Supplementary Material

This document provides supplementary material for the paper "Text and Style Conditioned GAN for Generation of Offline-Handwriting Lines" submitted to BMVC 2020, including details of the human study described in the paper, additional image results, additional experimental ablation study results, and architectural details for the networks described in the paper. The sections are as follows:

- S.1 Details on FID (and GS) computation.
- S.2 Details on human experiment.
- S.3 Additional generation results.
- S.4 Additional ablation results.
- S.5 Network specifications of each model part.

S.1 Discussion of FID evaluation and GS details

FID [12] is evaluated by passing an image through the convolutional network Inception-v3 and computing statistics on the average pooled features. Inception-v3 was designed to accept images of size 299×299, and thus most implementations of FID rescale images to this size before feeding them to the network. In most situations this is fine since GANs typically generate square images. However, in the case of handwriting, particularly lines, images are generally much wider than they are tall. Resizing them to be square causes significant distortions to the image. Thus, it would make sense to resize images to a height of 299 and maintain the aspect ratio. Since Inception-v3 is fully convolutional up to the average pooling, it can accept variable sized images. We evaluated FID with both the original square resizing and aspect ratio preserving resizing. We found the scores produced when preserving the aspect ratio appeared closest to the FID reported in [2] and [6] and thus assume these authors applied something similar, although they do not report this. We follow [2] in using 25,000 training set images and generate 25,000 images using the same lexicon (words or lines depending on dataset), but styles extracted from the test set. Like [6], we only run the experiment once.

When comparing our generated images to RIMES words, there is a distribution difference caused by segmentation differences. RIMES words are segmented tightly to each word. Our model is trained on RIMES lines, which generally have more whitespace on the top and bottom of each word. Fig. 5 demonstrates this difference. To make comparison more fair, we crop our generated words on the top and bottom to the first ink pixel (value less than 200). This cropping resembles the segmentation of the word images and slightly improves our FID score.

We also question in general the validity of using FID score for handwriting images. As Inception-v3 is trained on natural images, not handwriting, FID seems ill-suited for evaluating the quality of handwriting images. Further investigation is required into the topic of applying FID to image domains other than natural images.

For GS [22], the data is expected to all be the same size. Because the dataset has variable width images and our method produces variable width images, we pad images to be the same width. Neither [2] nor [6] report how they handled this. Like [6], we only run the experiment once.

S.2 Human Study Details

We submitted 78 image tasks to Amazon Mechanical Turk (35 real, 35 generated, 8 poorly generated), requesting 200 workers to review each image. Each task consisted of instructions, with example images, a task image (real, generated, or poorly generated) and two multiple choice questions. The first question asked the worker to select the correct transcription for the task image. Two choices were shown, one with the correct transcription, the other a permutation of the correct transcription's words (where the first and last words remained in the same place). We removed punctuation so the permutation didn't create artifacts that made the choice too easy. This was to ensure the worker actually looked at the image and was paying attention to what they were doing. The second asked if they thought the image was written by a human or a computer.

The interface the workers saw can be seen in Fig. S1. The correct and incorrect transcription options were randomly ordered, the options between human and computer remained in the same order.

The real instances used in the study were randomly selected from the test set. The generated images used the same text as the selected real instances, but the styles were from interpolations between styles extracted from randomly selected test set images.

To help measure the reliability of the workers, we included poorly generated images which should appear to not be written by a human. These were created using a model only trained 2,000 iterations. The responses on these images were not included in the final evaluation, but were held out to help gauge the confidence that can be placed in the workers efforts. The poorly generated images used in the study are shown in Fig. S2. The generated and dataset images used in the study are in Figs. S3 and S4 respectively.

The transcription question was used to filter out workers which were unreliable (likely clicking random responses to complete the tasks quickly). We only used workers who had at least 90% accuracy on transcription (permutations can sometimes be very close to the correct transcription leading to some error in even engaged workers). Additionally, we only used workers we had at least 6 responses for. The selected workers had 89.5% accuracy on the poorly generated images, the left-out workers had 79.0% accuracy.

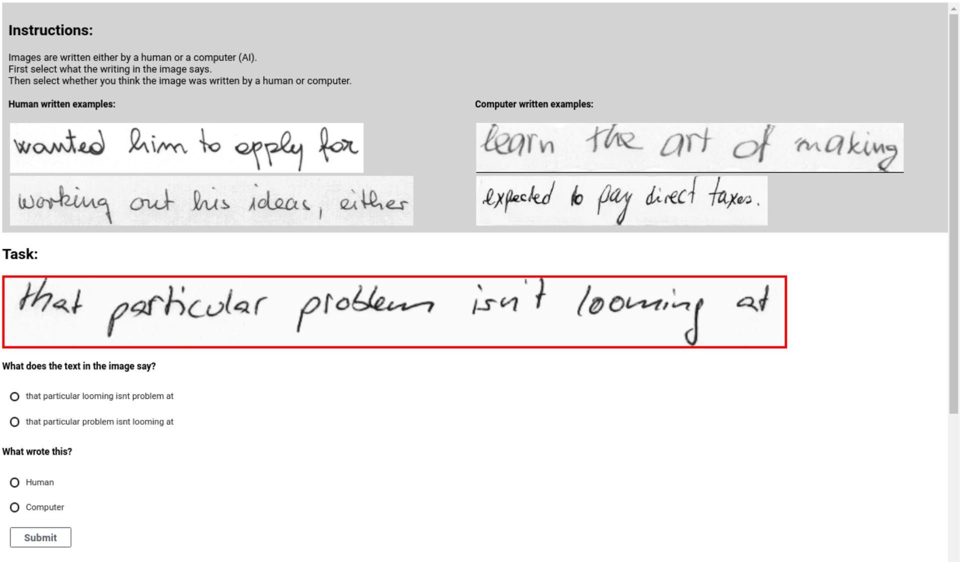

Figure S1: A screenshot of the interface the workers saw when completing a task. The example images remained the same each task. The order in which the correct and incorrect transcription responses were placed was random. We kept the task image large so detail could be seen.

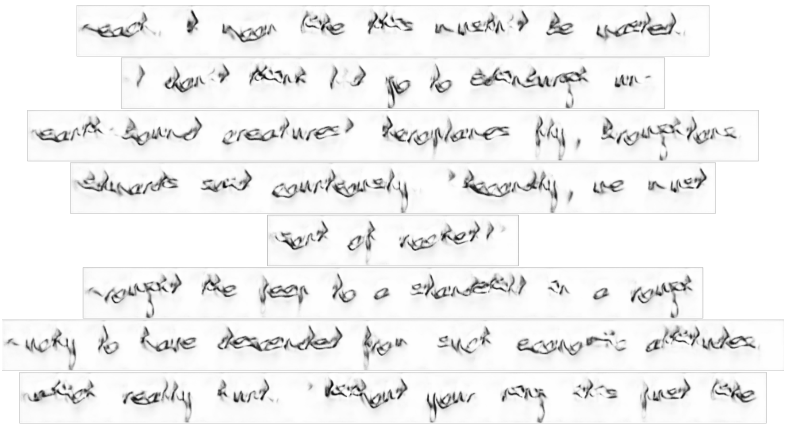

Figure S2: Poorly generated images from an intentionally under-trained model used in human study to evaluate participant ability or attention. These samples are *not* from our final model.

Figure S3: Generated images used in human study that were generated using random styles (i.e. random interpolation of style vectors extracted from random pairs of real images from IAM) and random text from the IAM corpus.

Figure S4: Dataset images used in human study. These are randomly sampled from IAM.

S.3 Additional Generation Results

We here show additional results from our model. Fig. S5 shows additional examples of style interpolation. Figs. S6 and S7 shows generation using random interpolated/extrapolated styles with fixed and varying text respectively. Figs. S8 and S9 show reconstruction results.

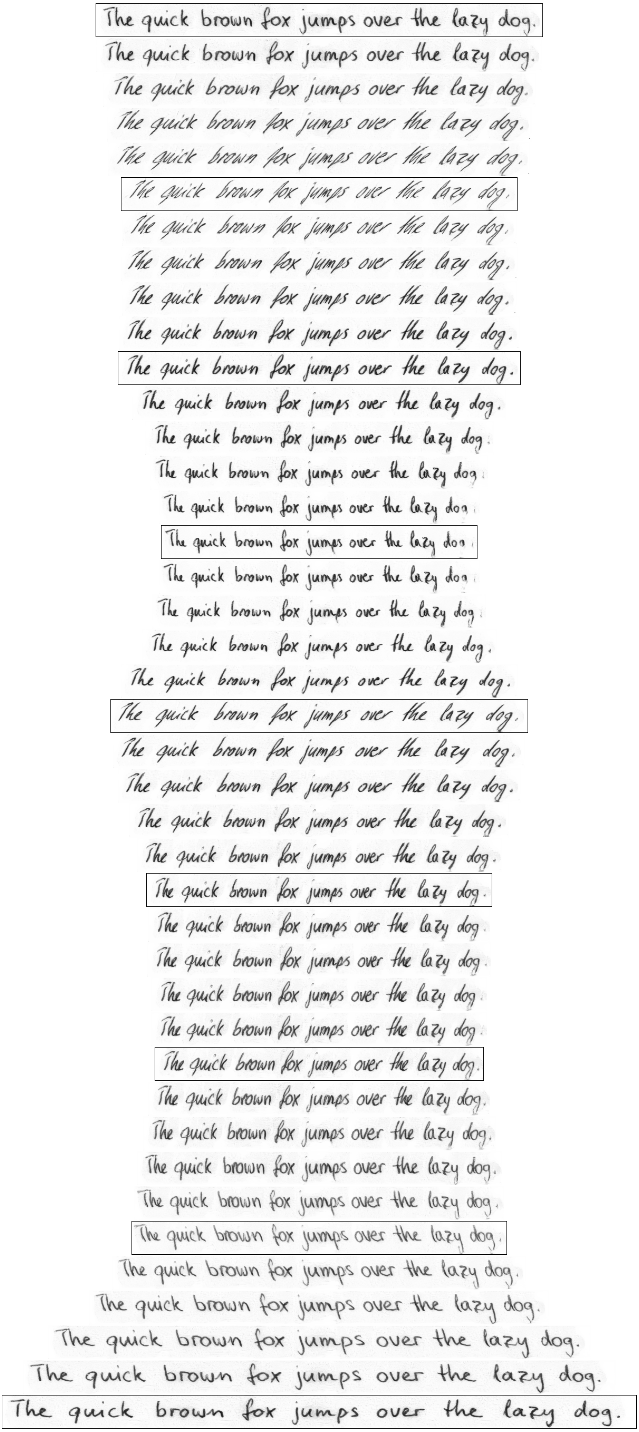

Figure S5: Additional interpolation results between 9 different styles extracted from test set images.

Figure S6: Additional generation results using random extra/interpolations between test set styles using the same text.

Figure S7: Additional generation results using random extra/interpolations between test set styles using varying text.

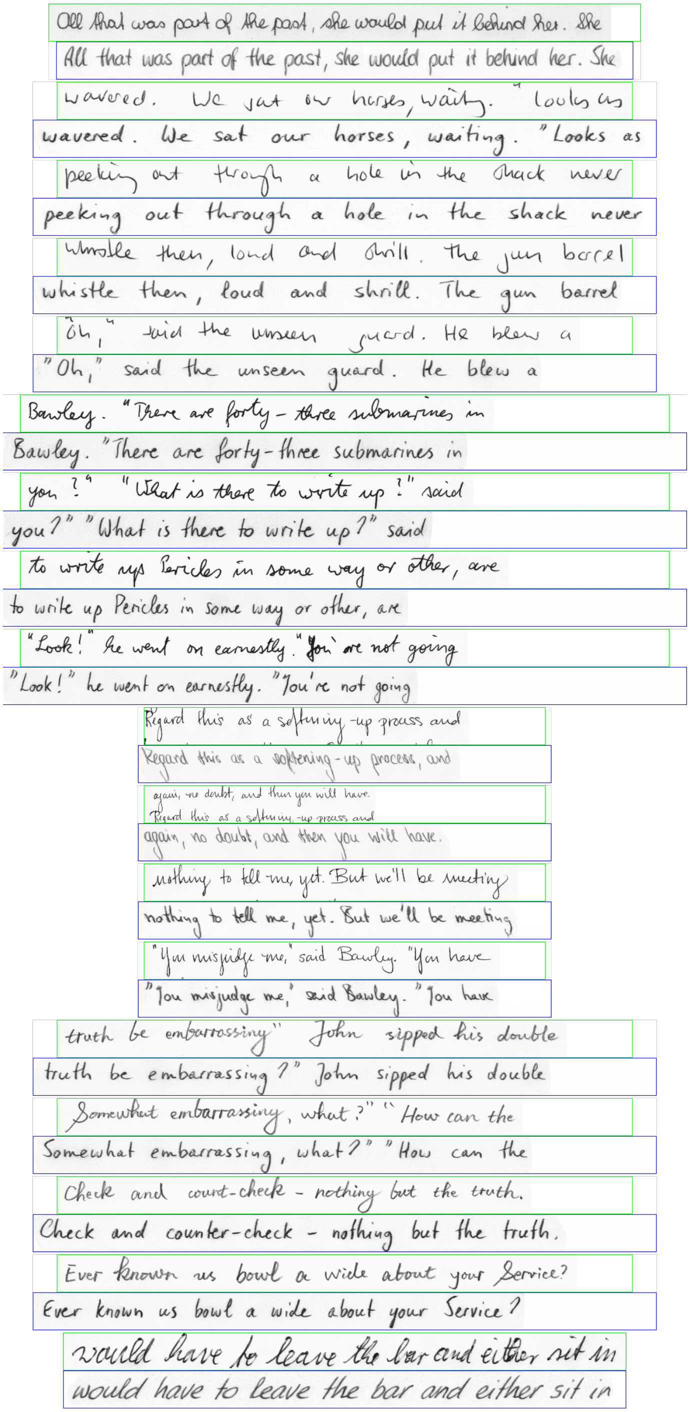

Figure S8: Additional Reconstruction results. Green is original, blue is our model's reconstruction.

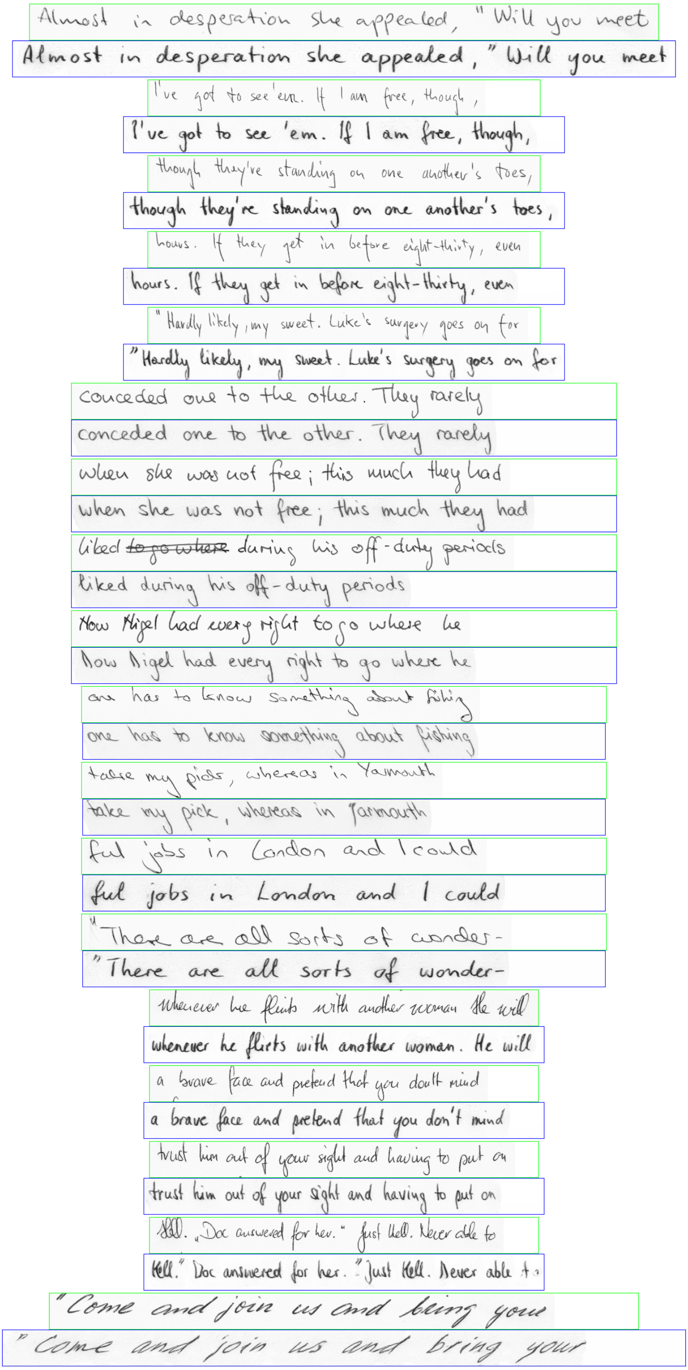

Figure S9: Additional Reconstruction results. Green is original, blue is our model's reconstruction.

S.4 Additional Ablation Results

We present additional results for each of the ablation models:

- Fig. S10: No reconstruction loss
- Fig. S11: No adversarial loss
- Fig. S12: No handwriting recognition supervision
- Fig. S13: No character specific components of *S*
- Fig. S14: No pixel reconstruction loss

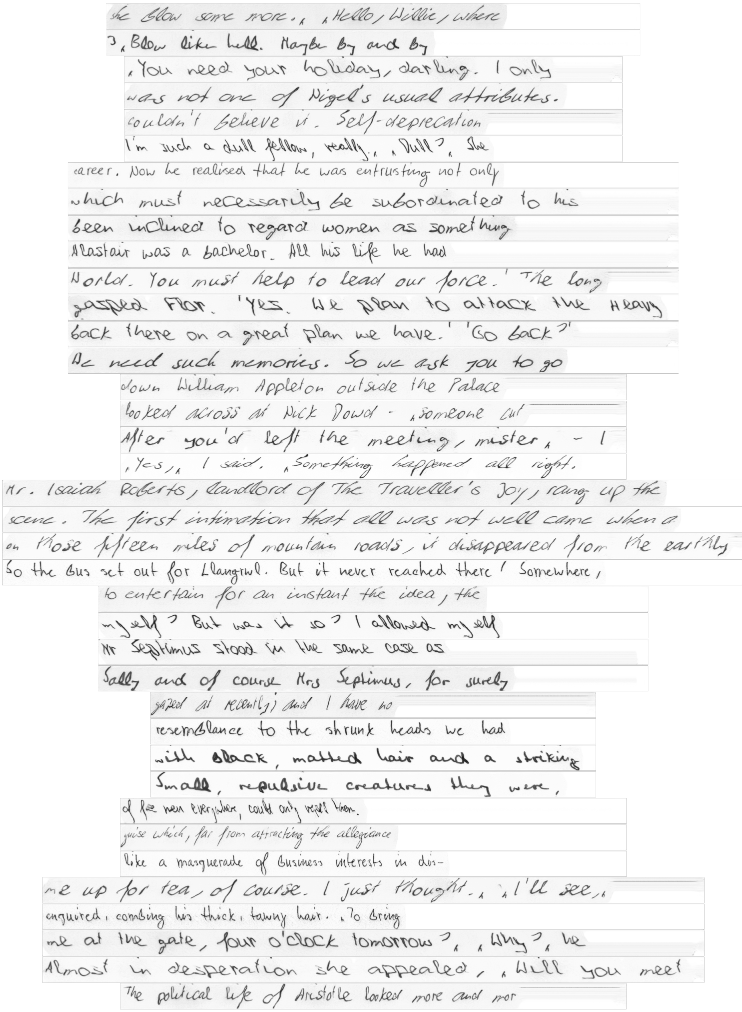

Figure S10: Additional ablation results, without the reconstruction losses (random styles).

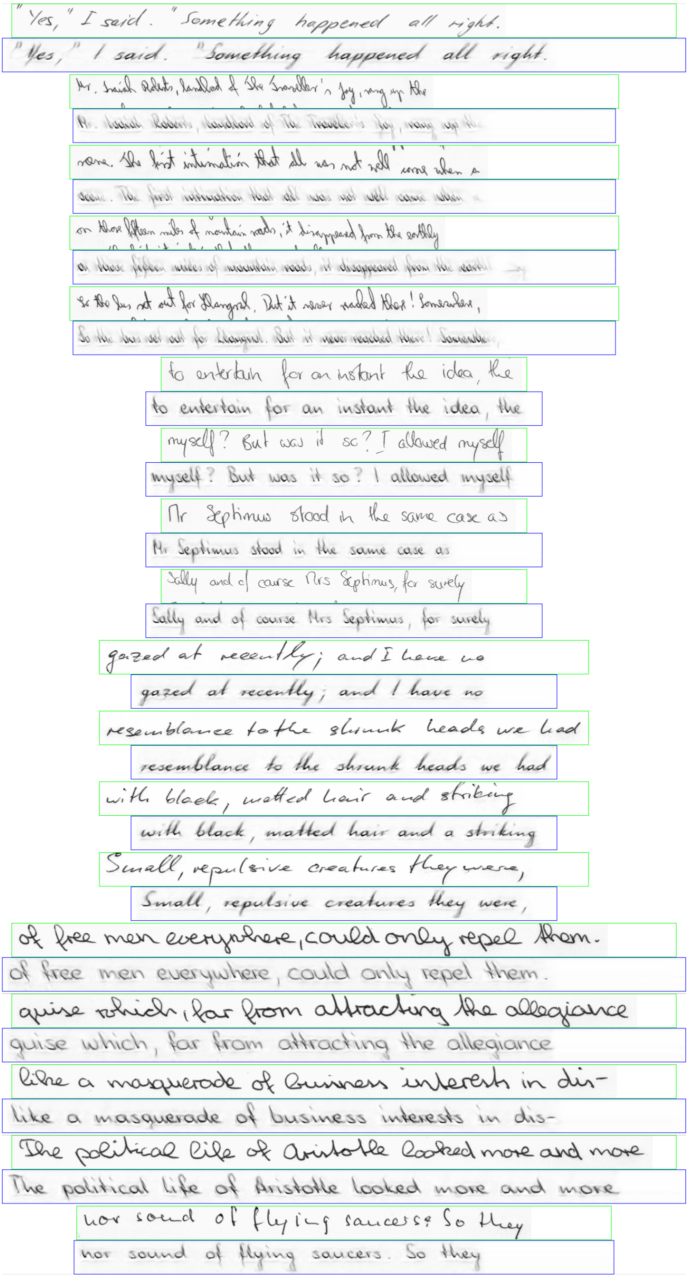

Figure S11: Additional ablation results, without adversarial loss.

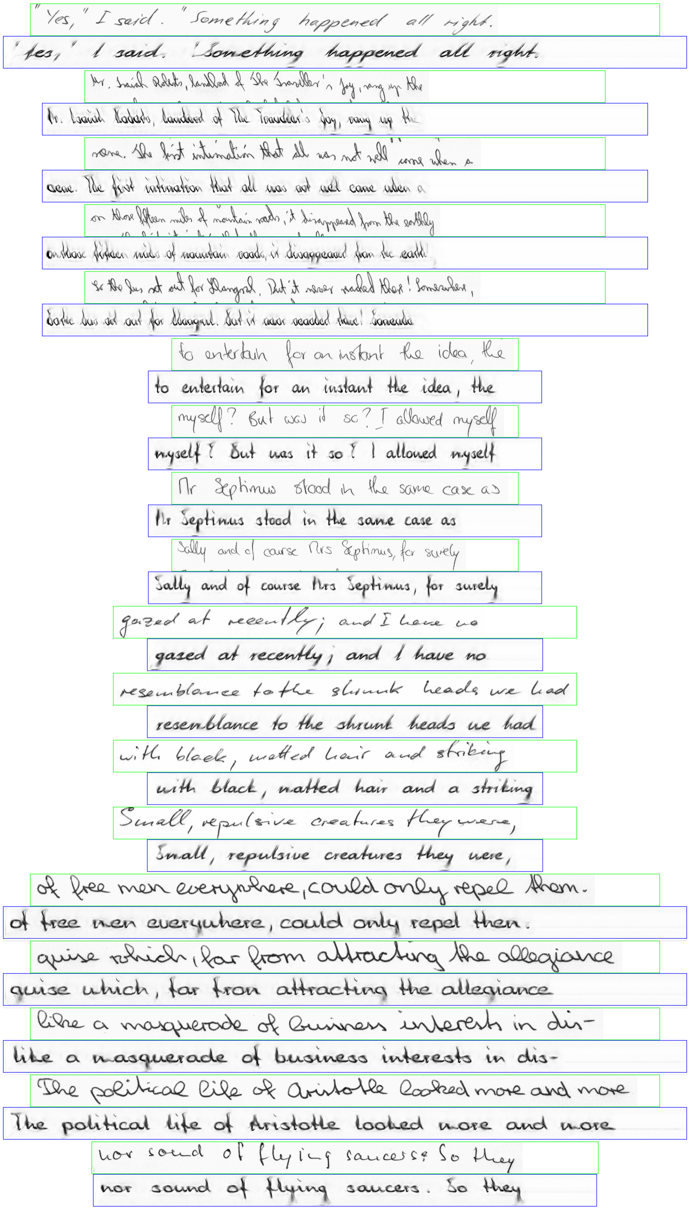

Figure S12: Additional ablation results, without handwriting recognition supervision.

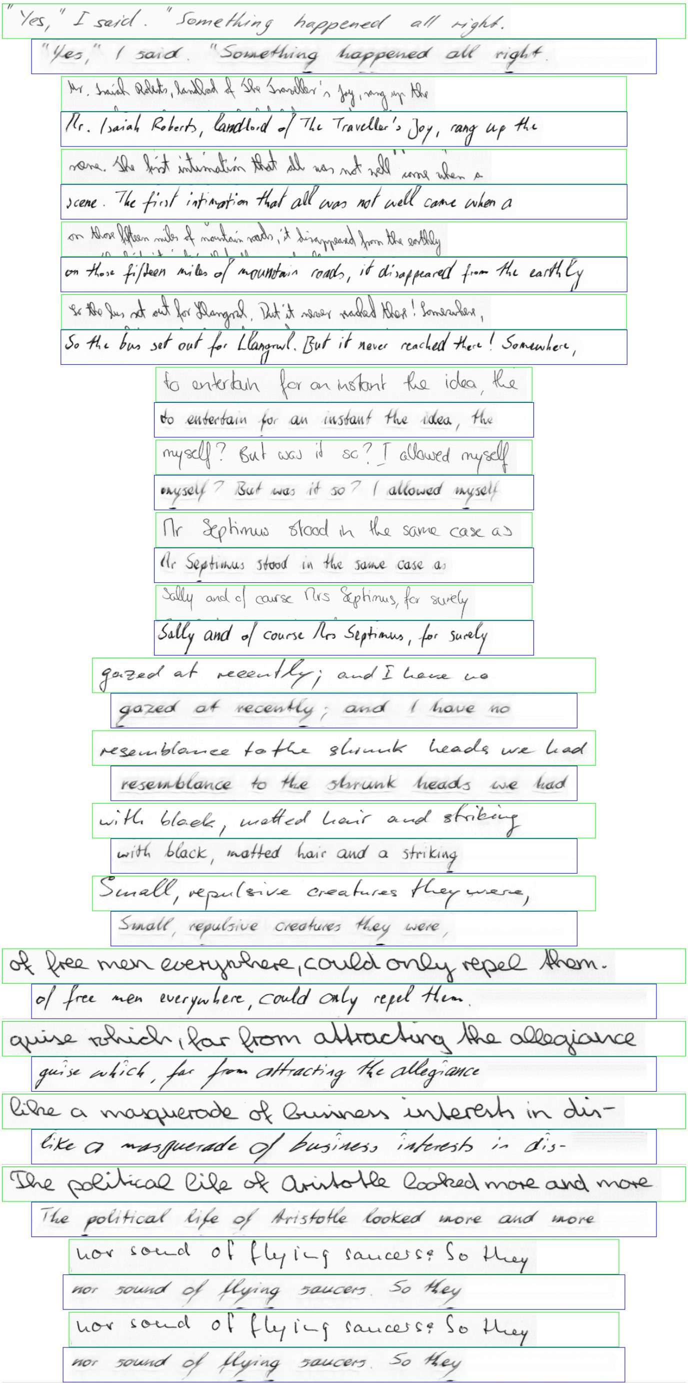

Figure S13: Additional ablation results, without character specific components of *S*.

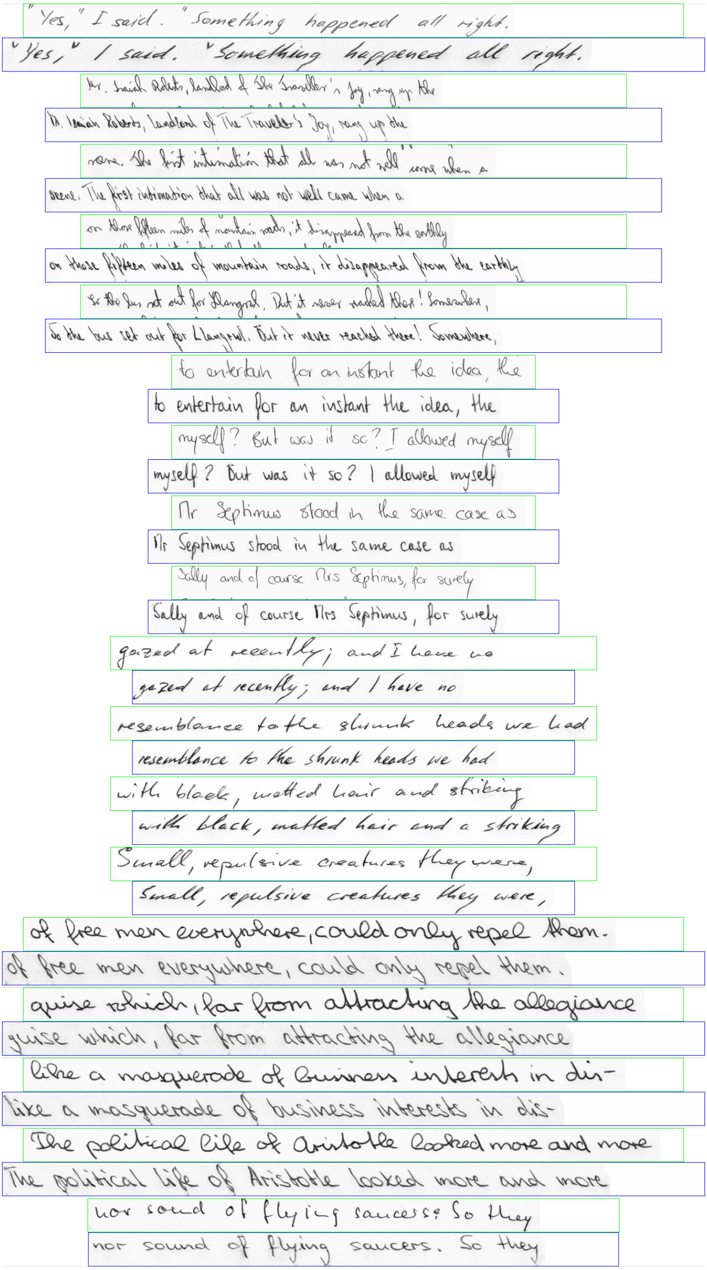

Figure S14: Additional ablation results, without pixel-wise reconstruction loss.

S.5 Model Specifications

We present here detailed diagrams of various parts of the model:

- Fig. S15: The handwriting recognition model R
- Fig. S16: The generator G
- Fig. S17: The auxiliary spacing network C
- Fig. S18: The discriminator D
- Fig. S19: The encoder E
- Fig. S20: The style extractor S

The encoder E is trained using the same IAM training set. It is jointly trained with a decoder as an autoencoder with an L1 reconstruction loss and as a handwriting recognition network with a recognition head using the CTC loss. It is trained with the Adam optimizer 6000 iterations with a learning rate of 0.0002.

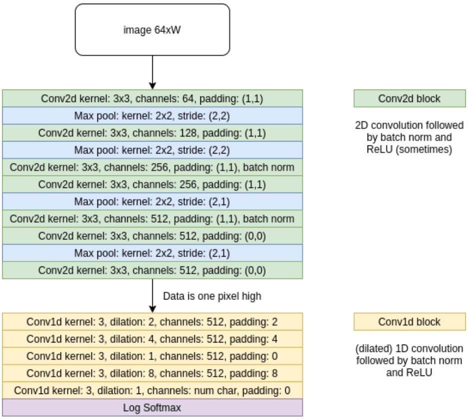

Figure S15: Handwriting recognition network R architecture

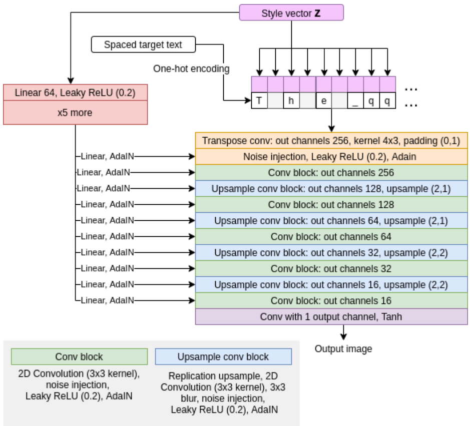

Figure S16: Generator G architecture

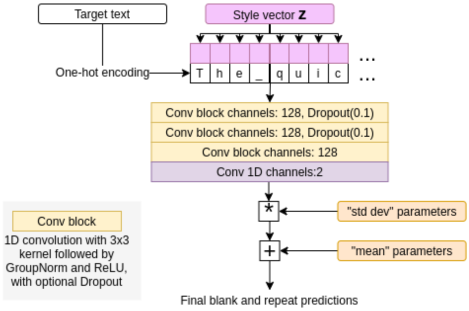

Figure S17: Spacer network C which predicts the spaced text. It predicts the number of blanks proceeding each character and the number of times the character should be repeated.

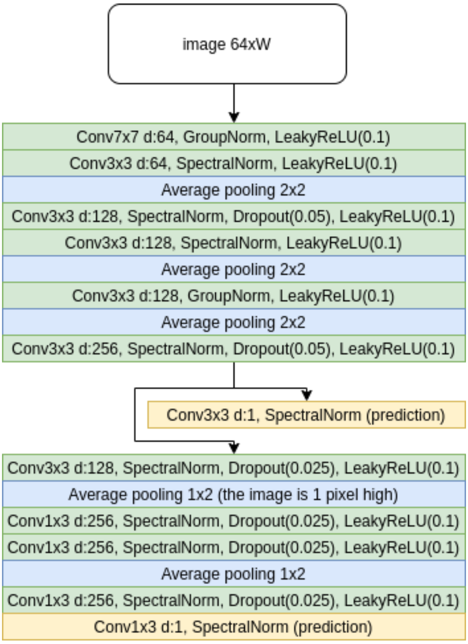

Figure S18: Discriminator D architecture.

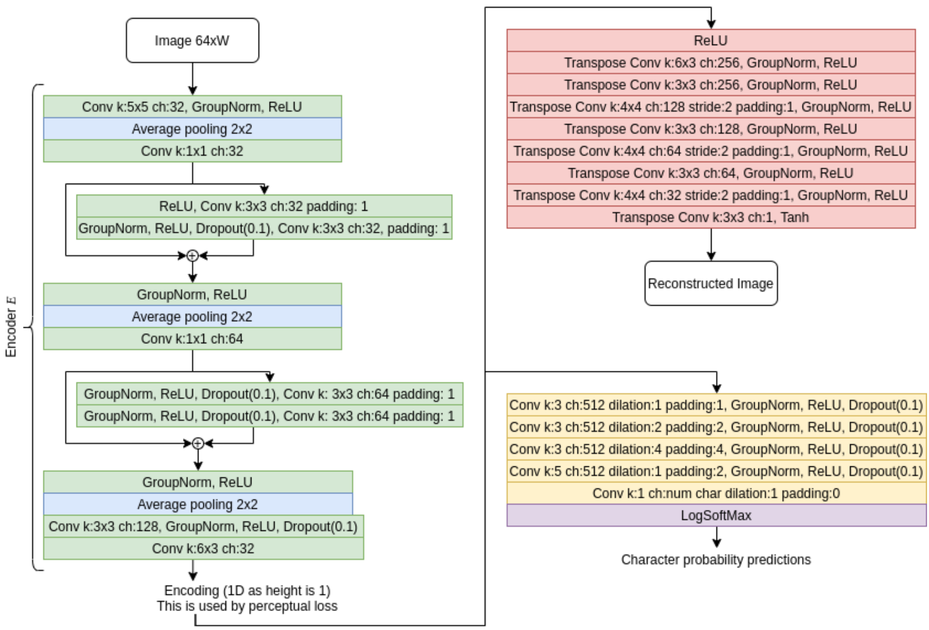

Figure S19: Encoder network E (green) and auxiliary decoder (red) and recognition head (yellow) used to train E.

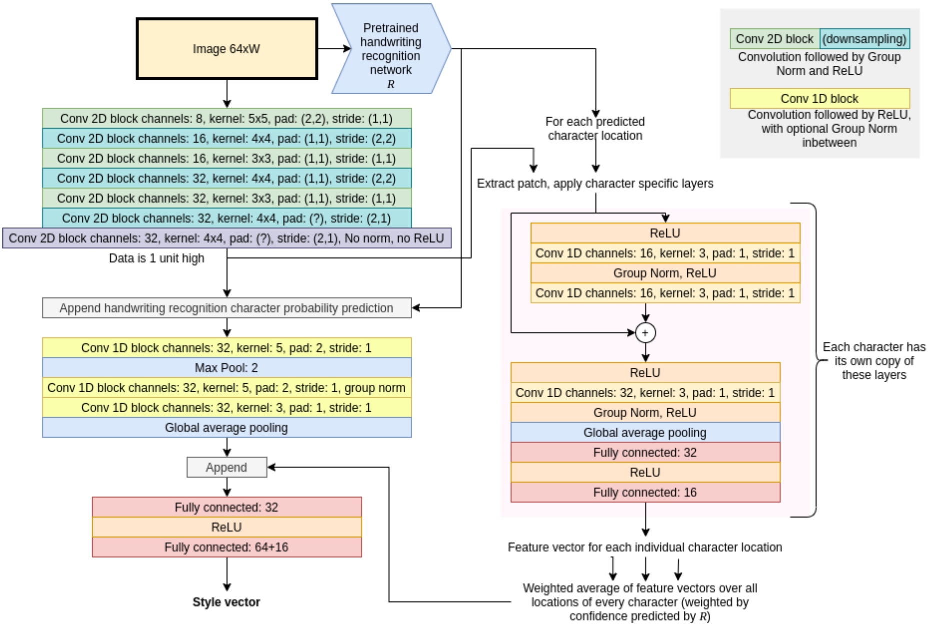

Figure S20: Style Extractor S. It leverages the output of R both as additional input and to (roughly) locate characters. The locations are used to crop features to pass to character specific layers (the learn to extract features for one character).