instance, on IAM, WER drops from 18.5% to 10.3% and CER from 8.0% to 2.0%; on the more challenging LAM dataset, WER reduces from 17.5% to 6.9% and CER from 9.5% to 2.4%. Similar trends hold for RIMES, GNHK, and our in-house dataset, confirming that local supervision complements the global objective, yielding more robust representations. Figure 5 further supports this: the **Global**-only setup (red) converges slowly with high variance, while **Global+Local** (blue) converges faster and more stably, indicating that local patch-level contrastive loss regularizes training and accelerates optimization. Overall, combining global and local objectives enhances both accuracy and training efficiency. *(See Appendix F for further details.)*

SSL Method	IAM	GNHK	RIMES	LAM	OURS
	WER / CER	WER / CER	WER / CER	WER / CER	WER / CER
Global	18.5 / 8.0	22.8 / 14.2	14.0 / 6.2	17.5 / 9.5	14.8 / 9.7
Global+Local	**10.3 / 2.0**	**12.1 / 7.2**	**5.5 / 1.8**	**6.9 / 2.4**	**5.2 / 1.2**

Table 6. The effectiveness of the combined Global+Local loss, which significantly reduces Word and Character Error Rates (WER/CER) across all benchmarks.

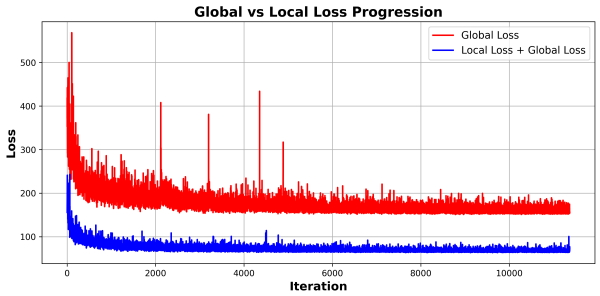

Figure 5. Self-supervised pretraining loss progression. The combined Global+Local loss (blue) converges faster and more stably than the Global-only loss (red).

8. Conclusion

In this work, we proposed **LoGo-HTR**, a self-supervised framework for handwritten text recognition that reduces dependence on large labeled datasets. It learns robust representations from unlabeled handwriting using a combination of local contrastive learning and global decorrelation. We also introduced **SSL-HWD**, a large-scale dataset containing 10 million word-level samples to support generalization across writing styles. Experimental results show that **LoGo-HTR** consistently outperforms supervised baselines in low-label regimes, while ablation and cross-data set evaluations confirm its effectiveness and generalizability. These results indicate that accurate recognition can be achieved with minimal annotation.

Acknowledgment

This work is supported by MeitY, Government of India, through the NLTM-Bhashini project.

References

[1] Fetulhak Abdurahman, Eyob Sisay, and Kinde Anlay Fante. AHWR-Net: offline handwritten amharic word recognition using convolutional recurrent neural network. *SN Applied Sciences*, 3(8):760, 2021. 1

[2] Aviad Aberdam, Ron Litman, Shahar Tsiper, Oron Anschel, Ron Slossberg, Shai Mazor, R. Manmatha, and Pietro Perona. Sequence-to-sequence contrastive learning for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15302–15312, 2021. 3, 8

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 2

[4] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *European Conference on Computer Vision*, pages 178–196, Cham, 2022. Springer Nature Switzerland. 2

[5] Ayan Kumar Bhunia, Aneeshan Sain, Pinaki Nath Chowdhury, and Yi-Zhe Song. Text is text, no matter what: Unifying text recognition using knowledge distillation. In *ICCV*, pages 983–992, 2021. 1, 6, 7, 8

[6] Théodore Bluche and Ronaldo Messina. Gated convolutional recurrent neural networks for multilingual handwriting recognition. In *ICDAR*, pages 646–651, 2017. 1, 6, 7

[7] Silvia Cascianelli, Vittorio Pippi, Maarand Martin, Marcella Cornia, Lorenzo Baraldi, Kermorvant Christopher, and Rita Cucchiara. The lam dataset: A novel benchmark for line-level handwritten text recognition. In *International Conference on Pattern Recognition*, 2022. 7

[8] Adrian Chan, Anupam Mijar, Mehreen Saeed, Chau-Wai Wong, and Akram Khater. Hatformer: Historic handwritten arabic text recognition with transformers. *arXiv preprint arXiv:2410.02179*, 2024. 1

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 8

[10] Kyunghyun Cho, B van Merrienboer, Caglar Gulcehre, F Bougares, H Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014. 2

[11] Denis Coquenet, Clément Chatelain, and Thierry Paquet. Recurrence-free unconstrained handwritten text recognition using gated fully convolutional network. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 19–24, 2020. 7

[12] Denis Coquenet, Clément Chatelain, and Thierry Paquet. SPAN: A simple predict & align network for handwritten paragraph recognition. In *Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part III*, page 70–84, Berlin, Heidelberg, 2021. Springer-Verlag. 7

[13] Denis Coquenet, Clément Chatelain, and Thierry Paquet. DAN: A segmentation-free document attention network for handwritten document recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8227–8243, 2023. 7

[14] Denis Coquenet, Clément Chatelain, and Thierry Paquet. End-to-end handwritten paragraph text recognition using a vertical attention network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):508–524, 2023. 7

[15] Arthur Flor de Sousa Neto, Byron Leite Dantas Bezerra, Alejandro Héctor Toselli, and Estanislau Baptista Lima. A robust handwritten recognition system for learning on different data restriction scenarios. *PRL*, 159:232–238, 2022. 1

[16] Marwa Dhiaf, Ahmed Cheikh Rouhou, Yousri Kessentini, and Sinda Ben Salem. MSdocTr-Lite: A lite transformer for full page multi-script handwriting recognition. *PRL*, 169: 28–34, 2023. 1

[17] Daniel Hernandez Diaz, Siyang Qin, Reeve Ingle, Yasuhisa Fujii, and Alessandro Bissacco. Rethinking text line recognition models. *arXiv preprint arXiv:2104.07787*, 2021. 1, 6, 7

[18] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7094–7103, 2021. 2

[19] Carlos Garrido-Munoz, Antonio Rios-Vila, and Jorge Calvo-Zaragoza. Handwritten text recognition: A survey. *arXiv preprint arXiv:2502.08417*, 2025. 2

[20] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *NeurIPS*, 2008. 2

[21] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006. 2

[22] Emmanuele Grosicki and Haikal El-Abed. Icdar 2011 - french handwriting recognition competition. In *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, page 1459–1463, USA, 2011. IEEE Computer Society. 7

[23] Wenhao Gu, Li Gu, Chingyee Yee Suen, and Yang Wang. Metawriter: Personalized handwritten text recognition using meta-learned prompt tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23494–23504, 2025. 6, 7

[24] Tongkun Guan, Wei Shen, Xue Yang, Qi Feng, Zekun Jiang, and Xiaokang Yang. Self-supervised character-to-character distillation for text recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19473–19484, 2023. 3

[25] Tongkun Guan, Wei Shen, and Xiaokang Yang. CCDPlus: Towards accurate character to character distillation for text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3546–3562, 2025. 3

[26] Jamshid Hassanpour, Vinkle Srivastav, Didier Mutter, and Nicolas Padoy. Overcoming dimensional collapse in self-supervised contrastive learning for medical image segmentation. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1–5, 2024. 4

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[28] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018. 2, 5

[29] Lei Kang, Pau Riba, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. Pay attention to what you read: non-recurrent handwritten text-line recognition. *PR*, 129: 108766–108776, 2022. 1, 6, 7

[30] Martin Kišš, Michal Hradiš, Karel Beneš, Petr Buchal, and Michal Kula. SoftCTC—semi-supervised learning for text recognition using soft pseudo-labels. *International Journal on Document Analysis and Recognition (IJDAR)*, 27(2):177–193, 2024. 3

[31] Praveen Krishnan and CV Jawahar. Generating synthetic data for text recognition. *arXiv preprint arXiv:1608.04224*, 2016. 1

[32] Alex WC Lee, Jonathan Chung, and Marco Lee. Gnhk: a dataset for english handwriting in the wild. In *International Conference on Document Analysis and Recognition*, pages 399–412, 2021. 3, 7

[33] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. Trocr: Transformer-based optical character recognition with pre-trained models. In *AAAI*, pages 13094–13102, 2023. 1, 2, 6, 7

[34] Yuting Li, Dexiong Chen, Tinglong Tang, and Xi Shen. Htr-vt: Handwritten text recognition with vision transformer. *PR*, 158:110967–110979, 2025. 1, 2, 6, 7

[35] Hao Liu, Bin Wang, Zhimin Bao, Mobai Xue, Sheng Kang, Deqiang Jiang, Yinsong Liu, and Bo Ren. Perceiving stroke-semantic context: Hierarchical contrastive learning for robust scene text recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2):1702–1710, 2022. 8

[36] Dongliang Luo, Hanshen Zhu, Ziyang Zhang, Dingkang Liang, Xudong Xie, Yuliang Liu, and Xiang Bai. Semiets: Integrating spatial and content consistencies for semi-supervised end-to-end text spotting. *CVPR*, 2025. 3

[37] U.-V. Marti and H. Bunke. The iam-database: an english sentence database for offline handwriting recognition. *IJDAR*, 5 (1):39–46, 2002. 3, 7

[38] Johannes Michael, Roger Labahn, Tobias Grüning, and Jochen Zöllner. Evaluating sequence-to-sequence models for handwritten text recognition. In *ICDAR*, pages 1286–1293, 2019. 1, 6, 7

[39] Ajoy Mondal, Krishna Tulsyan, and C. V. Jawahar. Bridging the gap in resource for offline english handwritten text recognition. In *ICDAR*, pages 413–428, 2024. 1, 3, 6, 7

[40] Duc Nguyen, Nhan Tran, and Hung Le. Improving long handwritten text line recognition with convolutional multi-way associative memory. *arXiv preprint arXiv:1911.01577*, 2019. 1

[41] Gaurav Patel, Jan Allebach, and Qiang Qiu. Seq-UPS: Sequential uncertainty-aware pseudo-label selection for semi-supervised text recognition. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6169–6179, 2023. 3

[42] Carlos Penarrubia, Carlos Garrido-Munoz, Jose J. Valero-Mas, and Jorge Calvo-Zaragoza. Spatial context-based self-supervised learning for handwritten text recognition. *Pattern Recognition Letters*, 196:79–85, 2025. 8

[43] Carlos Penarrubia, Jose J. Valero-Mas, and Jorge Calvo-Zaragoza. Self-supervised learning for text recognition: A critical survey. *International Journal of Computer Vision*, 133(9):6221–6250, 2025. 3

[44] Joan Puigcerver. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 67–72, 2017. 7

[45] Yadong Qu, Yuxin Wang, Bangbang Zhou, Zixiao Wang, Hongtao Xie, and Yongdong Zhang. Boosting semi-supervised scene text recognition via viewing and summarizing. In *Advances in Neural Information Processing Systems*, pages 105503–105527. Curran Associates, Inc., 2024. 3

[46] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. 2

[47] George Retsinas, Konstantina Nikolaidou, and Giorgos Sfikas. Enhancing CRNN HTR architectures with transformer blocks. In *ICDAR*, pages 425–440, 2024. 1

[48] Mohamed Ali Souibgui, Sanket Biswas, Andres Mafla, Ali Furkan Biten, Alicia Fornés, Yousri Kessentini, Josep Lladós, Lluis Gomez, and Dimosthenis Karatzas. Text-DIAE: A self-supervised degradation invariant autoencoder for text recognition and document enhancement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37 (2):2330–2338, 2023. 8

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 2, 5

[50] Zhaoyi Wan, Minghang He, Haoran Chen, Xiang Bai, and Cong Yao. Textscanner: Reading characters in order for robust scene text recognition. In *AAAI*, pages 12120–12127, 2020. 3

[51] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. Decoupled attention network for text recognition. In *AAAI*, pages 12216–12224, 2020. 1, 6, 7

[52] Yuxin Wang, Hongtao Xie, Shancheng Fang, Jing Wang, Shenggao Zhu, and Yongdong Zhang. From two to one: A new scene text recognizer with visual language modeling network. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14174–14183, 2021. 2

[53] Mingkun Yang, Minghui Liao, Pu Lu, Jing Wang, Shenggao Zhu, Hualin Luo, Qi Tian, and Xiang Bai. Reading and writing: Discriminative and generative modeling for self-supervised text recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, page 4214–4223, New York, NY, USA, 2022. Association for Computing Machinery. 3, 8

[54] Mohamed Yousef and Tom E. Bishop. OrigamiNet: Weakly-supervised, segmentation-free, one-step, full page text recognition by learning to unfold. In *2020 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 14698–14707, 2020. 7

[55] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, pages 12310–12320, 2021. 3, 4, 5

[56] Xiaoyi Zhang, Tianwei Wang, Jiapeng Wang, Lianwen Jin, Canjie Luo, and Yang Xue. ChaCo: Character contrastive learning for handwritten text recognition. In *Frontiers in Handwriting Recognition*, pages 345–359, Cham, 2022. Springer International Publishing. 8

[57] Xiaoyi Zhang, Jiapeng Wang, Lianwen Jin, Yujin Ren, and Yang Xue. CMT-Co: Contrastive learning with character movement task for handwritten text recognition. In *Computer Vision – ACCV 2022*, pages 626–642, Cham, 2023. Springer Nature Switzerland. 8

[58] Wenqi Zhao and Liangcai Gao. CoMER: Modeling coverage for transformer-based handwritten mathematical expression recognition. In *ECCV*, pages 392–408, 2022. 5, 6

[59] Wenqi Zhao, Liangcai Gao, Zuoyu Yan, Shuai Peng, Lin Du, and Ziyin Zhang. Handwritten mathematical expression recognition with bidirectionally trained transformer. In *ICDAR*, pages 570–584, 2021. 5

[60] M. Zimmermann and H. Bunke. Automatic segmentation of the iam off-line database for handwritten english text. In *ICPR*, pages 35–39, 2002. 3