Supplementary Material: 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

Appendix A

Word-Level Data Collection Process: To enable semi-supervised learning for handwritten text recognition at word granularity, we constructed a large-scale dataset through an automated and scalable pipeline, illustrated in Fig. 1. The process begins by scraping a diverse collection of handwritten document images from various public sources on the internet, such as scanned notebooks, academic worksheets, and archival forms. This variety ensures broad coverage of writing styles, layouts, and background conditions.

The raw images are processed using the `AnalyzeDocument` API from AWS Textract, a commercial OCR engine designed to extract structured textual content from documents. Specifically, we use the feature to obtain bounding boxes and transcriptions for detected word-level entities. The API outputs are returned in JSON format, preserving spatial metadata (bounding boxes, geometry) alongside text content and confidence scores for each detected word.

Following extraction, we perform a filtering step to construct an unlabeled pool with reliable structural quality. OCR-detected words with extremely low confidence or missing transcriptions are discarded. For the remaining samples, we retain the image crops corresponding to the bounding boxes without associating the raw transcriptions as labels. This process yields approximately 7.92 million high-resolution word images, which serve as the foundation for our self-supervised and weakly-supervised objectives.

To construct a labeled subset from this pool, we randomly sample a representative portion of the filtered data and subject it to manual verification. Annotators cross-check each transcription against the word image and correct any misrecognized outputs. In some cases, bounding box adjustments are also applied to improve cropping accuracy. This manually curated subset results in a labeled dataset of 2.08 million word images with verified ground-truth annotations. Together, the large-scale unlabeled pool and high-quality labeled set form a hybrid training resource tailored for our LoGo-HTR framework, facilitating effective learning under limited supervision.

To facilitate downstream supervision and error analysis, we visualize word-level outputs generated by AWS Textract, which serves as the detection and recognition engine in our backend pipeline. As shown in Fig. 2, the predicted word regions are tightly bounded, effectively segmenting individual tokens across diverse handwriting styles. This visualization provides qualitative insight into the spatial and lexical fidelity of the Textract engine, which we leverage to construct high-quality training and evaluation datasets. Accurate word-level segmentation is essential for structured handwritten document understanding and forms the foundation of our semi-supervised learning framework.

Appendix B

Loss Formulation. Let $x^{(1)}, x^{(2)} \in \mathbb{R}^{B\times C\times H\times W}$ be two augmented views. A patch size $P = \lfloor H/K \rfloor$ with stride $S = \lfloor (1-\text{overlap})P \rfloor$ yields $N = \lfloor H/S \rfloor \times \lfloor W/S \rfloor$ patches.

To extract local features from spatial patches, we perform average pooling on each corresponding region of the feature maps from both augmented views. Given the n-th patch of image b, the patch-level embeddings are computed as:

$$z_{b,n}^{(1)} = \text{AvgPool}\Big(x_b^{(1)}[\text{patch } n]\Big)\,, \tag{1}$$

$$z_{b,n}^{(2)} = \text{AvgPool}\Big(x_b^{(2)}[\text{patch } n]\Big)\,, \tag{2}$$

where $x_b^{(1)}$ and $x_b^{(2)}$ are the feature maps from two augmentations of the same image. This operation yields two feature vectors corresponding to the same spatial location across views.

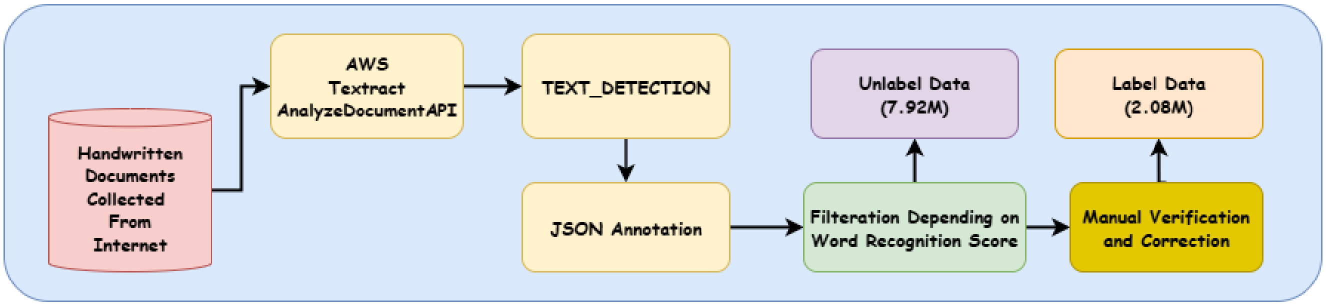
Handwritten Documents Collected From Internet
AWS Textract AnalyzeDocumentAPI
TEXT_DETECTION
JSON Annotation
Filteration Depending on Word Recognition Score
Unlabel Data (7.92M)
Manual Verification and Correction
Label Data (2.08M)

Figure 1. Word-level data collection pipeline. Starting from raw handwritten documents sourced from public internet repositories, we employ the AWS Textract AnalyzeDocument API to detect and annotate word-level text regions. The resulting JSON outputs are filtered based on OCR confidence to retain high-quality word instances, yielding 7.92 million unlabeled samples. A selected portion of these is manually verified and corrected to generate a high-quality labeled set of 2.08 million word-level instances.

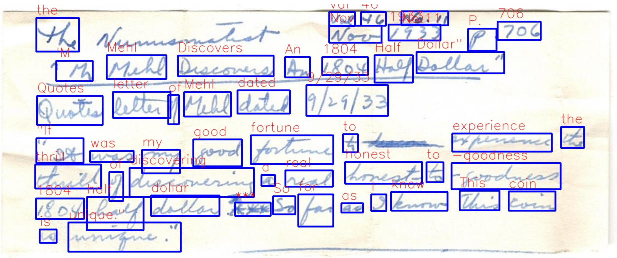
the
Nov
46
P.
706
'M
Mehl
Discovers
An
1804
Half
Dollar''
Quotes
letter
of Mehl
dated
9/29/33
''If
was
my
good
fortune
to
experience
the
thrill
of
discovering
a
real
honest
to
-goodness
1804
half
dollar
***
So
for
as
I
know
This
coin
is
unique.''

Figure 2. Word-level recognition using AWS Textract. The bounding boxes and transcriptions are generated automatically and used in our backend data pipeline.

Local Contrastive Loss:

$$s^{\text{pos}}_{b,n} = \frac{\langle z^{(1)}_{b,n},\, z^{(2)}_{b,n}\rangle}{\|z^{(1)}_{b,n}\|_2\;\|z^{(2)}_{b,n}\|_2}, \tag{3}$$

$$S^{(v)}_b = Z^{(v)}_b\big(Z^{(v)}_b\big)^{\top},\; \tilde{S}^{(v)}_b = S^{(v)}_b - \operatorname{diag}\!\big(S^{(v)}_b\big), \tag{4}$$

$$\ell_{b,n} = \Big[\frac{s^{\text{pos}}_{b,n}}{\tau}\;\Big|\;\frac{\tilde{S}^{(1)}_{b,n,:}}{\tau}\;\Big|\;\frac{\tilde{S}^{(2)}_{b,n,:}}{\tau}\Big], \tag{5}$$

$$L_{\text{local}} = \frac{1}{BN}\sum_{b=1}^{B}\sum_{n=1}^{N}\operatorname{CE}\!\big(\ell_{b,n},0\big). \tag{6}$$

Global Decorrelation Loss: After an MLP head we obtain $y^{(1)}, y^{(2)} \in \mathbb{R}^{B\times d}$:

$$C_{ij} = \frac{\sum_{b=1}^{B} y^{(1)}_{b,i}\, y^{(2)}_{b,j}}{\sqrt{\sum_b \big(y^{(1)}_{b,i}\big)^2}\;\sqrt{\sum_b \big(y^{(2)}_{b,j}\big)^2}}, \tag{7}$$

$$L_{\text{global}} = \sum_i \big(1 - C_{ii}\big)^2 + \lambda \sum_i \sum_{j\neq i} C_{ij}^2. \tag{8}$$

Total Loss:

$$L_{\text{total}} = \alpha\, L_{\text{global}} + \beta\, L_{\text{local}}, \qquad (\alpha = 0.4,\; \beta = 0.6).$$

Theoretical Insights — Global vs. Local Objectives: The global decorrelation loss [18], inspired by the Barlow Twins framework, encourages representations from two augmented views to be invariant by maximizing the similarity between their corresponding feature dimensions. This is achieved by aligning their cross-correlation matrix to the identity, thereby ensuring that each feature dimension encodes unique, non-redundant information. Although categorized as a non-contrastive method, it implicitly aligns positive pairs (augmented versions of the same input) through correlation matching, sharing conceptual goals with contrastive learning approaches such as SimCLR [3] or MoCo [7].

While the global decorrelation loss promotes semantic robustness and invariance, it operates on the level of globally pooled features, potentially overlooking fine-grained spatial information. This limitation is especially significant in domains like handwritten text recognition, where localized spatial patterns carry critical structural and semantic cues.

To address this, we introduce a local contrastive loss that operates at the level of spatial patches. Specifically, it enforces consistency between corresponding regions in the two augmented views by contrasting each local patch with its counterpart while treating non-corresponding patches as negatives. This promotes structural alignment and positional awareness, thereby preserving the spatial integrity of representations. The local objective ensures that features not only encode semantics but also retain spatial discriminability, which is essential for tasks involving structured visual input such as handwritten documents.

Our formulation draws theoretical support from prior work on regularizing global learning objectives. For instance, VICReg [1] and W-MSE [4] highlight the importance of introducing constraints such as variance preservation and covariance regularization to prevent representational collapse. Additionally, Jing *et al.* [12] provide a theoretical analysis showing that spatial or feature-level constraints can mitigate over-smoothing and improve generalization.

In our case, the local contrastive loss acts as a regularizer on top of the global decorrelation objective. While the global term encourages semantic abstraction, the local term anchors features to meaningful spatial locations. This synergy results in representations that are both semantically rich and spatially precise—properties that are particularly crucial for handwritten visual data, where spatial context (e.g., alignment, stroke placement) and fine-grained details (e.g., loops, diacritics) influence recognition performance.

The global loss improves representation invariance and reduces redundancy, while the local loss preserves structural details by enforcing region-level consistency. Their combination leads to a well-regularized feature space with enhanced generalization, spatial fidelity, and semantic robustness.

Appendix C

Implementation Details: We utilize a DenseNet-based encoder with three blocks of 16 bottleneck layers, a growth rate of $k = 24$, dropout probability $p = 0.2$, and compression factor $\theta = 0.5$ in the transition layers. During the proposed SSL-based pretraining, each image is divided into 5×5 patches after passing through the first convolution layer of the encoder. All the patches are then passed through an average pooling layer. After that, the local contrastive loss is calculated using those patches, which act as a local loss. To calculate the global loss, a projection head with dimensions $1024 \rightarrow 512$ is applied to the end of the DenseNet encoder, resulting in embeddings of shape $(N, 512)$, where N is the number of batches. The image augmentations include scale jittering within $[0.7, 1.4]$ and color jittering with brightness and contrast values set to 0.25, saturation to 0.2, and hue to 0.2, all applied with probability $p = 0.5$.

The model is trained using stochastic gradient descent (SGD) with a learning rate of 10^{-3}, weight decay of 10^{-6}, and a batch size of 128. The total loss is a weighted combination of global and local objectives, with weights of 0.4 and 0.6, respectively.

In the downstream stage, we follow the architecture of CoMER [19], incorporating the pre-trained DenseNet encoder and the trained attention module with a Transformer-based decoder. The decoder consists of 3 layers, each with a model dimension of 256, a feed-forward network dimension of 1024, a dropout rate of 0.3, and 8 attention heads. A coverage attention mechanism is also integrated to refine alignment during decoding. All the experiments have been done on 2 RTX A6000 GPUs, each having a VRAM of 48 GB.

Appendix D

Evaluation Metrics: **Word Recognition Rate (WRR)** is defined as the percentage of word-level predictions that exactly match the ground truth:

$$\text{WRR} = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100, \tag{9}$$

where N_{correct} is the number of correctly predicted words and N_{total} is the total number of word samples.

Character Recognition Rate (CRR) measures the percentage of correctly predicted characters over all characters in the dataset:

$$\text{CRR} = \frac{C_{\text{correct}}}{C_{\text{total}}} \times 100. \tag{10}$$

Character Error Rate (CER) quantifies the normalized edit distance between predicted and ground truth sequences at the character level [5]:

$$\text{CER} = \frac{\sum \text{EditDist}_{\text{char}}(\hat{y}, y)}{\sum |y|} \times 100, \tag{11}$$

where $\hat{y}$ is the predicted character sequence and y is the ground truth.

Word Error Rate (WER) follows a similar formulation but operates at the word level. It is computed as:

$$\text{WER} = \frac{\sum \text{EditDist}_{\text{word}}(\hat{w}, w)}{N_{\text{total}}} \times 100, \tag{12}$$

where $\hat{w}$ and w denote predicted and ground truth word sequences, respectively. A non-zero word-level edit distance counts as wrong recognition.

For CER and WER computation, we use the standard Levenshtein [15] distance (edit distance) via the `editdistance` library. All evaluations are conducted on the test splits of IAM, GNHK, and HTRx datasets, using the pretrained model with approximate joint decoding. The metric suite ensures both holistic and fine-grained assessment of model performance, covering strict match accuracy as well as error trends at the character level.

Appendix E

Loss Dynamics and Patch-Level Behavior in SSL Pretraining: To further understand the optimization dynamics of our proposed **LoGo-HTR** framework, we analyze the training loss progression under two configurations: (i) using only the global decorrelation loss (L_{global}), and (ii) combining local patch-wise contrastive loss with global decorrelation ($L_{\text{local}} + L_{\text{global}}$). Figure 5 in the main paper, captures the evolution of the loss over 11k training iterations.

We observe that the combined loss (in **blue**) converges faster and exhibits significantly lower variance compared to the global-only variant (in **red**), which remains high and oscillatory. This behavior indicates the stabilizing effect of local contrastive learning. The local patch-wise objective contributes by anchoring spatially consistent features across augmentations, allowing the model to resolve ambiguities present in full-image alignment, especially in complex handwritten samples.

Moreover, the local loss implicitly acts as a regularizer—by enforcing fine-grained correspondence, it prevents representational collapse and encourages the encoder to learn more structured, disentangled features. This aligns with our t-SNE and Q–Q visualizations (*Figs. 9–11 in Appendix F*), where the model with local loss exhibits improved cluster separability and better statistical properties.

In essence, the complementary nature of L_{local} and L_{global} creates a training regime where both *semantic invariance* and *spatial specificity* are co-optimized. This synergy leads to faster convergence, smoother optimization, and more robust representations for downstream handwritten text recognition.

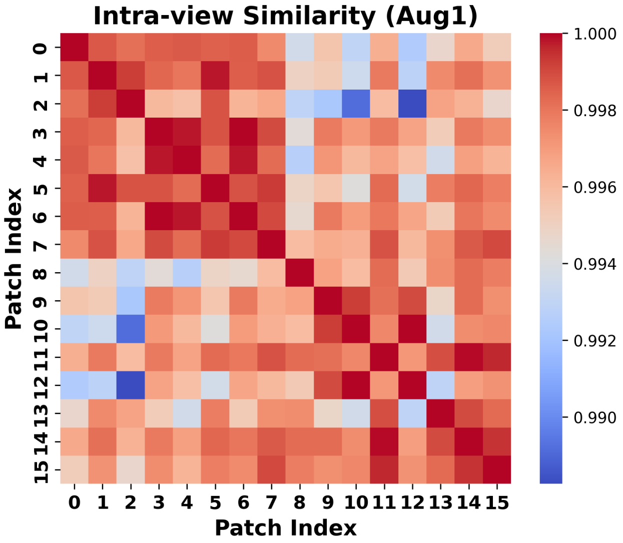

Figure 3. Intra-view similarity matrix for Aug1. Cosine similarities between all patch pairs within the same view. A strong diagonal and structured off-diagonal blocks suggest spatial locality and distinctiveness.

Patch-Level Similarity Analysis Global SSL objectives such as Barlow Twins [18] promote invariance but often overlook local cues crucial for handwriting. Building on insights from SimCLR [3], MoCo [7], PixPro [17], and DINO [2], our local contrastive term aligns corresponding patches while repelling others, preventing representational collapse often observed in purely global methods (cf. BYOL [6] and VICReg [1]).

In Fig. 3, we observe a distinct diagonal pattern—indicating high self-similarity for each patch—and cooler off-diagonal bands that reflect intra-image dissimilarity. These cooler cells, arising naturally, serve as *hard negatives* in our contrastive formulation, sharpening spatial discrimination. This supports the hypothesis that patch-wise self-supervision enhances local structure encoding, consistent with the locality principles outlined in [17].

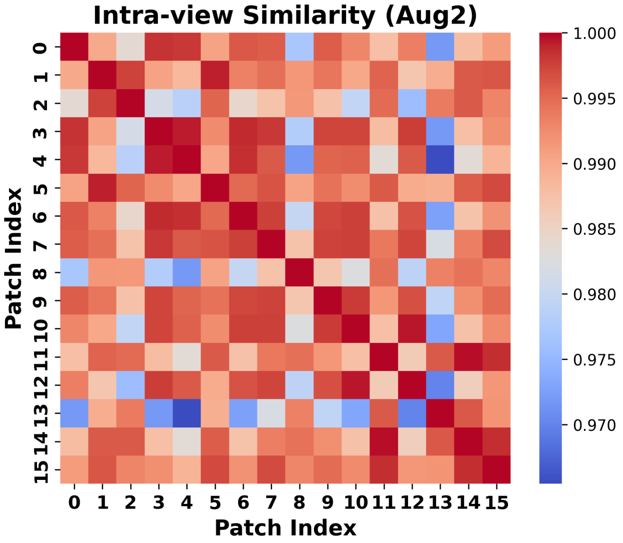

Figure 4. Intra-view similarity matrix for Aug2, showing more diffuse similarities due to stronger augmentation. Despite this, diagonal dominance persists, indicating robustness of local representations.

In Fig. 4, the similarity map is noticeably more diffused, primarily due to aggressive data augmentation (e.g., color jitter, random erase). Despite this, diagonal elements remain dominant, indicating that the encoder maintains alignment of corresponding patches. Importantly, the mean off-diagonal similarity drops from ∼0.992 to ∼0.988, increasing feature diversity.

These observations align with the findings of Li *et al.* [12], where intra-view diversity is exploited to mine effective negatives without large memory banks. This strategy reduces computational overhead while maintaining representational richness — a favorable property in handwritten domains where structural details matter. These intra-view similarity maps reveal that our local contrastive objective not only enforces fine-grained alignment across views but also regularizes the encoder against trivial collapse by encouraging diversity within each view. This results in embeddings that are robust across augmentations yet sensitive to spatial variations — essential for handwritten text recognition.

Patch-level contrastive learning therefore complements global decorrelation by (i) stabilizing optimization, (ii) enforcing stroke-level discrimination, and (iii) yielding a collapse-free, structured latent space — crucial for robust handwritten text recognition.

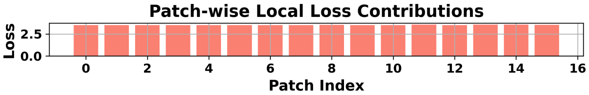

Figure 5. Patch-wise contributions to the local contrastive loss. All patches contribute relatively equally, indicating balanced gradients and uniform attention across spatial regions.

Patch-Level Contrastive Behavior: To understand the spatial behavior of the local contrastive objective, we visualize the per-patch loss contributions (Fig. 5) and positive-pair similarity (Fig. 6) for a single training instance. These values are extracted after forward-passing two augmentations of the same image.

We observe that the local loss is distributed fairly evenly across all patches, with no single patch dominating the overall loss. This balanced distribution reflects stable optimization and suggests that the encoder has learned to represent all regions with similar fidelity. This behavior is crucial in HTR where uniform representation of fine-grained strokes and curves matters for accurate decoding.

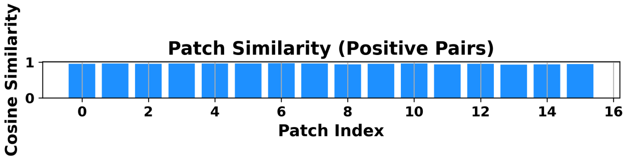

Figure 6. Cosine similarity of positive patch pairs across two augmentations. High and uniform similarity confirms effective local alignment and robustness to perturbations.

In Fig. 6, the cosine similarity between corresponding patches across views is consistently high, validating that the model has successfully learned augmentation-invariant local representations. The low variance in similarity supports the stability of the learned embeddings, especially in the context of contrastive self-supervision.

These findings align with prior work emphasizing spatial uniformity in learned representations [2, 17]. In our case, such behavior is not just beneficial but necessary due to the high-resolution nature of handwritten text and its dependence on spatial consistency.

Negative Similarity Distribution Analysis: To understand the effectiveness of local contrastive learning in our framework, we analyze the intra-view similarity of negative patch pairs under two augmentation regimes. Fig. 7 reveals that with milder augmentation (**Aug1**), most negative similarities cluster near 1.0, indicating weak negatives due to the inherent structural similarity in handwritten text. In contrast, Fig. 8 shows that stronger augmentation (**Aug2**) introduces greater variability, creating harder negatives that improve contrastive discrimination. These trends align with our hypothesis: local contrastive loss is essential to resolve spatial ambiguities in handwriting, and aggressive augmentations amplify its effectiveness. Together, these findings reinforce the importance of both local objectives and augmentation strategies in building robust, spatially-aware representations.

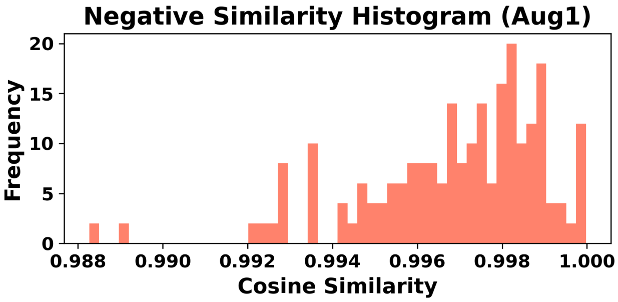

Figure 7. Cosine similarity histogram of intra-view negative patch pairs for **Aug1**. The tight concentration around high similarity (> 0.992) highlights the challenge of discriminating spatially similar patches in handwritten data. This supports the necessity of a local contrastive loss for resolving fine-grained ambiguities.

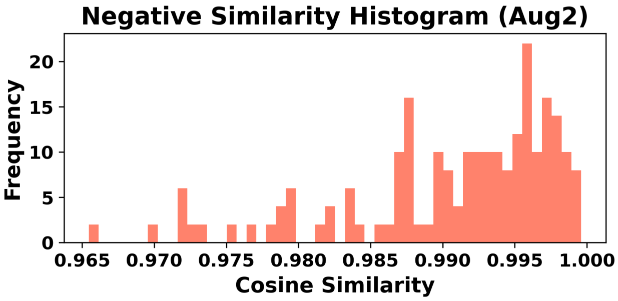

Figure 8. Cosine similarity histogram for **Aug2**. Stronger augmentations lead to a broader similarity distribution (0.965–1.000), injecting harder negatives and enhancing feature diversity. This empirically supports our design choice of aggressive augmentations in patch-level contrastive learning.

Appendix F

Ablation On Projection Head We evaluate the impact of incorporating a projection head [12], showing that it facilitates the learning of richer and more informative representations by reducing redundancy in the embedding space and improving transferability. Fig. 9 presents the t-SNE visualization of the learned embeddings, where the model with the

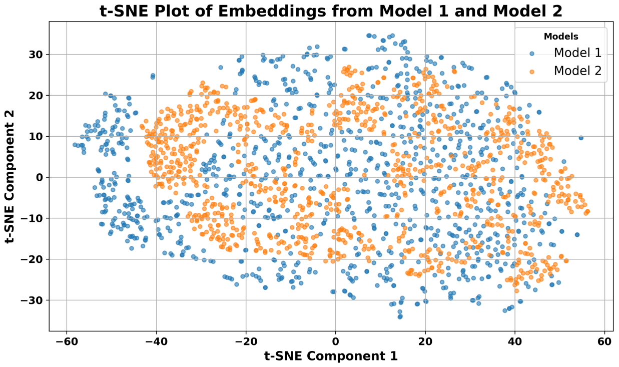

Figure 9. t-SNE plot comparing embeddings from Model 1: **LoGo-HTR** with projection head and Model 2: **LoGo-HTR** without projection head. Model 1 shows better spatial separability and structure.

projection head (Model 1) produces more distinct and well-separated clusters compared to the model without it (Model 2). This indicates that the projection head not only enhances spatial encoding but also improves representational quality by enforcing better alignment of intra-class samples while simultaneously increasing inter-class separability. Moreover, Figs. 10 and 11 further demonstrate that **LoGo-HTR** equipped with the projection head learns more discriminative and semantically meaningful image features for recognition, resulting in sharper boundaries across classes and more robust invariance to handwriting variations. Together, these analyses underscore the importance of the projection head in stabilizing the self-supervised training process and in enabling the encoder to capture richer and more transferable features.

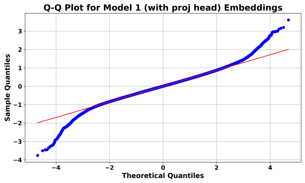

Figure 10. Q–Q plot of Model 1 (with projection head) embeddings shows close alignment with a Gaussian distribution, indicating well-distributed and statistically consistent feature representations.

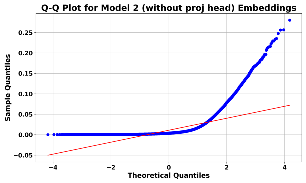

Figure 11. Q–Q plot of Model 2 (without projection head) embeddings shows noticeable deviation from Gaussianity, indicating less statistically consistent and poorly distributed representations.

Dimensional Collapse in Self-Supervised Learning:

Preliminaries: Let $\mathbf{Z} \in \mathbb{R}^{B \times d}$ denote a centred batch of representations output by the encoder, where B is the batch

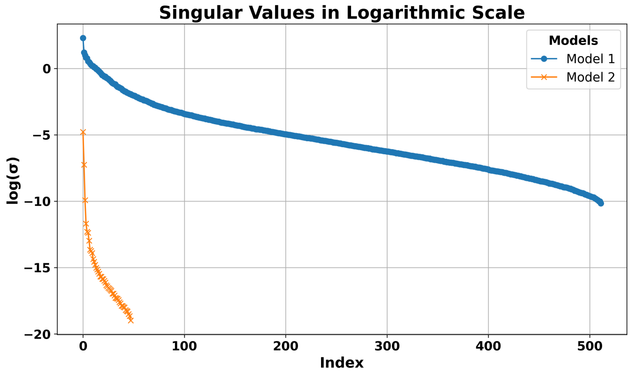

Figure 12. **Log-singular-value spectrum of the representation matrix. Model 1** — our DenseNet encoder followed by a two-layer projection head and the $\mathcal{L}_{\text{global}} + \mathcal{L}_{\text{local}}$ objectives—exhibits a smooth, power-law decay, maintaining large singular values deep into the spectrum. **Model 2** — the same encoder trained *without* a projection head—shows a precipitous drop after σ_{20}, indicating an effective rank collapse. This figure provides empirical evidence that the projection head (together with decorrelation regularization) mitigates dimensional collapse by preserving variance across a substantially larger number of directions.

size and d is the embedding dimension. The singular value decomposition (SVD) is:

$$\begin{aligned} &\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}, \\ &\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_d), \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d \geq 0. \end{aligned} \tag{13}$$

Dimensional collapse arises when a majority of singular values become negligible. The *effective rank* [16] quantifies spectral flatness:

$$\begin{aligned} r_{\text{eff}}(\mathbf{Z}) &= \exp\left(-\sum_{i=1}^{d} p_i \log p_i\right), \\ \text{where} \quad p_i &= \frac{\sigma_i}{\sum_{j=1}^{d} \sigma_j}. \end{aligned} \tag{14}$$

Letting $\mathbf{\Sigma}_z = \frac{1}{B-1}\mathbf{Z}^{\top}\mathbf{Z}$ be the empirical covariance, its rank approximates feature diversity:

$$\operatorname{rank}(\mathbf{\Sigma}_z) = \#\{\sigma_i > 0\}, \quad \det(\mathbf{\Sigma}_z) \to 0 \Rightarrow \text{Collapse}. \tag{15}$$

Dual Self-Supervised Loss — Global + Local: To mitigate dimensional collapse and promote meaningful representation learning, we employ a dual loss:

$$\mathcal{L}_{\text{SSL}} = \alpha\,\mathcal{L}_{\text{global}} + \beta\,\mathcal{L}_{\text{local}}, \tag{16}$$

where α and β are weighting hyperparameters (default: $\alpha = 0.4$, $\beta = 0.6$).

While the global and local terms act directly on the representation space, our design critically relies on a projection head $g_\phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$ inserted after the encoder. This component plays a vital role in regulating the representational structure and preventing collapse. It consists of two linear layers, BatchNorm [11], and a GELU [9] activation.

The projection head serves as a buffer zone between the encoder and the SSL objectives. It absorbs much of the pressure from the decorrelation and contrastive losses, allowing the encoder f_θ to focus on preserving task-relevant features. Without this separation, strong regularisers such as Barlow Twins or patch-level InfoNCE often force the encoder into degenerate regimes—compressing its output to a low-rank or invariant subspace.

Moreover, BatchNorm within the projection head contributes to whitening and variance equalisation across dimensions, both of which have been shown to mitigate collapse [2, 12, 14]. We also observe that the projection head stabilizes optimization dynamics in the early phases of training, effectively acting as a gradient conditioner that mitigates sudden alignment collapse. This stabilization is particularly crucial during the initial epochs, where unstable gradients can otherwise lead to poor convergence. The architectural decoupling provided by the projection head becomes especially important in low-label regimes, where strong supervision signals are scarce or noisy. By introducing a controllable bottleneck with expressive nonlinearity, the head not only enhances representational capacity but also allows the model to disentangle feature richness from invariance constraints. Consequently, the learned embeddings are both more robust and more adaptable to downstream tasks, demonstrating improved generalization even under limited supervision.

Empirically, we find that this architectural choice dramatically improves the singular value spectrum of the encoder outputs. As illustrated in Fig. 12, training without the projection head leads to a sharp drop in the singular value curve—indicating rank deficiency and loss of information diversity. In contrast, with the projection head, the spectrum exhibits a smoother decay, suggesting high-rank, anisotropic representations.

The projection head and dual loss together mitigate global [12] and local [2, 8, 10, 14] representational collapse, yielding robust and highly discriminative features for downstream handwritten text recognition. By jointly addressing both levels of collapse, this combination ensures that the learned embeddings maintain structural diversity while preserving essential invariances. This results in features that are not only more stable during training but also more transferable across different datasets and tasks, improving generalization and overall performance in low-label or challenging scenarios.

Appendix G

Computational Efficiency Analysis: To complement the computational efficiency analysis in the main paper (Fig. 4), we further report the average inference speed of our model on both word-level and line-level benchmarks. The results are summarized in Table 1. On word-level datasets (IAM, GNHK, and RIMES), our model achieves an average inference speed of 0.1046 seconds per word, while on the line-level LAM dataset, it processes a line in just 0.2833 seconds on average. These measurements are taken end-to-end, including feature extraction through the DenseNet encoder and autoregressive decoding by the Transformer-based decoder.

When compared with the model efficiency plot in the main paper(Fig. 4), these results highlight a key advantage of our approach: LoGo-HTR not only provides an excellent trade-off between accuracy (low WER) and computational complexity (GFLOPS) but also demonstrates practical runtime efficiency during inference. While larger models such as TrOCR-LARGE[13] demand significantly higher computational cost and memory, our lightweight 6.4M parameter model achieves competitive or superior recognition performance with far lower decoding latency.

This efficiency is especially critical for deployment in real-world applications such as large-scale digitization projects, mobile OCR systems, and archival preservation, where models must operate on massive volumes of handwritten text with limited computational resources. The strong runtime characteristics of LoGo-HTR reinforce its applicability to such scenarios, ensuring that the benefits of self-supervised pretraining extend beyond accuracy to also include scalability and usability.

Model Name	Word-Level	Line-Level
LoGo-HTR	0.1046 S/Word	0.2833 S/Line

Table 1. Average inference speed of our model across word-level and line-level benchmarks.

Appendix H

Visual Results: Figure 13 presents an expanded set of sample outputs predicted by our **LoGo-HTR** model on the SSL-HWD dataset. The visualization highlights the model's capability to handle diverse handwriting styles, varying from neat block letters to complex cursive scripts.

As observed in the figure, the model maintains high recognition accuracy even in the presence of noise, slant, and irregular spacing. We have highlighted wrongly recognized characters in red to facilitate error analysis. These errors typically occur in cases of extreme ambiguity where character strokes overlap significantly or are indistinguish-

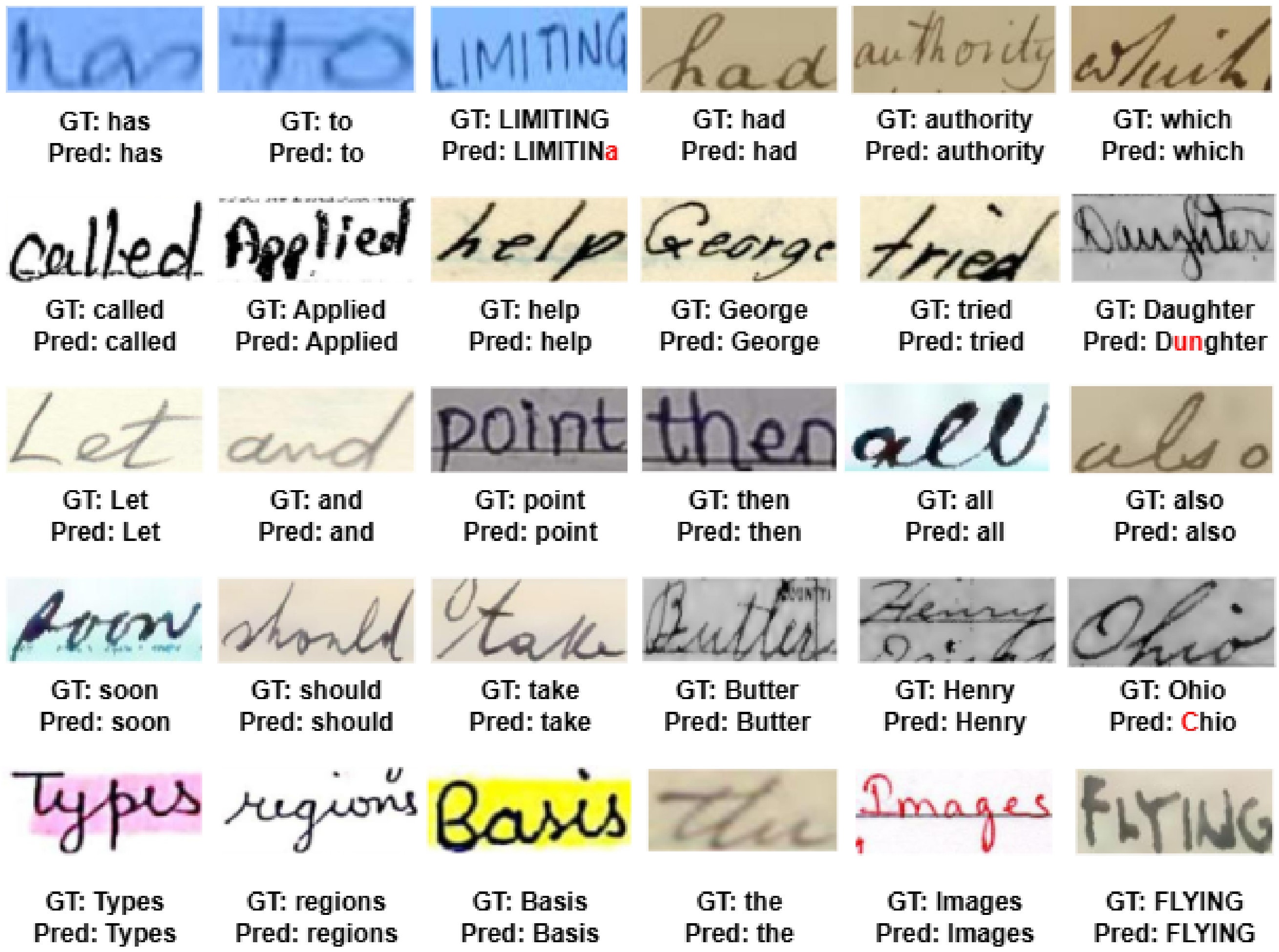

Figure 13. Qualitative visualization of the handwritten word recognition results from our model. Each cell displays the ground truth (GT) and the predicted (Pred) word below the corresponding handwritten image. The model demonstrates a high degree of accuracy in most cases, successfully recognizing diverse handwriting styles and maintaining semantic consistency with the ground truth. Cases such as “LIMITING” vs. “LIMITINa” and “Daughter” vs. “Dunghetr” highlight the model’s sensitivity to fine-grained variations in cursive writing and potential confusion arising from unusual character spacing and slant. Moreover, words with different casing styles, noise levels, and complex pen strokes (e.g., “Butter”, “George”, “FLYING”) are handled reasonably well. These results demonstrate that the model is robust to a variety of visual distortions and maintains recognition quality across uppercase, lowercase, and mixed-case inputs. Overall, this figure showcases the model’s effectiveness and generalization capability across different styles of handwritten text.

able due to writing style (e.g., distinguishing ”u” from ”n” or ”a” from ”o” in careless handwriting). Despite these localized errors, the semantic consistency of the predicted words remains high, validating the effectiveness of the proposed local-global self-supervised learning objectives.

References

[1] Adrien Bardes, Jean Ponce, and Yann Lecun. VICReg: Variance-Invariance-Covariance Regularization For Self-Supervised Learning. In *ICLR 2022 - International Conference on Learning Representations*, Online, United States, 2022. 2, 4

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 4, 5, 7

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 2, 4

[4] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *ICML*, pages 3015–3024, 2021. 2

[5] 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. 3

[6] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPs*, pages 21271–21284, 2020. 4

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 4

[8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 7

[9] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 7

[10] Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *IEEE Trans. on Pattern Anal. Mach. Intell.*, 46(4): 2506–2517, 2024. 7

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 7

[12] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning, 2022. 2, 4, 5, 7

[13] 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. 7

[14] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, pages 1–31, 2024. 7

[15] CA Romein, Achim Rabus, Gundram Leifert, and Phillip Benjamin Ströbel. Assessing advanced handwritten text recognition engines for digitizing historical documents. *IJDH*, pages 1–20, 2025. 3

[16] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *ESPC*, pages 606–610, 2007. 6

[17] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, pages 16684–16693, 2021. 4, 5

[18] 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. 2, 4

[19] Wenqi Zhao and Liangcai Gao. CoMER: Modeling coverage for transformer-based handwritten mathematical expression recognition. In *ECCV*, pages 392–408, 2022. 3