On the Generalization of Handwritten Text Recognition Models

Carlos Garrido-Munoz
Univeristy of Alicante
carlos.garrido@ua.es

Jorge Calvo-Zaragoza
University of Alicante
jcalvo@dlsi.ua.es

Abstract

Recent advances in Handwritten Text Recognition (HTR) have led to significant reductions in transcription errors on standard benchmarks under the i.i.d. assumption, thus focusing on minimizing in-distribution (ID) errors. However, this assumption does not hold in real-world applications, which has motivated HTR research to explore Transfer Learning and Domain Adaptation techniques. In this work, we investigate the unaddressed limitations of HTR models in generalizing to out-of-distribution (OOD) data. We adopt the challenging setting of Domain Generalization, where models are expected to generalize to OOD data without any prior access. To this end, we analyze 336 OOD cases from eight state-of-the-art HTR models across seven widely used datasets, spanning five languages. Additionally, we study how HTR models leverage synthetic data to generalize. We reveal that the most significant factor for generalization lies in the textual divergence between domains, followed by visual divergence. We demonstrate that the error of HTR models in OOD scenarios can be reliably estimated, with discrepancies falling below 10 points in 70% of cases. We identify the underlying limitations of HTR models, laying the foundation for future research to address this challenge. Code is available at github.com/carlos10garrido/HTR-OOD.

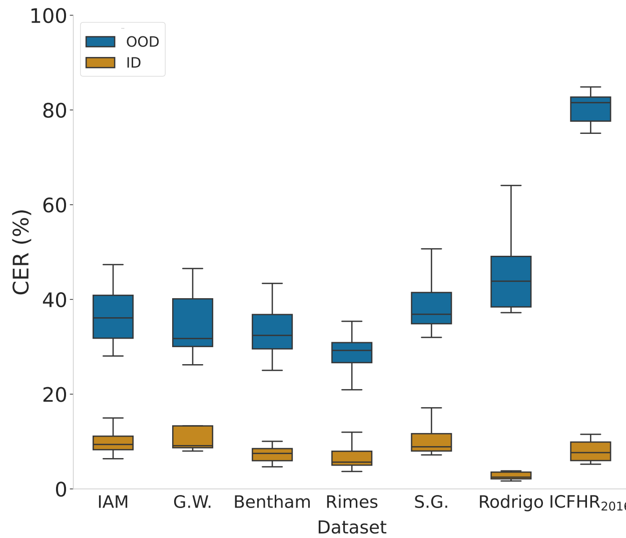

Figure 1. Average performance of HTR models for in-distribution (ID) and out-of-distribution (OOD) scenarios. The significant performance drop in OOD scenarios highlights the limited generalization capability of current models. Details in Sect. 4.

1. Introduction

Handwritten text plays a central role in transmitting human knowledge and culture across generations. Despite the advancements in digital storage, significant volumes of handwritten documents remain to be transcribed and thus inaccessible for further analysis or retrieval. The field of Handwritten Text Recognition (HTR) focuses on automatically transcribing handwritten text into digital format. HTR has wide-range applications, including the preservation of cultural heritage [52], digital transcription of historical and matrimonial records [69], digitization of personal and official documents [66], and even real-time digital note-taking through graphic tablets [67]. However, this transcription process is challenging due to the vast variations in writing styles, different languages and alphabets, and even the mediums on which the text is written [35].

Perhaps influenced by the long-standing competitions in the field of HTR [2, 5, 76, 78], the latest improvements—from both models [10, 11, 20, 39, 46, 47, 64] and data generation [17, 36, 44, 61, 82]—have been oriented towards increasing performance in the test sets of standard benchmarks, relying on the assumption that test data will conform to the same source distribution. This adheres to the independent and identically distributed (i.i.d) condition, which is often overlooked in real-world settings [12, 13, 27]. However, as demonstrated by our preliminary results shown in Fig. 1, the performance gap between in-distribution (ID) data and out-of-distribution (OOD) data in HTR is particularly remarkable. This phenomenon is yet to be explored in this field. Although this generalization issue remains largely unexplored, the literature recognizes the importance of gener-

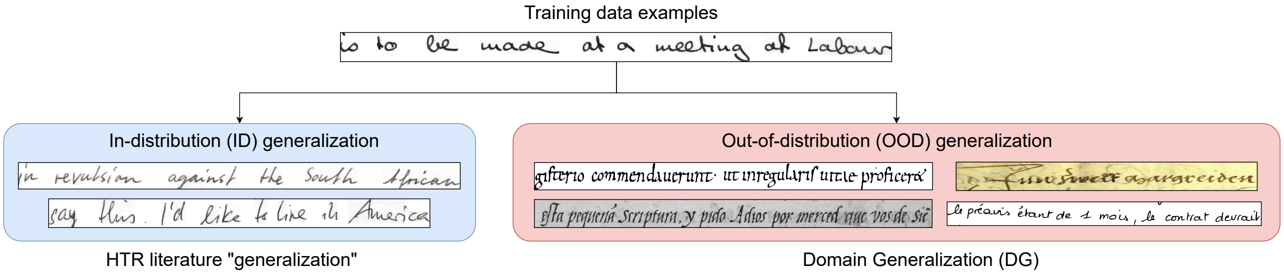

Figure 2. Illustration of the different approaches for the concept of "generalization" in HTR literature. While the field commonly considers in-distribution (ID) train–test scenarios as 'generalization' (variations within similar scripts and languages), true out-of-distribution (OOD) generalization—encompassing vastly different scripts, languages, and historical periods—is not addressed.

alizing beyond the training data and focuses on two main approaches: (1) Transfer Learning (TL), where pre-trained models are fine-tuned on a target domain [79, 87, 98], and (2) Unsupervised Domain Adaptation (UDA), as a particular case of TL, that aims to adapt prior knowledge to new unlabeled data [85, 91] in synth-to-real scenarios [38] or with real data [16, 73]. However, TL approaches in HTR are constrained to scenarios where the source and target domains share similar characteristics [6, 43] or limited to a single architecture [5]. Likewise, existing UDA studies are limited by small datasets [73, 96] or focus on word-level tasks [16, 38], restricting their applicability to general HTR at the line level. In this work, departing from the TL and UDA literature, we rather adopt a Domain Generalization (DG) framework, which aims to generalize to OOD data—data significantly different from the training set—without any prior access to it [84, 97].

Given the scarcity and diversity of labeled datasets in the HTR field [35], we stress on a single-domain DG formulation [65, 86] moving away from the deceiving concept of "generalization" in the HTR literature (see Fig. 2). Our experimentation comprises two approaches. First, we provide practical insights into the generalization capabilities of HTR models, addressing questions such as: What is the best model in terms of generalization? Is there a general HTR method that outperforms others in generalization? In this first part, we also explore how key factors from the DG literature, such as model selection [33] and model capacity [29] enhance generalization performance. Additionally, we explore which models best leverage synthetic data to predict on real sets. The second part focuses on identifying the factors that most impact generalization in HTR models. To this end, we conduct a factor analysis [23, 74, 92] to reveal the most significant contributors. Before determining these factors, we address two critical challenges in HTR: (1) quantifying visual divergence between domains, which refers to measuring how different the visual features of characters or sentences are across various domains and (2) assessing textual disparities, which involves evaluating differences in the underlying linguistic content. These two challenges are addressed with two metrics—visual divergence and textual divergence—that intuitively should play a significant role in explaining OOD performance. Finally, leveraging the identified factors, we assess whether it is possible to estimate the amount of generalization error in advance, following the OOD literature on robustness [8, 55]. To mitigate potential biases in our conclusions, we analyze 336 distinct OOD evaluations under consistent and standardized conditions, examining 8 distinct HTR models from the literature across 7 datasets covering 5 different languages: English, Spanish, French, German, and Latin.

Our key contributions are: (1) We conduct the first large-scale analysis of HTR model performance evaluating both ID and OOD scenarios and provide the first comprehensive cross-lingual generalization study using real and synthetic datasets; (2) We introduce and analyze proxy metrics for explaining HTR performance in OOD scenarios; (3) We identify key factors that most influence generalization in HTR; (4) We demonstrate that OOD error can be estimated robustly from the considered proxies; (5) We provide a thorough generalization analysis framework in HTR that provides the groundwork for future research in the field.

2. Related Work

2.1. Handwritten Text Recognition

The use of bidirectional Long Short-Term Memory (LSTMs) networks [31, 34] with Connectionist Temporal Classification (CTC) [32] has been dominating the state of the art in HTR competitions for decades [2, 5, 76, 78]. Despite this prevalence of CTC, attention-based encoder-decoder approaches [9] have recently gained popularity because of their competitive results [1, 20, 40, 45]. The work of Michael et al. [54] provides a comprehensive study of several sequence-to-sequence approaches for HTR. There has also been a growing interest in the use of more scalable and parallelizable architectures such as the Transformer [83] by adapting the work of [25] (Vision Transformer) to the field [11, 24, 28, 58, 63, 70]. HTR has benefited from this adaptation, either in isolation with an encoder-decoder [46, 56, 57, 90] or in combination with the CTC objective function [10, 21, 89]. The work of Diaz et al. [24] explores universal architectures for text recognition, concluding that a CNN backbone with a Transformer encoder, a CTC-based decoder, plus an explicit language model, is the most effective strategy to date. Regardless of this progress, however, the need for large labeled corpora as a pre-training strategy in Transformer-based models has become noticeable [24, 46, 49, 50, 95]. This issue could be mitigated through the use of Self-Supervised Learning (SSL) training methods [3, 60, 94].

2.2. Adaptation and Generalization in HTR

Transfer Learning (TL). Transfer Learning [79, 87, 98] has emerged as a popular approach to improve HTR systems, particularly when dealing with small datasets or adapting models to new domains. [6] investigated the knowledge transfer from larger datasets to smaller ones, focusing on which layers of neural networks require retraining. In a follow-up study, [4] further explored the combination of Domain Adaptation (DA) and TL, concluding that while it yields the best results, TL alone can achieve nearly comparable performance. Aligned with these findings, [43] demonstrated that fine-tuning is surprisingly effective as a domain adaptation baseline in handwriting recognition, focusing on architectures using CTC. [62] explored pre-training strategies for HTR models, including the use of synthetic data and data generated by Handwritten Text Generation (HTG) [61, 82], evaluating scenarios where only the target language or author information is known.

Domain Adaptation (DA). The closest field to our research is that of Unsupervised Domain Adaptation (UDA) [85, 91], sometimes referred to as Writer Adaptation (WA) in the HTR literature, which investigates techniques with which to adapt HTR models trained on samples from either one or multiple writers to unseen writers [43, 73, 80, 81, 96, 97] or adapting synth-to-real data [38]. This field explores adaptation to new data by assuming access to unlabeled samples from the target domain. In contrast, our work is more akin to that of Domain Generalization (DG) [33, 65, 84, 86, 97], where there is no access to any type of data from the OOD target. To the best of our knowledge, this scenario has not been explored in HTR. Nonetheless, [62] introduces some close-to-DG experiments, focusing on examining TL from several real-world source datasets to only a few real-world target datasets, without deeply exploring the generalization capabilities of these models. Instead, authors use this scenario to compare various strategies using different percentages of target data, blending both synthetic and real datasets within a TL framework. In our work, we rather focus on studying the generalization capabilities of HTR models to entirely new manuscripts without making assumptions about the availability of specific unlabeled data from the target domain. This approach is more challenging (as depicted in Fig. 2), since the robustness to the OOD scenario has to be performed without prior information about the new domain. We emphasize that in this work we do not apply any type of adaptation to any target dataset, in contrast to the WA/DA field.

3. Methodology

3.1. HTR formulation

HTR models take an image $\mathbf{x}$ and predict a sequence $y = (y_1, y_2, \dots y_L)$, where each y_i is a character from an alphabet Σ. Modern HTR models are trained in an end-to-end fashion to align image-text pairs $(\mathbf{x}, \mathbf{y})$ from a training set $\mathcal{D}_{\text{train}}$. Then, they are evaluated on a test set $\mathcal{D}_{\text{test}}$ to measure "generalization" performance. In this paper, we explore the generalization capabilities of HTR models in unseen domains, addressing a more challenging task than previously reported in the literature and aligning more closely with the DG paradigm, as illustrated in Fig. 2. Note that this type of generalization extends beyond the typical author-to-author generalization in HTR [16, 18, 73] and addresses new OOD challenges such as new manuscripts, alphabets, and languages.

3.2. Baseline HTR Models

We have selected eight distinct architectures [10, 11, 20, 26, 37, 47, 54, 64] specifically designed for HTR, representing all this broad spectrum. Despite the number of methods, these can be primarily organized by their approach to align the input image x with the output sequence y, fitting into one of the three main categories: (1) Connectionist Temporal Classification (CTC) [20, 26, 47, 64], (2) Sequence-to-Sequence models [37], or (3) Hybrid models [10, 11, 54].

CTC. Most architectures in the modern HTR literature employ the CTC objective [32] as an effective method for aligning images with unsegmented transcriptions. We selected four distinct architectures with different feature extractors. These architectures are: CRNN [64], VAN [20], CNN-SAN [26], HTR-VIT [47]. This selection comprises different CTC-based architectures with a broad range of encoders and decoders.

Sequence-to-sequence. Sequence-to-sequence models (Seq2Seq) have been dominating the arena of language processing [9], particularly boosted by the introduction of the Transformer [83].[1] To investigate this type of Seq2Seq

[1] In this work, we do not include TrOCR [46], as it was not originally designed for HTR but rather as a secondary application.

architectures, we utilize the Transformer presented in [37] (ResNet + Transformer) since other pure Transformer-based architectures merely differ in post-processing techniques [90].

Hybrid. Several studies have adopted hybrid approaches for HTR [10, 11, 54]. Inspired by [42], which combined CTC for the encoder and Cross-Entropy (CE) for the decoder as a multitask learning for Speech Recognition, a similar method was first applied in HTR in [54]. This approach utilizes a hybrid loss function that merges CTC and CE losses as $\mathcal{L} = \lambda\mathcal{L}_{\text{ctc}} + (1-\lambda)\mathcal{L}_{\text{ce}}, \lambda \in [0,1]$,

with λ typically set to 0.5. This combined loss has proven beneficial in HTR, as shown in [54]. Additionally, recent trends in HTR explore transcription as a multitask learning problem, particularly with lightweight CNN and Transformer architectures [10, 11]. We explore this late trend in HTR throughout the architectures proposed in [10, 11, 54].

Table 1 summarizes the architectures considered in this work.

Table 1. Description of HTR architectures considered, including alignment type, number of parameters, and input image sizes. C=CNN; FCN=Fully Convolutional Network; SA=Self Attention; T=Transformer; Att="Classic" attention.

Model	**Architecture**	**Align.**	**Params**	**Input size**
CRNN [64]	CRNN	CTC	9.6M	128,1024
VAN [20]	FCN	CTC	2.7M	64,1024
C-SAN [26]	C+SA	CTC	1.7M	128,1024
HTR-VT [47]	C+ViT	CTC	53.5M	64, 512
Kang [39]	ResNet+T	Seq2Seq	90M	64,2227
Michael [54]	CRNN+Att.Dec	Hybrid	5M	64,1024
LT [10]	C+T.Enc+CTC	Hybrid	7.7M	128,1024
VLT [11]	C+T.Enc+CTC	Hybrid	5.6M	128,1024

3.3. Experimental setup

Data. We assessed the performance of the models in the following datasets: IAM [52], Rimes [41], Bentham [19], Saint-Gall (S.G.) [71], George Washington (G.W.) [66], Rodrigo [72] and ICFHR 2016 (READ 2016) [77]. Table 2 details each dataset.

Performance metrics. As the majority of works in the field, we assessed the performance of the models using the common Character Error Rate (CER).[2] This ensures consistency with existing research and facilitates a more direct comparison with our findings. Additionally, we computed

[2]Results with the Word Error Rate will also be reported in Appendix 7.

Table 2. Handwritten document datasets considered, with an indication of the language, historical period, number of writers, number of samples in each partition, and size of the alphabet (Σ).

Dataset	**Lang.**	**Period**	**No. writ**	**Train**	**Val**	**Test**	**—Σ—**
IAM	En	1999	657	6.4K	976	2.9K	79
Rimes	Fr	2011	1.3K	10K	1.1K	778	100
Bentham	En	18-19th c.	1	9.1K	1.4K	860	91
S.G.	Lat	9-12th c.	1	1.4K	235	707	49
G.W.	En	1755	1	325	168	163	68
Rodrigo	Sp	1545	1	20K	1K	5K	115
ICFHR_{2016}	De	15-19th c.	*unk.*	8.2K	1K	1K	91

the Expected Calibration Error (ECE) for each model, following the definition in [7]. We aim to assess not only the accuracy of the models but also their capacity to generate calibrated probability outputs. This is critical for understanding model behavior and uncertainty when applied to unseen OOD data, thereby offering a more comprehensive evaluation of model performance.

Implementation details. Each model was trained from scratch for 500 epochs, with the best model saved based on validation performance (CER) for the corresponding domain. Training is always performed with a single source domain, and the run is stopped if the CER on the validation set does not improve at least 0.1 within 100 epochs. We implemented a comprehensive set of typical data augmentation techniques from HTR literature [51, 68]: rotation, dilation, erosion, random perspective, elastic transformation, shearing, and Gaussian noise, applied with a probability of 0.5. Lastly, we grayscaled all images as in [64]. Hyperparameters such as architecture, input size, batch size, optimizers, and schedulers were consistent with the original configurations. Further details on data augmentation and hyperparameters are provided in Appendix 8. Given that each dataset contains a unique alphabet, we combined all alphabets using Unicode[3] to standardize evaluations, resulting in a vocabulary of 94 characters including special symbols for the beginning of sequence [BOS], padding [PAD], unknown [UNK], and the end of sequence [EOS]. No post-processing techniques or lexicon-based predictions were applied to the outputs of the models.

4. Practical out-of-distribution insights

We first present the experiments conducted to explore the practical implications of our large-scale study on generalization in HTR. Specifically, we aim to answer the following key questions regarding the studied architectures: (1) Is

[3]https://pypi.org/project/Unidecode/

Table 3. In-distribution (ID) and out-of-distribution (OOD) results (CER %) for HTR models across datasets. The OOD result is reported from the best-performing source. Results marked with $*$ indicate outliers, meaning that the model did not converge in the ID setting. Average results (bottom row) are computed filtering out outliers. † denotes architectures implemented from the papers (no code provided).

Dataset	CRNN [64]		VAN [20]		C-SAN† [26]		HTR-VT [47]		Kang† [39]		Michael† [54]		LT† [10]		VLT† [11]	
	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD
IAM	6.4	34.9(+28.5)	6.6	28.6(+22.0)	15.0	31.5(+16.6)	5.8	33.7(+27.9)	8.0	42.1(+34.1)	7.5	49.1(+41.6)	7.9	42.0(+34.1)	8.9	41.3(+32.4)
Rimes	3.7	25.0(+21.2)	5.6	21.3(+15.6)	12.0	29.8(+19.8)	7.9	28.3(+20.4)	5.7	32.0(+26.3)	6.9	35.5(+28.6)	5.0	30.8(+25.8)	5.1	29.4(+24.3)
G.W.	8.2	31.1(+22.9)	9.3	32.0(+22.7)	9.0	49.8(+40.8)	34.9*	38.6	78.4*	44.0	53.8*	43.6	79.6*	32.3	25.2*	32.1(+6.9)
Bentham	4.7	25.3(+20.6)	7.4	26.6(+19.2)	10.0	38.9(+28.9)	8.4	33.3(+24.9)	8.5	39.4(+30.9)	8.5	43.5(+34.9)	6.0	33.8(+27.8)	6.1	33.3(+27.2)
S.G.	7.2	33.6(+26.3)	7.8	39.8(+32.0)	8.6	35.0(+26.4)	17.1	36.5(+19.3)	78.7*	51.8	76.9*	55.3	12.5	37.8(+25.3)	9.2	38.7(+29.4)
Rodrigo	1.7	40.9(+39.3)	2.3	38.5(+36.2)	3.8	45.2(+41.4)	3.9	38.5(+34.6)	2.6	60.6(+57.9)	3.8	65.3(+61.5)	2.0	48.4(+46.4)	2.2	47.4(+45.3)
ICFHR$_{2016}$	5.2	78.7(+73.5)	7.5	75.3(+67.8)	17.0	83.4(+66.4)	11.6	79.6(+68.0)	7.8	92.6(+84.8)	9.5	85.2(+75.7)	5.9	90.5(+84.6)	6.0	85.1(+79.1)
Average	**5.3**	38.5(+33.2)	6.7	**37.4**(+30.8)	9.7	44.8(+35.4)	7.5	41.2(+33.7)	6.5	51.8(+45.3)	7.2	53.9(+46.7)	6.5	45.1(+38.5)	6.3	43.9(+37.6)

there a model or alignment strategy that consistently outperforms others in terms of OOD performance? (2) How do model selection and capacity impact OOD performance? Lastly, considering the increasing use of synthetic data in HTR models, we conduct experiments to address the following: (1) Which model or alignment makes the best use of synthetic data to predict real HTR data? and (2) How does the internal Language Model (LM) impact HTR performance?

4.1. HTR performance

The results of the first experiments are reported in Table 3, where the rows represent the target domain, the first column indicates the ID error, and the OOD column shows the best error achieved from any other source domain. In this scenario, the typical ID model selection is applied by choosing the one with the best validation error in the source domain. The average performance of both models in the two scenarios (ID, OOD) is presented in the last row, with the best results highlighted in bold. We compare three state-of-the-art VLMs [22, 48, 59] and TrOCR [46] in a zero-shot setting in Appendix 7, highlighting their limitations for HTR.

OOD results are terrible. OOD results are poor across all architectures and alignment types, with CER values ranging from 37.4% to 53.9%—error rates that are notably high for a transcription system. The model that performs best in terms of generalization is VAN, showing only a slight absolute improvement of 1% over the second-best model (CRNN), which also uses CTC alignment. When comparing the best model (VAN) to the worst model (Michael) in terms of generalization, the absolute improvement is only 16.5%. The average difference between ID and OOD CERs is 37.6%, posing that generalization gaps in HTR remain significant even under the best OOD conditions.

Alignment type does not really matter, but choose CTC. OOD performance is ordered and color-coded according to the alignment type in Fig. 3. While the observed differences are not substantial, a macro-level analysis reveals that models utilizing CTC alignment exhibit the best generalization performance, followed by hybrid approaches, and finally, purely Seq2Seq models. The results suggest that, within the DG scenario, a CTC-alignment model is the most effective choice. Notably, even the model with the poorest OOD performance (C-SAN) is comparable to the second-best alignment type (hybrid approach, VLT), with CER performances of 44.8% and 43.9%, respectively. Grouping models by their decoding method—CTC or autoregressive (AR)—we observe that CTC-based models achieve an average CER of 40.4, while AR models have a CER of 48.7, resulting in a relative drop of 20%.

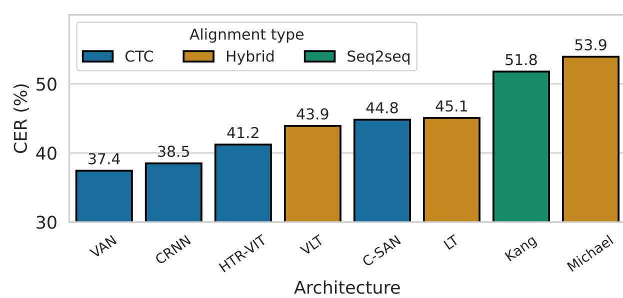

Figure 3. Average CER in the out-of-distribution (OOD) scenario where each color represents the alignment type (sorted by performance).

4.2. Leveraging synthetic data

This section investigates how HTR models utilize synthetic data for predicting on the typical HTR datasets. The study has two primary objectives: (1) to assess the effectiveness

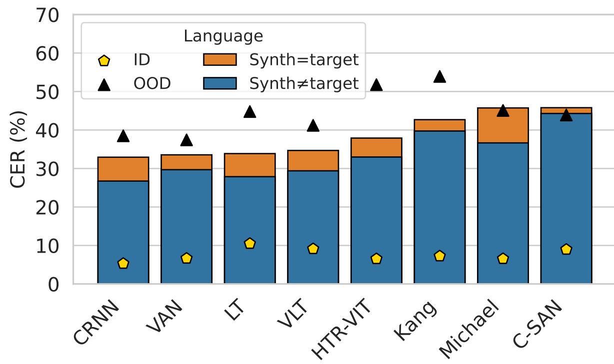

Figure 4. Average performance of the HTR models trained with synthetic data across real historical datasets. Blue bars represent performance when training with the same language as target, whereas orange represents the best result with any other language. ID (green) and OOD (black) performances training with real data are provided as references.

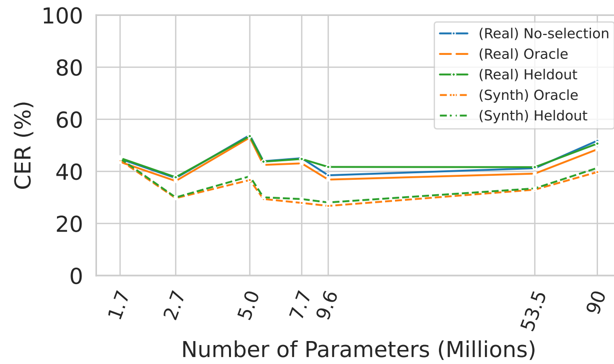

Figure 5. Average CER (%) in out-of-distribution (OOD) performance for HTR architectures according to the number of parameters (millions). X-axis is in log-scale for better visualization.

of each architecture in leveraging synthetic data relative to real data, and (2) to quantify the influence of the underlying language model (LM) on the overall performance. We designed controlled experiments to evaluate the effect of synthetic text generation across multiple languages on real datasets' performances. We generated synthetic 100K text lines randomly selected from WIT dataset [75] in English, French, Spanish, German, and Latin using 4,000 publicly available fonts[4] with handwritten styles ensuring an equal number of lines for each language. We excluded HTG methods [17, 61, 82] as they rely on labeled data targeting ID performance, limiting their applicability to DG. For this scenario, we compare the performance of models trained on synthetic datasets rendered in the same language as the target versus those trained on a different language. To ensure fair comparisons, we consistently selected the best-performing models based on validation results for the target dataset regarding the language used. The results are shown in Fig. 4, where results are sorted from best to worst (left-to-right) to facilitate visualization. We also report the full results with synthetic data in Appendix 7 (Table 7).

Synthetic data (slightly) improves OOD, even blindly. The first observation in Fig. 4 is that using synthetic data consistently improves OOD performance compared to real data, even when a different language is used from the target domain. Although the CER remains high, ranging from 32.9 to 45.7, models generalize an average of 6 points better when using synthetic data. In the best case, where the target domain language is known, models show an absolute gain of over 11 CER points.

[4]Fonts were sourced from open-access repositories like 1001fonts.com.

Choose CTC again (AR-models are more biased). Fig. 4 shows that the model that best utilizes synthetic data is the CRNN architecture, followed by VAN. While the differences are less pronounced compared to the OOD scenario using real data, the conclusion remains the same: CTC alignment is still the best option. When grouping models by their decoding method, whether CTC or autoregressive (AR), we find that the relative decrease compared to training with synthetic data—same or a different language—is 25% for AR and 19% for CTC. This intuitively reflects the nature and bias of AR models: they are explicitly trained to compute an LM, unlike CTC models, making the latter more robust to language variations.

4.3. Model selection and capacity

In this section, we evaluate the performance of HTR models through two key factors from the DG literature: model selection [33] and model complexity [29]. We compare three OOD model selection strategies inspired by DG studies: (1) *No-selection* (ID selection), where the best model is chosen based on source validation performance (standard i.i.d. setting, as shown in Table 3) used as a baseline; (2) *Held-out selection*, where model selection is based on the average performance across all validation data sources excluding the target dataset [33]; (3) *Oracle-based selection*, which sets an upper performance bound by selecting the best model based on validation results from the target dataset [33]. As with prior analyses, OOD performance reflects the best result, regardless of the source domain used for training. We present the results in Fig. 5, where the capacity of the HTR models (measured in millions of parameters) is compared to their OOD performance across the three model selection methods using real and synthetic data.

Model selection has no impact. Fig. 5 demonstrates that the choice of selection method has a negligible impact on OOD performance in the HTR models, with the results across the various selection strategies being practically identical. Contrary to findings in the broader (DG) literature [33], the oracle-based selection method does not yield substantial advantages over other strategies. These results, when applied to the oracle model, suggest that in the DG scenario, no "sweet spot" exists for selecting a model that consistently exhibits superior generalization performance.

More capacity is not useful. It can also be observed from Fig. 5 that an increase in model capacity (measured by the number of parameters) does not correspond to an improvement in the generalization capabilities in OOD scenarios, suggesting that increased model complexity does not necessarily translate to better generalization performance.

5. Factor analysis of out-of-distribution

In the previous sections, we analyzed the performance of HTR models in OOD scenarios from a practical perspective and compared key aspects of DG literature. In this section, we focus on identifying the hidden factors that most influence OOD generalization in HTR. To achieve this, we conducted a Factor Analysis [23, 74, 92], where the goal is to find the minimum set of latent variables, or factors, sufficient to explain the data and establish dependencies between features. The features in this analysis consist of the following base metrics that could potentially explain the OOD performance [55, 88]: model capacity (measured by the number of parameters), ID and OOD errors (measured by CER), and model ID and OOD calibration errors (ECE). However, given that HTR lies in the intersection between image and text processing, we consider two novel features that must be relevant in this context: visual and textual divergence between domains.

Visual divergence. To quantify the visual differences between domains, we approached this problem as an Anomaly Detection (AD) task [93]. Following the reconstruction-based literature [14, 15, 53], we measure whether a sample of a target T is out-of-distribution based on the reconstruction error of an Autoencoder (AE) trained on a source S that we denote as ϕ_{θ_S}. In this work, we trained a simple convolutional AE (ϕ_{θ_S}) for each source dataset (details and results in Appendix 9), and we compute $\Delta(X_S, X_T)$ as the average reconstruction error (measured in MSE) as:

$$\Delta(X_S, X_T) = \frac{1}{m}\sum_{i=1}^{m}\left\|\phi_{\theta_S}(X_i^{(T)}) - X_i^{(T)}\right\|^2$$

on the target dataset's images X_T. Therefore, $\Delta(X_S, X_T)$ estimates how far a dataset deviates from the original source domain distribution. Concretely, $\Delta_S(X_S, X_T)$ quantifies visual divergence between train-test splits from same source S and $\Delta_T(X_S, X_T)$ for different S (train) and T (test).

Textual divergence. To quantify the divergence between two textual distributions (Y_S, Y_T), we considered the averaged KL-divergence across varying n-grams [30, 62]. Specifically, we compute $\Delta(Y_S, Y_T)$ as the average KL-divergence for each n-gram from $n = 1 \ldots 5$:

$$\Delta(Y_S, Y_T) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j \in V_n} D_{KL}^{(n)}(P_j^{(S)} \parallel Q_j^{(T)})$$

where V_n is the set of n-grams for the source vocabulary and $D_{KL}^{(n)}(P_j^{(S)} \parallel Q_j^{(T)})$ represents the KL divergence between the n-gram j in the source and target distributions. Consequently, $\Delta(Y_S, Y_T)$ approximates how "unlikely" the text in the target domain is relative to the source domain. In our metrics, we denote $\Delta_L(Y_S, Y_T)$ as the divergence between the source text domain Y_S and a synthetic target domain T (the same ones used in the synthetic experiments) where the language L is known. We refer to $\Delta_{GT}(Y_S, Y_T)$ as the actual divergence (see Appendix 9) between the source domain Y_S and the ground truth of the target domain Y_T.

5.1. Main latent factors

Thus, with all metrics collected (model parameters, visual divergence, textual divergence, ID and OOD errors and calibrations) and after applying standard preprocessing to normalize the columns, we extracted the k eigenvectors corresponding to the k eigenvalues ≥ 1. In our case, $k = 4$ (more details in Appendix 10). After testing various rotations (although results did not differ significantly) the most interpretable explanation was achieved by *oblimax* rotation. Results are presented in Fig. 6, where color-coded (Pearson's) correlations illustrate how each metric relates to the four hidden factors.

Factor 1: Textual divergence explains most of the OOD error. Factor 1 shows a strong positive correlation with textual metrics for both ground-truth $\Delta_{GT}(Y_S, Y_T)$ (0.92) and generic texts $\Delta_L(Y_S, Y_T)$ (0.84). This suggests that Factor 1 is heavily linked to textual divergences, highlighting it as a significant contributor to OOD error. Noteworthy, Factor 1 shows a moderate positive correlation with OOD error (0.62), implying that greater textual misalignment between domains leads to increased difficulty in generalizing for HTR models. Additionally, Factor 1 shows positive correlations OOD ECE (0.52), suggesting that it may also capture aspects of calibration linked to textual divergence, although to a lesser extent. Overall, these results underscore that alignment in textual content between source and target domain is crucial for reducing OOD error in HTR models.

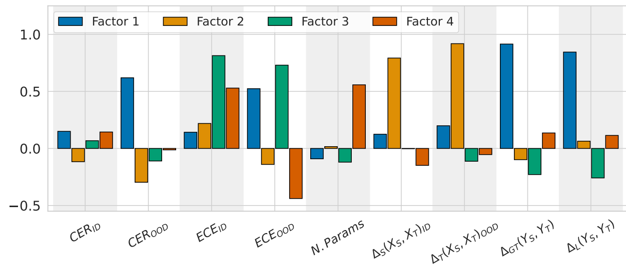

Figure 6. Factor loadings (contributions) of different metrics based on a factor analysis with 4 orthogonal (color-coded) factors: Factor 1 (blue): Textual divergence; Factor 2 (orange): Visual divergence; Factor 3 (green): calibration quality; Factor 4 (red): calibration influenced by model complexity. CER_{OOD} is OOD error.

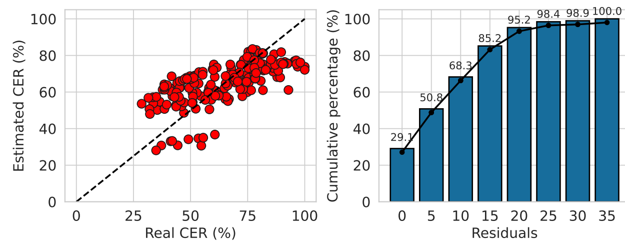

Figure 7. Left: Representation of the estimated vs. real CER values (MSE of 10.9 on average). Right: Cumulative distribution of grouped residuals. Approximately 70% of predictions yield an error below 10 points of CER.

Factor 2: Visual similarity has a modest influence on OOD Error: Factor 2 has a high positive correlation with ID reconstruction error $\Delta_S(X_S, X_T)$ (0.8) and OOD reconstruction error $\Delta_T(X_S, X_T)$ (0.92) indicating that this factor predominantly reflects visual divergence between domains. Interestingly, Factor 2 also shows a weak to moderate negative correlation with OOD error (-0.3). This suggests that while visual divergence may not be a dominant factor in predicting OOD error, it does exert a modest influence. This relationship implies that models experience some performance degradation as the visual divergence between source and target domains increases, albeit this effect is weaker compared to the impact of textual divergence.

Factor 3 and Factor 4: Calibration quality and model complexity with differing effects on ID and OOD scenarios. Factor 3 is strongly correlated with both ID (0.81) and OOD (0.73) ECE, thus representing calibration accuracy. Weak links to model complexity and OOD error imply that this factor mainly affects prediction reliability in OOD scenarios without substantially influencing OOD error itself. Factor 4 correlates positively with ID ECE (0.53) and model parameters (0.56), suggesting it captures ID calibration influenced by model size. It has minimal effect on OOD error, indicating that while increased capacity enhances ID calibration, it does not translate to better generalization.

5.2. Can we predict OOD performance?

We established that the factors most affecting the generalization of models are those that quantify textual divergences. This section addresses a more practical question: Can OOD error be estimated from the proxy metrics investigated? To analyze this, we calculated the expected error using metrics that do not require target labels: ID error, ID ECE, number of parameters, reconstruction errors (ID and OOD), and KL divergence. The results are displayed in Fig. 7 (left), where the actual CER (X-axis) is plotted against the expected CER (Y-axis). The deviation from the black line represents the absolute error between the predicted and actual OOD values with an average deviation of 10.9 points, measured by Mean Absolute Error (MAE). We observe a clear positive correlation between the real and estimated CER values, suggesting that the model is generally successful in capturing the error trends, with higher predicted CER values tending to correspond with higher real ones. From a practical standpoint, the right plot in Fig. 7 presents the residuals distribution (measured by MAE), highlighting that nearly 30% of predictions have residuals between 0 and 5 and thus demonstrating high accuracy for a significant portion of samples. By the time the residuals reach the 10–15 range, about 70% of the predictions have been accounted for, indicating that most predictions are reasonably close to the true CER.

6. Conclusions

This paper provides a comprehensive analysis of generalization in Handwritten Text Recognition (HTR) models, addressing the significant gap in understanding performance in out-of-distribution (OOD) scenarios. We conduct the first large-scale evaluation of HTR model performance, analyzing OOD results across multiple datasets and architectures, first providing practical insights for researchers in the HTR field. These experiments suggested that greater emphasis should be placed on enhancing the generalization capabilities of HTR models, as no architecture or alignment type in the literature currently facilitates effective generalization. Additionally, the results indicated that when utilizing synthetic data, greater benefits are likely to be achieved with architectures based on alignment CTC. Furthermore, leveraging proxy metrics through factor analysis, we identified that the primary factor contributing to OOD error is the textual divergence between source and target, with a weaker contribution of the visual divergence factor. We emphasize that more research studying this last factor has to be done. Finally, we also found that these proxy metrics can robustly predict generalization errors with considerable precision.

References

[1] Abdelrahman Abdallah, Mohamed A. Hamada, and Daniyar Nurseitov. Attention-based fully gated cnn-bgru for russian handwritten text. *Journal of Imaging*, 2020. 2

[2] Haikal El Abed, Volker Märgner, and Michael Blumenstein. International Conference on Frontiers in Handwriting Recognition (ICFHR 2010) - Competitions Overview. 2010. 1, 2

[3] 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. *Computer Vision and Pattern Recognition*, 2021. 3

[4] Jose Carlos Aradillas, Juan Jose Murillo-Fuentes, and Pablo M. Olmos. Boosting Offline Handwritten Text Recognition in Historical Documents With Few Labeled Lines. *IEEE Access*, 9:76674–76688, 2021. 3

[5] Jose Carlos Aradillas, Juan Jose Murillo-Fuentes, and Pablo M. Olmos. Boosting offline handwritten text recognition in historical documents with few labeled lines. *IEEE Access*, 2021. 1, 2

[6] Jose Carlos Aradillas Jaramillo, Juan Jose Murillo-Fuentes, and Pablo M. Olmos. Boosting Handwriting Text Recognition in Small Databases with Transfer Learning. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 429–434, Niagara Falls, NY, USA, 2018. IEEE. 2, 3

[7] Eric Ayllon, Francisco J. Castellanos, and Jorge Calvo-Zaragoza. Analysis of the calibration of handwriting text recognition models. In *Document Analysis and Recognition - ICDAR 2024*, pages 139–155, Cham, 2024. Springer Nature Switzerland. 4

[8] Christina Baek, Yiding Jiang, Aditi Raghunathan, and Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. *Neural Information Processing Systems*, 2022. 2

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2, 3

[10] Killian Barrere, Yann Soullard, Aurélie Lemaitre, and Bertrand Coüasnon. A light transformer-based architecture for handwritten text recognition. 2022. 1, 3, 4, 5, 2

[11] Killian Barrere, Yann Soullard, Aurélie Lemaitre, and Bertrand Coüasnon. Training transformer architectures on few annotated data: an application to historical handwritten text recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 2024. 1, 3, 4, 5, 2

[12] Sara Beery, Grant Van Horn, and P. Perona. Recognition in terra incognita. *European Conference on Computer Vision*, 2018. 1

[13] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine-mediated learning*, 2010. 1

[14] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and C. Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *VISIGRAPP*, 2018. 7

[15] Paul Bergmann, Michael Fauser, David Sattlegger, and C. Steger. Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection. *Computer Vision and Pattern Recognition*, 2019. 7

[16] Ayan Kumar Bhunia, Shuvozit Ghose, Amandeep Kumar, Pinaki Nath Chowdhury, Aneeshan Sain, and Yi-Zhe Song. MetaHTR: Towards Writer-Adaptive Handwritten Text Recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15825–15834, Nashville, TN, USA, 2021. IEEE. 2, 3

[17] Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Mubarak Shah. Handwriting Transformers . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1066–1074, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 1, 6

[18] 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 *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 963–972, Montreal, QC, Canada, 2021. IEEE. 3

[19] Tim Causer and Valerie Wallace. Building a volunteer community: Results and findings from transcribe bentham. *Digital Humanities Quarterly*, 6(2), 2012. 4

[20] Denis Coquenet, Clement Chatelain, and Thierry Paquet. End-to-end handwritten paragraph text recognition using a vertical attention network. 2022. 1, 2, 3, 4, 5

[21] Denis Coquenet, Clément Chatelain, and Thierry Paquet. Dan: a segmentation-free document attention network for handwritten document recognition. 2023. 3

[22] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: towards general-purpose vision-language models with instruction tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2023. Curran Associates Inc. 5, 1

[23] N. Dehak, P. Kenny, Réda Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 2011. 2, 7

[24] Daniel Hernandez Diaz, Reeve Ingle, Siyang Qin, Alessandro Bissacco, and Yasuhisa Fujii. Rethinking text line recognition models. *arXiv*, 2021. 3

[25] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. 2

[26] Rafael d'Arce, Terence Norton, Sion Hannuna, and Nello Cristianini. Self-attention networks for non-recurrent hand-

written text recognition. In *Frontiers in Handwriting Recognition*, pages 389–403. Springer International Publishing, 2022. Series Title: Lecture Notes in Computer Science. 3, 4, 5, 2

[27] Sabri Eyuboglu, M. Varma, Khaled Kamal Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared A. Dunnmon, James Y. Zou, and Christopher R'e. Domino: Discovering systematic errors with cross-modal embeddings. *International Conference on Learning Representations*, 2022. 1

[28] Masato Fujitake. Dtrocr: Decoder-only transformer for optical character recognition. *arXiv.org*, 2023. 3

[29] Henry Gouk, Ondrej Bohdal, Da Li, and Timothy Hospedales. On the limitations of general purpose domain generalisation methods, 2024. 2, 6

[30] Evandro Gouvêa and Marelie H. Davel. Kullback-leibler divergence-based ASR training data selection. In *12th Annual Conference of the International Speech Communication Association, INTERSPEECH 2011, Florence, Italy, August 27-31, 2011*, pages 2297–2300. ISCA, 2011. 7

[31] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. IJCNN 2005. 2

[32] A. Graves, Santiago Fernández, Faustino J. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. *ICML*, 2006. 2, 3

[33] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *International Conference on Learning Representations*, 2021. 2, 3, 6, 7

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

[35] R. Reeve Ingle, Yasuhisa Fujii, Thomas Deselaers, Jonathan Baccash, and Ashok C. Popat. A Scalable Handwritten Text Recognition System . In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 17–24. IEEE Computer Society, 2019. 1, 2

[36] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv: Computer Vision and Pattern Recognition*, 2014. 1

[37] Lei Kang, Pau Riba, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. Pay attention to what you read: Non-recurrent handwritten text-line recognition. *arXiv (Cornell University)*, 2020. 3, 4

[38] Lei Kang, Marcal Rusinol, Alicia Fornes, Pau Riba, and Mauricio Villegas. Unsupervised Adaptation for Synthetic-to-Real Handwritten Word Recognition. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3491–3500, Snowmass Village, CO, USA, 2020. IEEE. 2, 3

[39] Lei Kang, Pau Riba, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. Pay attention to what you read: Non-recurrent handwritten text-line recognition. *Pattern Recognition*, 129:108766, 2022. 1, 4, 5, 2, 3

[40] Dmitrijs Kass and Ekta Vats. Attentionhtr: Handwritten text recognition based on attention encoder-decoder networks. 2022. 2

[41] Christopher Kermorvant and Jérôme Louradour. Handwritten mail classification experiments with the rimes database. In *International Conference on Frontiers in Handwriting Recognition, ICFHR 2010, Kolkata, India, 16-18 November 2010*, pages 241–246. IEEE Computer Society, 2010. 4

[42] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839, 2017. 4

[43] Jan Kohút, Michal Hradiš, and Martin Kišš. Towards Writing Style Adaptation in Handwriting Recognition, 2023. Version Number: 1. 2, 3

[44] Praveen Krishnan and C. V. Jawahar. Generating synthetic data for text recognition. *arXiv: Computer Vision and Pattern Recognition*, 2016. 1

[45] Lalita Kumari, Sukhdeep Singh, Vaibhav Varish Singh Rathore, Anuj Sharma, Lalita Kumari, Sukhdeep Singh, Vaibhav Varish Singh Rathore, and Anuj Sharma. Lexicon and attention based handwritten text recognition system. 2022. 2

[46] 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. *Proceedings of the ... AAAI Conference on Artificial Intelligence*, 2023. 1, 3, 5

[47] Yuting Li, Dexiong Chen, Tinglong Tang, and Xi Shen. HTR-VT: Handwritten text recognition with vision transformer. 158:110967, 2024. 1, 3, 4, 5, 2

[48] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 5, 1

[49] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Neural Information Processing Systems*, 2019. 3

[50] Zhihe Lu, Sen He, Xiatian Zhu, Li Zhang, Yi-Zhe Song, and Tao Xiang. Simpler is better: Few-shot semantic segmentation with classifier weight transformer. 2021. 3

[51] Canjie Luo, Yuanzhi Zhu, Lianwen Jin, and Yongpan Wang. Learn to augment: Joint data augmentation and network optimization for text recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13743–13752, 2020. 4

[52] Urs-Viktor Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 2002. 1, 4

[53] Shuang Mei, Hua Yang, and Z. Yin. An unsupervised-learning-based approach for automated defect inspection on textured surfaces. *IEEE Transactions on Instrumentation and Measurement*, 2018. 7

[54] Johannes Michael, R. Labahn, Tobias Grüning, and Jochen Zöllner. Evaluating sequence-to-sequence models for hand-

written text recognition. *IEEE International Conference on Document Analysis and Recognition*, 2019. 2, 3, 4, 5

[55] John Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Y. Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. *International Conference on Machine Learning*, 2021. 2, 7

[56] Saleh Momeni and B. BabaAli. A transformer-based approach for arabic offline handwritten text recognition. *arXiv.org*, 2023. 3

[57] Aly Mostafa, Omar Mohamed, Ali Ashraf, Ahmed Elbehery, Salma Jamal, Ghada Khoriba, and A. Ghoneim. Ocformer: A transformer-based model for arabic handwritten text recognition. *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, 2021. 3

[58] S. Paul, Gagan Madan, Akankshya Mishra, N. Hegde, Pradeep Kumar, and Gaurav Aggarwal. Weakly supervised information extraction from inscrutable handwritten document images. *arXiv*, 2023. 3

[59] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *ArXiv*, abs/2306.14824, 2023. 5, 1

[60] Carlos Peñarrubia, J. J. Valero-Mas, and Jorge Calvo-Zaragoza. Self-supervised learning for text recognition: A critical survey. *arXiv.org*, 2024. 3

[61] Vittorio Pippi, S. Cascianelli, and R. Cucchiara. Handwritten text generation from visual archetypes. *arXiv.org*, 2023. 1, 3, 6

[62] Vittorio Pippi, Silvia Cascianelli, Christopher Kermorvant, and Rita Cucchiara. How to Choose Pretrained Handwriting Recognition Models for Single Writer Fine-Tuning, 2023. Version Number: 1. 3, 7

[63] Jason Poulos and Rafael Valle. Character-based handwritten text transcription with attention networks. *Neural Computing and Applications*, 2021. 3

[64] Joan Puigcerver. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017*, pages 67–72. IEEE, 2017. 1, 3, 4, 5, 2

[65] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. *Computer Vision and Pattern Recognition*, 2020. 2, 3

[66] Toni M. Rath, R. Manmatha, and Victor Lavrenko. A search engine for historical manuscript images. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 369–376, New York, NY, USA, 2004. Association for Computing Machinery. 1, 4

[67] Janet C. Read. A study of the usability of handwriting recognition for text entry by children. *Interacting with Computers*, 19(1):57–69, 2007. Moving Face-to-Face communication to Web-based systems. 1

[68] George Retsinas, Giorgos Sfikas, Basilis Gatos, and Christophoros Nikou. Best practices for a handwritten text recognition system. In *Document Analysis Systems*, pages 247–259, Cham, 2022. Springer International Publishing. 4

[69] Verónica Romero, Alicia Fornés, Nicolás Serrano, Joan Andreu Sánchez, Alejandro H. Toselli, Volkmar Frinken, Enrique Vidal, and Josep Lladós. The esposalles database: An ancient marriage license corpus for off-line handwriting recognition. *Pattern Recognition*, 46(6):1658–1669, 2013. 1

[70] D. V. Sang and Le Tran Bao Cuong. Improving crnn with efficientnet-like feature extractor and multi-head attention for text recognition. *SoICT 2019*, 2019. 3

[71] Gustav Scherrer. *Verzeichniss der Handschriften der Stiftsbibliothek von St. Gallen*. Halle, 1875. 4

[72] Nicolas Serrano, Francisco Castro, and Alfons Juan. The RODRIGO database. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010. European Language Resources Association (ELRA). 4

[73] Yann Soullard, Wassim Swaileh, Pierrick Tranouez, Thierry Paquet, and Clement Chatelain. Improving Text Recognition using Optical and Language Model Writer Adaptation. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1175–1180, Sydney, Australia, 2019. IEEE. 2, 3

[74] C. Spearman. General intelligence objectively determined and measured. 1904. 2, 7

[75] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2443–2449, New York, NY, USA, 2021. Association for Computing Machinery. 6

[76] Joan Andreu Sánchez, Verónica Romero, A. Toselli, and E. Vidal. Icfhr2014 competition on handwritten text recognition on transcriptorium datasets (htrts). *2014 14th International Conference on Frontiers in Handwriting Recognition*, 2014. 1, 2

[77] Joan Andreu Sánchez, Verónica Romero, Alejandro H. Toselli, and Enrique Vidal. Icfhr2016 competition on handwritten text recognition on the read dataset. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 630–635, 2016. 4

[78] Joan-Andreu Sánchez, Verónica Romero, A. Toselli, M. Villegas, and E. Vidal. Icdar2017 competition on handwritten text recognition on the read dataset. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017. 1, 2

[79] Chuanqi Tan, F. Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. *International Conference on Artificial Neural Networks*, 2018. 2, 3

[80] Debapriya Tula, Sujoy Paul, Gagan Madan, Peter Garst, Reeve Ingle, and Gaurav Aggarwal. Is it an i or an l: Test-time Adaptation of Text Line Recognition Models, 2023. Version Number: 1. 3

[81] Tobias van der Werff, Maruf A. Dhali, and Lambert Schomaker. Writer adaptation for offline text recognition:

An exploration of neural network-based methods, 2023. Version Number: 1. 3

[82] Bram Vanherle, Vittorio Pippi, S. Cascianelli, Nick Michiels, F. Reeth, and R. Cucchiara. Vatr++: Choose your words wisely for handwritten text generation. *arXiv.org*, 2024. 1, 3, 6

[83] 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, 3

[84] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 2, 3

[85] Mei Legam Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 2018. 2, 3

[86] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[87] Karl R. Weiss, Taghi M. Khoshgoftaar, and Dingding Wang. A survey of transfer learning. *Journal of Big Data*, 2016. 2, 3

[88] Florian Wenzel, Carl-Johann Simon-Gabriel, David Kernert, Bernt Schiele, Andrea Dittadi, Max Horn, Chris Russell, Bernhard Schölkopf, Peter Gehler, Dominik Zietlow, Thomas Brox, and Francesco Locatello. Assaying out-of-distribution generalization in transfer learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2024. 7

[89] Christoph Wick, Jochen Zöllner, and Tobias Grüning. Rescoring sequence-to-sequence models for text line recognition with ctc-prefixes. *arXiv: Computer Vision and Pattern Recognition*, 2021. 3

[90] C. Wick, Jochen Zöllner, and Tobias Grüning. Transformer for handwritten text recognition using bidirectional post-decoding. *ICDAR*, 2021. 3, 4

[91] Garrett Wilson and D. Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology*, 2018. 2, 3

[92] Phil Wood. Confirmatory factor analysis for applied research. 2008. 2, 7

[93] Jie Yang, Ruijie Xu, Zhiquan Qi, and Yong Shi. Visual anomaly detection for images: A systematic survey. *Procedia Computer Science*, 199:471–478, 2022. The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy after COVID-19. 7

[94] Mingkun Yang, Minghui Liao, Pu Lu, Jing Wang, Shenggao Zhu, Hualin Luo, Qingzhen Tian, and X. Bai. Reading and writing: Discriminative and generative modeling for self-supervised text recognition. *ACM Multimedia*, 2022. 3

[95] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv: Computation and Language*, 2019. 3

[96] Yaping Zhang, Shuai Nie, Wenju Liu, Xing Xu, Dongxiang Zhang, and Heng Tao Shen. Sequence-To-Sequence Domain Adaptation Network for Robust Text Image Recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2735–2744, Long Beach, CA, USA, 2019. IEEE. 2, 3

[97] Kaiyang Zhou, Ziwei Liu, Y. Qiao, T. Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2, 3

[98] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 2019. 2, 3

On the Generalization of Handwritten Text Recognition Models

Supplementary Material

7. Complementary results

In this section, we provide additional metrics to the Character Error Rate (CER) presented in the main paper. We provide the following information regarding the results: (1) Results from the main table of the paper (Table 3) presented in terms of Word Error Rate (WER) in Table 4 and (2) Dataset source for the best test result obtained for each architecture in Table 5.

We observe in Table 4 that the ID results remain generally consistent, comparable to those reported in the literature. However, the performance in the OOD scenario remains significantly poor. Additionally, the numerical values highlight that the WER, being a stricter metric than the CER, amplifies the performance gap in the OOD scenario. While the average ID to OOD gap was 37.6% in terms of CER, it increases to 60.3% when measured using WER. Table 5 presents the best-performing source domain for each target domain in the OOD scenario. The dominance of the IAM dataset is noticeable, accounting for nearly 60% of cases, followed by Bentham with slightly over 20%. However, as reported in Section 4.3, the choice of source domain has minimal impact on OOD performance, as all results remain poor even when the target domain is known (oracle scenario). Additionally, Tables 6 and 7 present all cross-domain results (both real and synthetic, respectively) for each architecture individually. In Table 6, the in-domain (ID) results are displayed on the main diagonals and highlighted in gray.

Lastly, we compare three state-of-the-art VLMs [22, 48, 59] including TrOCR [46] against the best-reported OOD results in the paper (HTR_{OOD} column) in Table 8. As observed, the zero-shot performance of these models is very low, as they are not originally designed for HTR tasks, with TrOCR being on par with the HTR models on the English datasets. Moreover, there is no established pipeline for effectively applying VLMs to such specific HTR datasets, highlighting the need for further investigation.

8. Hyperparameters

8.1. Architectures implementation

As stated in the main paper, we aimed to follow the implementation closest to the original papers using the available information for those that did not provide code. In all cases, the most significant architectural change occurred in the final prediction layer, where the output vocabulary size was adjusted to match the vocabulary size (94) reported in Section 3 of the main text.

8.2. Data augmentation

We detail the parameters used for data augmentation during training. No transformations are applied during validation or testing, except for padding, which is applied equally across the validation, training, and test splits. All transformations are applied independently with a 50% probability. For the transformations, we utilized those available in version 2 of transformations in torchvision (`torchvision.transforms.v2`). To simplify visualization and shorten the names, we directly referenced the v2 submodule. For operations involving OpenCV, we employed the opencv-python library (`cv2` module) to execute OpenCV transformations directly.

- **Dilation** (`Custom transformation`):
  - Parameters: kernel size = 3; iterations = 1.
- **Erosion** (`Custom transformation`):
  - Parameters: kernel size = 2; iterations = 1.
- **Elastic Transform** (`v2.ElasticTransform`):
  - Parameters: sigma = 5.0; alpha = 5.0; fill = 255 (white).
- **Random Affine (Rotation, Translation, Shear)** (`v2.RandomAffine`):
  - Parameters: rotation degrees = ± 1; translation = 1% horizontally and up to 5% vertically; shear = ± 1 pixels (sheared by a factor of 5); fill = 255 (white).
- **Perspective** (`v2.RandomPerspective`):
  - Parameters: Distortion scale = 0.1; fixed probability of applying the distortion = 100%; fill = 255 (white).
- **Gaussian Blur (Noise)** (`v2.GaussianBlur`):
  - Parameters: kernel size = 3; sigma = 2.0.
- **Padding** (`v2.Pad`):
  - Parameters: padding = 15 pixels on the left and right; fill = 255 (white).
- **Grayscale** (`v2.Grayscale`):
  - Parameters: num_output_channels = 1
- **Convert to Tensor** (`v2.ToTensor`):
  - Converts the input data to a PyTorch tensor format.

Dilation details. The image is first inverted using `cv2.bitwise_not`. Then, `cv2.dilate` is applied with the selected kernel, expanding the white areas in the image. The process is repeated for the specified number of iterations. Finally, the image is inverted again to restore its original colors.

Erosion details. The image is inverted using `cv2.bitwise_not`, followed by `cv2.erode` with the selected kernel, shrinking the white areas. This operation is also repeated for the given number of iterations.

Table 4. In-distribution (ID) and out-of-distribution (OOD) results (WER %) for HTR models across datasets. The OOD result is reported from the best-performing source. Results marked with $*$ indicate outliers, meaning that the model did not converge in the ID setting. Average results (bottom row) are computed filtering out outliers. † denotes architectures implemented from the papers (no code provided).

Dataset	CRNN [64]		VAN [20]		C-SAN† [26]		HTR-VT [47]		Kang† [39]		Michael† [54]		LT† [10]		VLT† [11]	
	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD
IAM	22.4	68.2	24.2	76.8	50.7	83.6	18.2	83.7	23.2	87.1	20.2	82.9	23.4	72.0	27.0	70.3
Rimes	11.5	74.2	18.2	66.9	45.1	80.7	24.5	78.7	14.8	78.3	20.0	84.5	13.2	77.4	13.3	76.1
G.W.	26.0	68.8	31.2	74.7	33.8	93.9	71.7*	83.3	104.3*	77.6	80.1*	76.1	195.4*	65.7	51.4*	65.8
Bentham	13.2	54.9	20.5	62.1	30.2	82.2	24.6	70.8	19.8	68.2	19.8	69.6	14.3	60.6	15.5	60.5
S.G.	31.1	96.8	33.1	95.3	41.0	96.4	59.5	97.4	203.3*	98.7	134.7*	98.8	45.0	97.9	37.6	97.3
Rodrigo	8.3	92.1	12.4	92.9	22.3	97.2	16.3	92.2	11.1	96.8	15.4	96.2	9.6	94.0	10.7	94.3
ICFHR_{2016}	22.1	104.5	33.9	100.4	64.2	100.7	43.3	103.1	31.7	106.8	33.9	104.8	24.2	110.3	24.6	109.7
Average	**19.2**	**79.9**	24.8	81.3	37.2	90.7	25.4	87.0	20.1	87.6	22.3	87.6	21.6	82.6	25.7	82.0

Table 5. Best source domain for each target (rows) across all architectures studied in the paper.

Dataset	CRNN [64]	VAN [20]	C-SAN† [26]	HTR-VT [47]	Kang† [39]	Michael† [54]	LT† [10]	VLT† [11]
IAM	Bentham	Rimes	Rimes	Rimes	Rimes	Bentham	Bentham	Bentham
G.W.	IAM	Bentham	IAM	Bentham	IAM	IAM	IAM	IAM
Bentham	IAM	IAM	IAM	IAM	IAM	IAM	IAM	IAM
Rimes	IAM	IAM	IAM	IAM	IAM	IAM	IAM	IAM
S.G.	Rodrigo	IAM	Rodrigo	Rodrigo	IAM	IAM	Rodrigo	Rodrigo
Rodrigo	IAM	IAM	Bentham	IAM	IAM	IAM	IAM	IAM
ICFHR 2016	Bentham	Bentham	Bentham	Bentham	Rimes	IAM	Rimes	Bentham

Afterward, the image is inverted back to its original color scheme.

9. Visual and textual divergences

In this section, we present the specific numerical metrics for visual and textual divergence across the various domains used in the factor analysis. Prior to presenting these results, we first describe the training procedure for the Convolutional Autoencoder (AE) (ϕ_{θ_S}) employed to measure reconstruction error (visual divergence).

9.1. Convolutional Autoencoder

9.1.1. Architecture

Regarding the Autoencoder (AE) used, we employed a rather simple convolutional architecture. The encoder progressively downsamples and compresses the input image into a 512-dimensional latent vector using four 3×3 convolutional layers, each followed by leaky ReLU activation and 2×2 max-pooling. The feature channels increase sequentially from 1 to $16, 32, 64$, and 128, with a fully connected layer producing the final latent representation. The decoder reconstructs the image from the latent vector by reversing the encoder's process. It uses a fully connected layer to reshape the latent vector into a tensor, followed by four transposed convolutional layers that upsample the feature map to the original image size. Feature channels decrease from 128 to $64, 32, 16$, and finally 1, with leaky ReLU activations applied after each layer, except the final layer, which uses a sigmoid activation to normalize output pixel values.

Despite the simplicity of the architecture, the input images are rescaled to dimensions of 64 pixels in height and 1024 pixels in width, result in a model with approximately 33 million parameters. Note that due to the large image size, the pre-flattened vector resulting from the encoder's downsampling has 32,768 dimensions (flattening the final feature map of the encoder with 128 channels, a height of 4, and a width of 64). Using an MLP to reduce this vec-

Table 6. Complete CER results in all datasets using real data.

Method	S/T	IAM	Rimes	G.W.	Bentham	S.G.	Rodrigo	ICFHR_{2016}
CRNN [64]	IAM	6.4	25.0	31.1	25.3	45.5	40.9	86.2
	Rimes	35.4	3.7	49.0	50.2	52.3	47.1	87.9
	G.W.	55.6	61.5	8.2	59.2	69.3	66.2	100.0
	Bentham	34.9	45.3	32.2	4.7	57.8	43.8	78.7
	S.G.	77.7	74.5	89.3	78.0	7.2	52.8	100.0
	Rodrigo	65.7	61.4	71.8	66.3	33.6	1.7	85.3
	ICFHR_{2016}	74.9	78.4	81.6	75.4	77.9	75.6	5.2
VAN [20]	IAM	6.6	21.3	34.5	26.6	39.8	38.5	82.9
	Rimes	28.6	5.6	46.1	45.0	47.2	43.7	88.4
	G.W.	73.7	67.4	9.3	59.3	67.1	69.6	100.0
	Bentham	37.2	41.7	32.0	7.4	49.4	38.6	75.3
	S.G.	96.1	85.0	93.1	83.1	7.8	57.7	100.0
	Rodrigo	76.5	70.7	78.2	68.1	41.1	2.3	87.3
	ICFHR_{2016}	70.8	74.8	76.4	67.8	72.1	71.4	7.5
C-SAN [26]	IAM	28.6	29.8	49.8	38.9	50.2	46.6	90.9
	Rimes	31.5	21.3	50.7	45.9	51.1	45.7	87.0
	G.W.	60.7	60.2	32.0	63.2	68.4	66.8	96.9
	Bentham	54.7	51.3	58.3	26.6	64.8	45.2	83.4
	S.G.	75.3	72.5	81.8	75.5	39.8	50.8	90.2
	Rodrigo	72.7	69.1	78.1	68.2	35.0	38.5	86.9
	ICFHR_{2016}	78.4	81.8	86.6	78.7	86.1	81.2	75.3
HTR-VT [47]	IAM	5.8	28.3	40.0	33.3	44.2	38.5	86.1
	Rimes	33.7	7.9	46.2	48.2	51.0	46.1	81.9
	G.W.	70.1	73.3	34.9	76.3	79.2	76.9	85.9
	Bentham	44.4	49.8	38.6	8.4	58.1	45.4	79.6
	S.G.	78.7	78.2	89.8	78.1	17.1	60.1	100.0
	Rodrigo	66.4	64.1	68.7	67.8	36.5	3.9	85.4
	ICFHR_{2016}	74.9	77.1	78.8	73.3	76.2	71.9	11.6
Kang [39]	IAM	8.0	32.0	44.0	39.4	51.8	60.6	96.2
	Rimes	42.1	5.7	65.5	63.5	62.1	65.2	92.6
	G.W.	82.8	81.8	78.4	77.7	78.3	77.4	100.0
	Bentham	53.4	59.7	46.6	8.5	80.8	71.1	95.5
	S.G.	100.0	100.0	100.0	100.0	78.7	86.5	100.0
	Rodrigo	81.7	78.6	85.9	78.6	61.1	2.6	95.9
	ICFHR_{2016}	74.7	76.5	75.3	74.1	75.9	74.9	7.8
Michael [54]	IAM	7.5	35.5	43.6	43.5	55.3	65.3	85.2
	Rimes	54.5	6.9	63.9	70.2	64.3	66.8	86.3
	G.W.	78.9	80.5	53.8	73.4	79.5	76.7	100.0
	Bentham	49.1	60.0	50.7	8.5	74.5	70.7	91.2
	S.G.	100.0	100.0	100.0	100.0	76.9	79.5	100.0
	Rodrigo	100.0	100.0	87.2	100.0	92.9	3.8	92.4
	ICFHR_{2016}	89.3	89.7	81.6	80.4	77.9	77.9	9.5
LT [10]	IAM	7.9	30.8	32.3	33.8	51.4	48.4	98.1
	Rimes	42.9	5.0	52.5	61.9	60.8	56.4	90.5
	G.W.	83.5	85.8	79.6	87.9	75.0	73.3	100.0
	Bentham	42.0	55.4	39.6	6.0	65.4	51.7	92.4
	S.G.	86.6	82.1	95.6	84.9	12.5	65.9	96.7
	Rodrigo	73.1	67.6	80.9	70.8	37.8	2.0	90.7
	ICFHR_{2016}	78.0	81.3	81.3	77.9	77.4	76.5	5.9
VLT [11]	IAM	8.9	29.4	32.1	33.3	48.2	47.4	89.7
	Rimes	44.9	5.1	56.4	63.2	62.4	59.1	96.4
	G.W.	69.5	72.2	25.2	69.6	76.3	75.9	100.0
	Bentham	41.3	53.7	43.7	6.1	65.8	53.5	85.1
	S.G.	91.8	83.3	98.4	86.9	9.2	58.9	100.0
	Rodrigo	71.7	67.2	80.2	72.0	38.7	2.2	90.7
	ICFHR_{2016}	78.3	80.3	80.8	81.7	79.5	81.8	6.0

Table 7. Complete CER results in all datasets using synthetic data.

Method	S/T	IAM	Rimes	G.W.	Bentham	S.G.	Rodrigo	ICFHR_{2016}
CRNN [64]	WIT-en	11.9	22.3	16.9	26.9	25.8	28.1	78.5
	WIT-fr	19.5	17.0	26.3	31.7	26.3	27.2	78.1
	WIT-es	20.0	22.7	27.4	32.9	27.2	22.2	79.4
	WIT-la	20.7	24.6	27.4	34.5	21.4	27.4	79.0
	WIT-de	20.4	25.3	27.3	32.1	28.8	28.3	77.6
VAN [20]	WIT-en	16.9	24.3	25.8	26.1	26.3	28.4	75.0
	WIT-fr	22.4	19.4	31.7	33.1	25.3	25.8	76.9
	WIT-es	23.1	23.5	32.6	34.8	26.0	23.2	77.5
	WIT-la	22.7	24.5	34.1	34.8	23.5	28.2	77.7
	WIT-de	21.9	26.0	30.8	33.8	28.2	29.6	74.3
C-SAN [26]	WIT-en	32.0	38.0	47.8	42.7	35.7	38.8	83.5
	WIT-fr	35.7	35.9	47.4	46.5	36.4	40.4	81.5
	WIT-es	36.2	38.0	46.4	47.4	35.0	37.3	82.7
	WIT-la	36.6	38.8	49.6	48.0	35.8	40.7	83.4
	WIT-de	34.8	38.6	48.9	46.0	36.6	40.5	81.1
HTR-VT [47]	WIT-en	20.7	31.5	26.3	28.3	29.4	32.2	77.5
	WIT-fr	27.5	26.6	34.2	38.1	29.3	32.1	78.1
	WIT-es	28.2	30.3	35.2	38.6	30.1	26.6	77.3
	WIT-la	28.9	33.1	35.4	39.6	27.8	30.6	78.2
	WIT-de	29.2	34.1	36.6	39.2	33.0	34.2	76.7
Kang [39]	WIT-en	28.7	44.5	45.1	51.3	40.0	46.1	85.3
	WIT-fr	38.6	33.1	47.7	57.4	36.2	41.9	87.8
	WIT-es	37.7	49.1	70.3	66.4	42.8	48.7	100.0
	WIT-la	26.2	29.6	39.6	41.2	22.7	35.3	95.3
	WIT-de	32.3	40.1	42.0	43.9	32.4	43.0	84.2
Michael [54]	WIT-en	20.6	34.0	30.5	36.6	35.2	43.4	83.6
	WIT-fr	32.9	25.2	42.8	52.4	36.8	41.1	83.5
	WIT-es	33.8	33.9	45.0	54.0	36.6	33.4	85.2
	WIT-la	37.7	39.5	47.3	58.5	30.7	45.6	82.9
	WIT-de	39.1	44.1	48.4	61.7	40.3	49.7	79.8
LT [10]	WIT-en	13.8	25.1	16.3	21.4	23.4	28.0	80.6
	WIT-fr	22.5	18.9	26.2	35.5	24.1	27.7	79.8
	WIT-es	23.3	25.0	27.1	38.0	22.5	24.8	81.4
	WIT-la	24.0	26.4	27.9	39.3	22.5	28.5	83.0
	WIT-de	23.1	27.6	24.9	36.3	26.0	28.2	78.8
VLT [11]	WIT-en	15.3	26.7	19.0	25.0	23.5	29.1	80.6
	WIT-fr	23.0	19.5	25.9	37.3	23.3	27.9	79.5
	WIT-es	24.9	27.0	28.4	40.6	25.1	24.2	79.8
	WIT-la	26.0	28.3	29.9	42.0	26.6	31.0	82.1
	WIT-de	23.6	25.9	27.0	39.2	24.4	28.2	79.2

Table 8. Zero-shot performance (CER) of VLMs on HTR datasets vs. the best-reported OOD results in the paper (HTR_{OOD} column).

Dataset	LLaVA1.6	Kosmos-2	TrOCR_{M}	InstructBlip	HTR_{OOD}
IAM	74.9	80.3	6.8	78.9	28.6
Rimes	93.5	81.5	27.2	80.4	21.3
G.W.	78.6	79.7	17.3	83.3	31.1
S.G.	80.4	82.5	44.1	87.5	25.3
Bentham	85.4	78.4	17.9	76.7	33.6
Rodrigo	76.4	81.2	38.1	86.2	38.5
ICFHR_{2016}	95.3	87.2	92.6	88.7	75.3

tor to 512 dimensions significantly increases the parameter count. These two layers (one in the encoder and one in the decoder) account for 99% of the model's parameters.

9.1.2. Training details

We train the AE to minimize the Mean Squared Error (MSE) between the input and reconstructed images. We employ the Adam optimizer with a learning rate of 0.001 for a maximum of 100 epochs. To avoid overfitting, we save the best-performing model according to the validation loss of the same source domain at the end of each epoch.

9.2. Visual divergence

This section presents the results of visual domain divergence, measured by the reconstruction error obtained from the autoencoder described in previous sections. Fig. 8 illustrates the divergence (calculated as the average MSE per image) between each domain pair, with the source represented on the Y-axis and target on the X-axis. Divergences are computed between training and test splits for each pair. To facilitate interpretability, the values are normalized, such that a value of 100 reflects high divergence (darker colors), while a value of 0 denotes indicating low divergence (lighter colors). To validate the visual divergence results in the OOD scenario, Fig. 11 presents images from three pairs of domains with low visual divergence (left) and three domains with high one (right) with their respective scores. Note that the left column features writing styles with very similar stroke densities, while the right column displays styles that differ significantly in both stroke appearance and density. The domain pairs were selected based on the scores presented in Fig. 8, ensuring minimal repetition of domains to better highlight the differences.

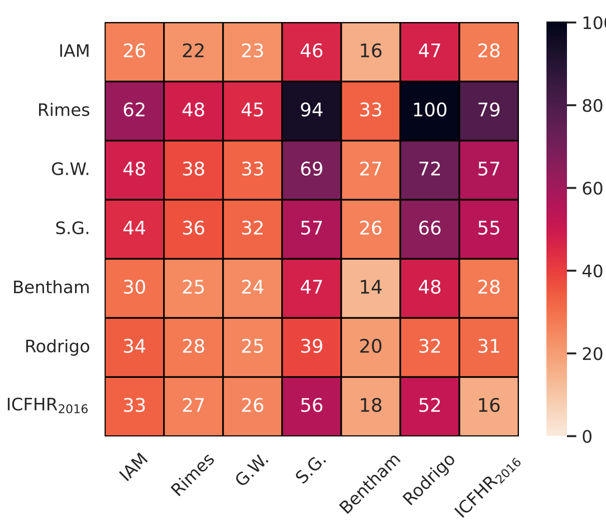

Figure 8. Heatmap of visual divergence between source (rows) and target (columns) from real HTR domains. Divergence values are normalized, with higher scores indicating greater divergence and lower scores reflecting lower divergence.

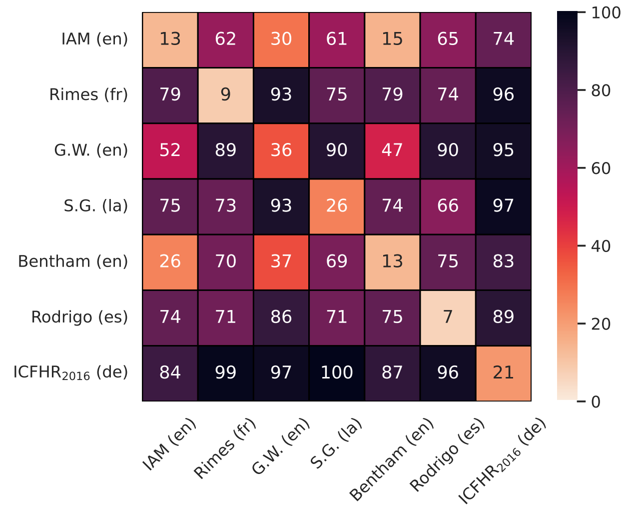

Figure 9. Heatmap of textual divergence from real HTR domains. Rows correspond to source domains, while columns represent target domains. The values are normalized, with 100 indicating maximum divergence and 0 representing minimum divergence.

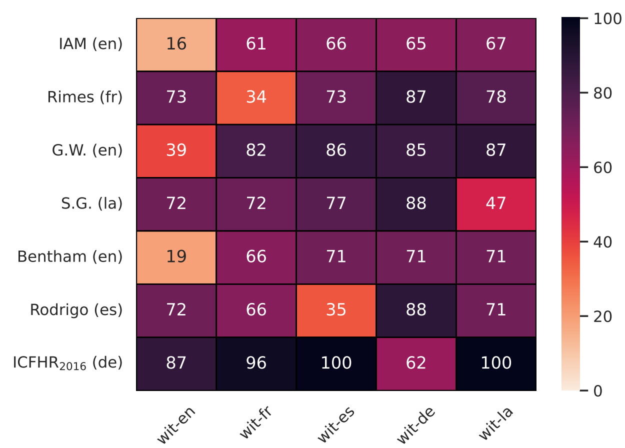

Figure 10. Heatmap of textual divergence between real and synthetic domains. The values are normalized, where 100 represents maximum divergence and 0 represents minimum divergence. Note that each source domain corresponds to a target domain that matches its language, except for English, where three target domains are used: IAM, George Washington, and Bentham.

9.3. Textual divergence

We present the results of textual domain divergences, quantified as the averaged KL-divergence across n-grams as described in the main text. Fig. 9 shows the divergence between textual distributions across domains, with the source represented on the Y-axis and the target the X-axis. Divergences are computed between training and test splits for each pair. Fig. 10 presents the textual divergences be-

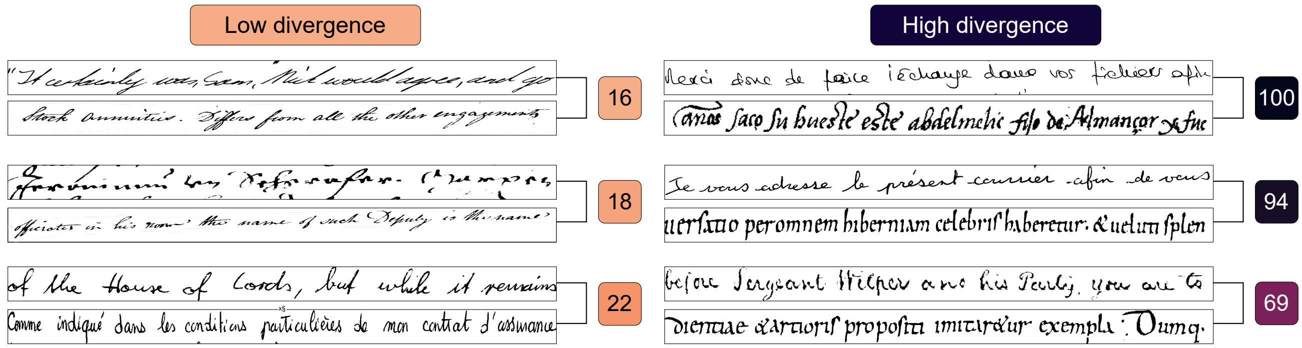

Figure 11. Representation of visual divergence between domains. Examples are presented in two columns: on the left, three domain pairs with low visual divergence, and on the right, three pairs with high visual divergence. For each domain pair, the source domain is shown at the top and the target domain at the bottom. Left (top to bottom): First pair (IAM-Bentham), second pair (ICFHR$_{2016}$-Bentham), third pair (IAM-Rimes). Right (top to bottom): First pair (Rimes-Rodrigo), second pair (Rimes-S.G.), third pair (G.W.-S.G.). The divergence percentage (see Fig. 8) is displayed for each pair.

tween real source domains (Y-axis) and synthetic domains (X-axis) for each language. In this case, the divergence is calculated between the training split of the source domain and the training split of the synthetic data, as these are used to compute the n-grams and train the models in the synthetic experiments. Both figures display normalized values, where a value of 100 indicates maximum divergence (darker colors) and 0 minimum divergence between texts.

10. Factor analysis

The selection of the number of factors is a crucial criterion for analyzing the outcomes of the factor analysis. Note that the first k factors span the subspace defined by the first k eigenvectors of the data matrix. To determine the number of factors (n), the simplest rule of thumb involves retaining all eigenvectors with eigenvalues ≥ 1. This can be simply visualized by plotting the eigenvalues in descending order using a scree plot, as shown in Fig. 12. Based on this analysis, we decided to retain four factors as stated in the main text of the paper.

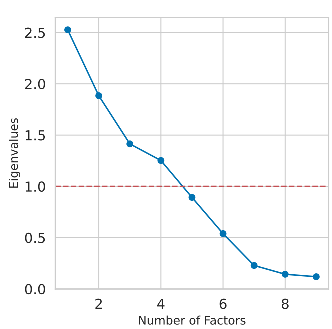

Figure 12. Scree plot: Eigenvalues of the standardized values used for factor analysis, ordered in descending magnitude. We chose to retain 4 factors, as these are the ones with eigenvalues ≥ 1.