# Zero-Shot Styled Text Image Generation, but Make It Autoregressive

## Supplementary Material

## Contents


**A Further Details about the Training Dataset** 1
**B Details about the Considered Datasets** 2
**C Details about the Considered Scores** 3
**D Analysis on the Style Input Length** 3
**E Analysis on the Styled Output Length** 4
**F. Detailed Comparison Between VAEs** 6
**G Multi-stage Training Ablation** 6
**H Additional Quantitative Results** 6
**I. Additional Qualitative Results** 6
**J. Editing Application** 11


## A. Further Details about the Training Dataset

The proposed synthetic dataset[1] comprises RGB images that feature text lines superimposed on diverse background patterns (Figure 2). The process to create a text image is divided into two macro-steps: Text Content Sampling and Text Line Rendering.

**Text Content Sampling.** First, we formed the text lines to render in the samples of the dataset. To this end, we consider the following English corpora available on the NLTK[2] library: `abc`, `brown`, `genesis`, `inaugural`, `state_union`, and `webtext`. To ensure a balanced character distribution and to address the learning of rare characters, we employ a rarity-based weighting strategy. Specifically, each word in the corpora is assigned a weight, allowing for weighted sampling from the resulting distribution. The weight of each word is computed based on the frequency of unigrams and bigrams contained within it. This approach prioritizes words that include infrequent character patterns, enabling the HTG model to learn from the long-tail distribution of characters more effectively. The Python implementation of the algorithm for computing the weight of each word is reported in Listing 1. The function `word_weight(word)` calculates a score for each word based on the unigram and bigram frequencies, stored in `u_counts` and `b_counts`, respectively. These counts represent the frequencies of unigrams and bigrams across the entire corpus. The resulting word weight is the average of the unigram and bigram-based scores. In Figure 1, we show the character's distribution with the custom weights per word and the naive sampling (original). Each bar of the distribution represents a character in our charset (which contains 157 characters) in this order: [ , e, a, n, s, i, r, t, o, l, A, ., m, c, d, u, 0, S, h, E, p, M, g, C, R, 1, I, -, D, b, T, 2, N, 3, O, y, P, B, L, ', F, k, f, H, 4, 5, G, U, v, 7, 8, ", 6, 9, V, K, x, w, Y, ), ä, W, ,, *, z, :, J, (, X, !, j, /, _, é, >, Z, ?, Q, ;, $, å, ], ö, q, %, <, [, á, ã, #, ¡, &, }, +, ü, â, ó, Ą, ê, =, {, ô, \, í, è, », ‼, É, ú, @, À, à, `, ç, ù, ˜, ¨, ´, Ş, ˝, µ, ‟, Ă, č, °, Â, ®, ŕ, --, ¢, Ä, û, Ó, ″, ƒ, Ť, ˆ, ï, ˘, |, î, ë, ¦, Ś, õ, Ö, Á, °, Å, t', Ł, Ü, ß, ż, ñ, Í, d']. Note that, our current Emuru is trained on a dataset derived from fonts covering primarily Latin characters, there is nothing in the approach itself that prohibits extension to non-Latin scripts. Specifically, to extend Emuru to non-Latin scripts, its VAE should be re-trained on a similarly-built synthetic dataset of non-Latin strings, rendered with fonts that support the desired characters. Given that there are a number of such fonts available online, we argue that there is no obvious method or model limitation as to why the results of Emuru would not translate also to non-Latin scripts.

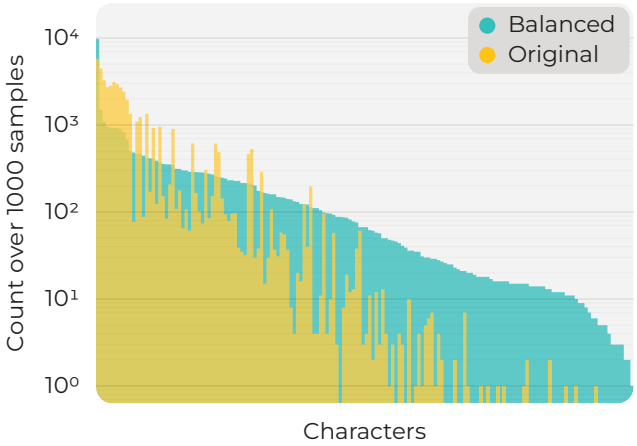


Figure 1. Characters frequency in text from the original corpora used to generate the synthetic training dataset, and from the balanced version of the corpora.

**Text Line Rendering.** Once we sample a sentence, we select a font from a list of 100,000 fonts and render the text image on a white background, obtaining $\hat{I}_T$. Then, we sample a background $\hat{I}_B$ from a set of images depicting reasonable supports for writings (*e.g.*,paper-like textures, wood, walls), a transparency value $\alpha \in [0.5, 1]$ and process $\hat{I}_T$ and $\hat{I}_B$ in parallel. To obtain $I_T$ (Figure 2, center), we use the following random transforms: Rotation, Warping, Gaussian Blur, Dilation, and Color Jitter (with probabilities respectively set to 0.5, 1.0, 0.5, 0.1, and 0.5) and apply the transparency $\alpha$. To obtain $I_B$ (Figure 2, right), we use

[1] https://huggingface.co/datasets/blowing-up-groundhogs/font-square-v2

[2] https://www.nltk.org/


Figure 2. Exemplar sample images $I$ from the synthetic dataset used for training Emuru (left). These samples consist of a rendered greyscale text image $I_T$ (center) superimposed to RGB paper-like background images $I_B$ (right).

```python
def word_weight(word):
    # Compute the score
    # based on the unigrams
    u_score = 0
    for c in word:
        u_score += u_counts[c]
    u_score /= len(word)

    # Compute the score
    # based on the bigrams
    bigrams = pairwise(f' {word} ')
    bigrams = list(bigrams)
    b_score = 0
    for b in bigrams:
        b_score += b_counts[''.join(b)]
    b_score /= len(bigrams)

    # Average the two scores
    return (u_score + b_score) / 2
```

Listing 1. Python code for computing the words weight, needed for the character frequency balancing procedure adopted to obtain the text in our synthetic training dataset.

the following random transforms: Dilation, Color Jitter, and Random Inversion of the values (with probabilities respectively set to 0.1, 0.5, 0.2). The image $I$ (Figure 2, left) is the result of superimposing $I_T$ on $I_B$.

## B. Details about the Considered Datasets

In this section, we give further details about the datasets we use for evaluation.

**IAM.** The IAM Handwriting Database [11] is a collection of greyscale document scans written in English by multiple writers. The dataset features free-layout modern English text lines from the Lancaster-Oslo/Bergen (LOB) corpus. The number of pages per writer varies significantly, ranging from 1 to 10. The dataset comes with both line-level and word-level segmentation information. The HTG community commonly used this dataset for training and evaluation of the proposed approaches and has adopted a split defined in [7], in which the samples from 339 writers are used for training, and those from 161 other writers are used for test. In this work, we adopt the same split. However, while most of the competitors are trained on IAM (and, therefore, see the test set as in-distribution samples), we never use any training data from it to train our model (therefore, test samples are out-of-distribution for us).

**CVL.** The CVL Database [8] is a collection of RGB scans of English and German manuscripts written with ink on white paper by 310 writers. In this work, we consider the line-level annotation of the dataset and perform the evaluation on its test set, which contains lines from 283 different authors. We include this dataset since it is mostly in the same language as the commonly-adopted IAM, but the handwriting styles are out-of-distribution *w.r.t.* IAM.

**RIMES.** The RIMES Database [1] is a collection of binary

images of customer service-themed letters in French, written by multiple authors. For our experiments, we consider the lines in the official test split of the dataset, which have been written by 100 authors. We include this dataset as a more challenging collection of out-of-distribution generation cases. In fact, both the language and the styles differ from those in IAM.

**Karaoke.**[3] To assess the capabilities of Emuru and the existing handwriting style imitation approaches, we devise a dataset of text lines images obtained by rendering song lyrics with 100 publicly available fonts[4], 50 calligraphy, and 50 typewritten, on a white background. This dataset, especially its typewritten split, serves as a challenging test to measure the performance of HTG models when dealing with significantly out-of-distribution, font-like styles.

## C. Details about the Considered Scores

Measuring the performance of styled HTG models is a challenging task per se. Early works on HTG simply employ the Fréchet Inception Distance (FID) to measure the realism of the generated images, and sometimes the CER to measure their readability. Recent works [14–16] have highlighted the limitations of such scores (especially the FID) and thus proposed to assess the performance of HTG models both by using multiple scores or introducing novel, task-specific scores. In line with these works, for our evaluation, we adopt the following scores (some of which have been reported in the main paper).

**Fréchet Inception Distance (FID).** The FID [6] is widely employed in HTG literature and captures the realism/similarity between the distribution of the real images and the generated ones. In the context of HTG, it somehow captures the texture-wise style similarity. Note that in most HTG works, the reported FID is computed only on the initial square crop of the real and generated images. In this work, we compute it on non-overlapping crops obtained from the entire images.

**Kernel Inception Distance (KID).** Similar to the FID, also the KID [2] captures the realism of the generated images by comparing their distribution to the real images distribution. Despite being less used in image generation evaluation, it is more numerically stable than the FID [15].

**Binarized FID (BFID).** To reduce the impact of the texture (both of the strokes and the background) on the similarity between generated and real images, we introduce this variant of the FID score by computing it on binarized images. Intuitively this measure should capture mostly the perceptual handwriting style and disregard color or textures.

**Binarized KID (BKID).** Similar to the BFID, this score is obtained by computing the KID on the binarized reference and generated images.

**Absolute Character Error Rate Difference ($\Delta$CER).** We consider the State-of-the-Art TrOCR-Base [10] model, which can handle both handwritten and typewritten images and compute its CER on the reference and generated images to capture their readability while taking into consideration the style. Intuitively, we want the generated images to be as readable as the reference ones (which suggest that they actually contain text) but not significantly more, which could be that the HTG model has collapsed to a very readable style that does not necessarily resemble the reference.

**Handwriting Distance (HWD).** The HWD [15] is a perceptual score that has been recently proposed specifically for HTG and captures the similarity in terms of calligraphic style between the real and generated images pairs. For this reason, we include it in our evaluation setup.

**Absolute Intra Learned Perceptual Image Patch Similarity Difference ($\Delta$I-LPIPS).** To measure the style preservation over long generated images, we split each image into non-overlapping crops and compute the LPIPS [17] (which is based on the similarity of feature maps) between all possible combinations of crops of the same image. We repeat this process for both the real and the generated image separately and then compute the absolute difference between these values. We want its value on the generated images to be similar to what is obtained on the real ones (to ensure mimicking the degree of variability of the reference style) and not significantly lower, which could indicate mode collapse to a repeated stroke.

## D. Analysis on the Style Input Length

As mentioned in Section 2, existing HTG methods have a variety of settings for the style reference image, ranging from single words [3–5, 9] to entire paragraphs [12]. These approaches offer different advantages depending on the use case, with some models demonstrating some adaptability to a style from minimal input and others leveraging more context for improved style fidelity. We argue that an HTG model should be flexible to the number of input words available, handling both data-scarce scenarios with limited reference samples and situations with more style images available. To deepen the analysis on this aspect, we evaluate the performance of Emuru when generating IAM lines given increasingly longer reference style images, which we build starting from the word-level annotations of the IAM dataset. Note that the IAM dataset contains all the information needed to determine which sequence of words appears within each line and where to cut the line image to obtain a new line image with the desired width. Therefore, to generate the set of style samples within a specific width range, we

[3] https://huggingface.co/datasets/blowing-up-groundhogs/karaoke, https://open.spotify.com/playlist/5GTPvzVNedUIkYkV7xOtgI

[4] https://fonts.google.com/

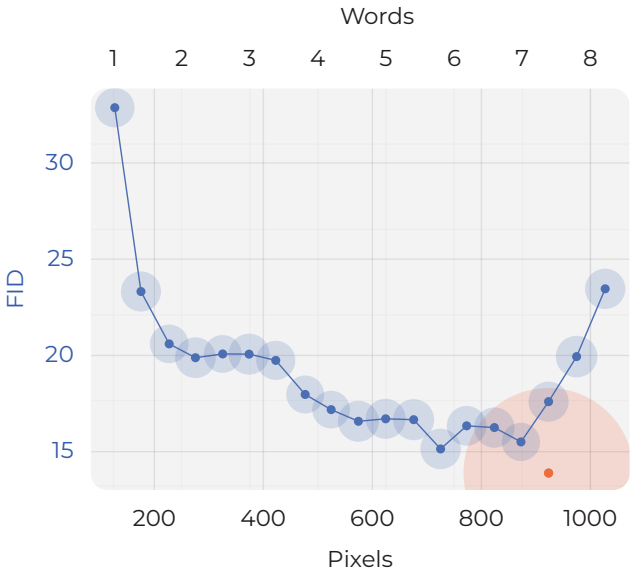


Figure 3. Emuru FID scores for IAM line generation using style inputs of varying length (in blue), compared to those generated with the original input style (in red). The length of the style input is expressed as the average pixel width of the style samples, with standard deviations represented by semi-transparent circles behind the data points. Additionally, the average number of words in each style sample is indicated, where an "average word" corresponds to approximately 125 pixels in width.

exploit the IAM information of the bounding boxes around the words inside the lines. Specifically, we group images whose width is in a range of $\pm$25px around average width values that vary from a minimum of 125px to a maximum of 1025px with steps of 50px. In Figure 3, we report the results of this analysis. As expected, the performance improves with increasing input length, showing that our model is flexible to different scenarios and more applicable to real-world handwriting generation tasks.

## E. Analysis on the Styled Output Length

To assess the optimal output text length manageable by Emuru, we perform experiments by increasing the number of output characters to render in the styles from the Karaoke dataset. We report FID and CER scores in Figure 4, representing style fidelity and text readability. We observe that the model performs best when generating text lines between 25 and 75 characters, which aligns with the training sequence length. Outside this range, Emuru exhibits performance degradation in both style fidelity and text readability. Note that some workarounds for improving the quality of generated long sequences can be applied, as mentioned in the following. In Figure 5, we report qualitative results of image generation with Emuru for increasingly longer text lines. In the top part of the figure, we feed Emuru with the


Figure 4. Emuru FID and CER when generating increasingly longer lines in unseen styles from the Karaoke dataset, separated into calligraphy and typewritten styles. We express the generated lines length in terms of the number of characters contained.

reference style image and the entire text line to be generated all at once. As we can see, the model maintains style consistency across long outputs but struggles to correctly render all the words as their number increases. This discrepancy between the input and the output text is reflected in the $\Delta$CER score, as observed in the quantitative analysis in Section 4 Section 4. To improve this aspect, in the bottom part of Figure 5, we also show a simple yet effective solution for generating longer lines with increased text fidelity. Specifically, we let Emuru iteratively generate one word at a time by using the previously generated line as a style reference. This incremental approach improves the alignment of the output image with the input text with a computational overhead of 29.5% (2.24s), mainly due to the additional 10 padding embeddings that the model has to generate as a stop signal after each new word generation.


Figure 5. Increasingly long text image generation with Emuru. We use the top, colored image as style input and task the model to generate an incrementally longer sentence in each line. In the top figure, we task the model to generate the whole sentence in one go, while, in the bottom one, each line is used as the style sample of the next one and the model generates only the last word.

## F. Detailed Comparison Between VAEs

In this section we expand the analysis on the performance of Emuru VAE on multiple datasets reported in Section 4. As the performance of the autoregressive Transformer in Emuru is tied to the quality of the latent vectors it has been trained on, this detailed analysis helps determine a soft bound on the reconstruction quality. In Table 1, we report the results of the reconstruction performance comparison on different datasets, which we compute on images reconstructed with different VAEs: Emuru VAE, SD1.5 VAE (used by DiffPen and OneDM), and SD3 VAE (the current State-of-the-Art in image reconstruction). As we can see, Emuru VAE consistently gives the best or second-best BFID and BKID due to its ability to closely capture and reproduce the handwritten style. Looking at FID and KID, in some datasets the performance is a bit lower *w.r.t.*the alternatives considered, likely due to the fact that Emuru VAE is trained to reconstruct text on a clean background (and not the original input image). Moreover, the $\Delta$I-LPIPS is consistently better, which indicates that the images reproduced by our VAE have a consistent style in all their parts. In general, we can observe either better or comparable performance to the other considered VAEs. The tailored training of our VAE allows us to obtain these results with a much more lightweight model and a more compressed latent space ($\sim$16% of the parameters of the other VAEs and 1 latent channel instead of 4 and 16, respectively for SD1.5 VAE and SD3 VAE).

## G. Multi-stage Training Ablation

In Table 2, we report a detailed analysis of the performance obtained with Emuru across different training stages. As explained in Section 3, we train Emuru with the MSE loss in two stages: pretraining with text lines containing 4 to 7 words (**Pretraining** in Table 2) and fine-tuning on text lines containing 1 to 32 words (**+ Var. len. ft** in Table 2). Predictably, there is a significant performance improvement across all metrics and datasets, which contain variable-length text lines. These results indicate the benefit of the two-stage training approach. During the first stage, the autoregressive Transformer learns to correlate the text input to the VAE embeddings extracted from the reference style image in a simpler setup. Then, in the second stage, the model learns to deal with variable-length inputs and can also focus on understanding when to stop generating. In fact, although the pertaining stage is longer, the samples used in this stage have similar text length and reference image width, and thus, batches of such samples are more informative since they contain less padding. On the other hand, the images used in the variable length fine-tuning stage need to be padded more to be batched, which makes this stage less efficient. Nonetheless, thanks to pertaining, few iterations are sufficient to achieve good performance.

| | FID↓ | BFID↓ | KID↓ | BKID↓ | ΔCER↓ | HWD↓ | ΔI-LPIPS↓ |
|---|---|---|---|---|---|---|---|
| **IAM words** | | | | | | | |
| **SD1.5 VAE** | 21.15 | 2.42 | 20.72 | 1.65 | **0.01** | 0.92 | 2.34 |
| **SD3 VAE** | **15.28** | **2.09** | **13.62** | 1.35 | **0.01** | 0.84 | 2.72 |
| **Emuru VAE** | 27.74 | 2.60 | 23.41 | **1.32** | 0.08 | **0.64** | **0.02** |
| **IAM lines** | | | | | | | |
| **SD1.5 VAE** | 15.87 | 1.64 | 16.47 | 1.27 | 0.01 | 0.68 | 74.04 |
| **SD3 VAE** | **9.80** | 1.77 | **9.66** | 1.53 | **0.00** | **0.59** | 82.95 |
| **Emuru VAE** | 16.74 | **1.11** | 13.76 | **0.53** | 0.01 | 0.76 | **6.24** |
| **CVL** | | | | | | | |
| **SD1.5 VAE** | 83.33 | 34.82 | 101.89 | 34.74 | 0.01 | 0.63 | 45.74 |
| **SD3 VAE** | 70.84 | 8.77 | 84.68 | 9.30 | **0.00** | **0.57** | 52.81 |
| **Emuru VAE** | **11.25** | **1.10** | **9.77** | **0.53** | 0.05 | 0.79 | **5.28** |
| **RIMES** | | | | | | | |
| **SD1.5 VAE** | 20.35 | 2.77 | 19.57 | 2.14 | 0.02 | **0.79** | **0.50** |
| **SD3 VAE** | **19.07** | 5.10 | **17.65** | 4.95 | 0.03 | 0.90 | 3.52 |
| **Emuru VAE** | 47.79 | **2.62** | 44.27 | **1.45** | **0.01** | 1.24 | 39.72 |
| **Karaoke Typewritten** | | | | | | | |
| **SD1.5 VAE** | 12.28 | 0.87 | 9.54 | 0.31 | **0.01** | **0.55** | 20.60 |
| **SD3 VAE** | **5.29** | 1.13 | **2.89** | 0.50 | **0.01** | 0.65 | 32.76 |
| **Emuru VAE** | 6.55 | **0.75** | 3.92 | **0.09** | **0.01** | 0.56 | **7.43** |
| **Karaoke Calligraphy** | | | | | | | |
| **SD1.5 VAE** | 23.38 | 1.63 | 24.65 | 1.16 | **0.00** | 1.04 | **9.37** |
| **SD3 VAE** | 11.09 | 2.80 | 9.58 | 2.37 | **0.00** | **0.87** | 13.84 |
| **Emuru VAE** | **5.28** | **1.55** | **2.98** | **0.51** | **0.00** | 1.13 | 10.63 |

Table 1. Comparison between the VAE featured in Emuru and other VAEs used in popular Diffusion Models when reproducing the images in the test set of different HTG datasets. Note that none of these VAEs has been trained on the considered datasets. KID, BKID, and $\Delta$I-LPIPS are multiplied by $10^3$ and the best performance is in bold for readability.

## H. Additional Quantitative Results

In Tables 3 to 5, we report the performance of Emuru compared to State-of-the-Art HTG solutions on the considered datasets, expressed in terms of all the scores in the extended set of considered scores, as described in Appendix C.

## I. Additional Qualitative Results

In Figures 6 to 11, we report additional qualitative results comparing Emuru, the GAN-based VATr++ [16], and the Diffusion Model-based DiffPen [13] when generating images from the considered datasets. For each sample, we report the input style image used for guiding the generation and another reference image in the same style. We let the models generate the same text as in the reference to better observe the style imitation capabilities of the models.

| | IAM words | | | | | | |
|---|---|---|---|---|---|---|---|
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **Pretraining** | 85.15 | 53.28 | 83.53 | 52.54 | 0.56 | 3.51 | 0.39 |
| **+ Var. len. ft** | **63.61** | **37.73** | **62.34** | **37.22** | **0.19** | **3.03** | **0.16** |
| | **IAM lines** | | | | | | |
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **Pretraining** | 44.90 | 42.36 | 28.19 | 29.75 | 0.83 | 2.70 | **11.39** |
| **+ Var. len. ft** | **13.89** | **6.19** | **11.30** | **5.36** | **0.14** | **1.87** | 38.27 |
| | **CVL** | | | | | | |
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **Pretraining** | 77.23 | 70.52 | 69.28 | 54.47 | 0.87 | 2.52 | 6.69 |
| **+ Var. len. ft** | **14.39** | **10.77** | **12.34** | **10.54** | **0.13** | **1.82** | **0.75** |
| | **RIMES** | | | | | | |
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **Pretraining** | 100.34 | 75.90 | 76.32 | 56.07 | 0.69 | 2.96 | 87.25 |
| **+ Var. len. ft** | **26.93** | **13.36** | **21.19** | **9.40** | **0.25** | **2.18** | **47.96** |
| | **Karaoke Typewritten** | | | | | | |
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **Pretraining** | 47.06 | 39.28 | 22.76 | 14.69 | 0.67 | 2.03 | 45.52 |
| **+ Var. len. ft** | **9.85** | **4.33** | **5.60** | **1.24** | **0.11** | 1.28 | **5.07** |
| | **Karaoke Calligraphy** | | | | | | |
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **Pretraining** | 37.26 | 29.65 | 18.54 | 16.81 | 0.54 | 2.99 | 4.90 |
| **+ Var. len. ft** | **13.87** | **7.99** | **9.24** | **5.37** | **0.13** | **2.24** | **0.73** |

Table 2. Ablation analysis on the multi-stage training strategy for the autoregressive Transformer ('ft' stands for 'fine-tuning'). The scores are computed on all the considered datasets. KID, BKID, and ΔI-LPIPS are multiplied by $10^3$ and the best performance is in bold for readability.

| | IAM words | | | | | | |
|---|---|---|---|---|---|---|---|
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **TS-GAN** | 129.57 | 86.45 | 141.08 | 92.48 | 0.28 | 4.22 | 1.15 |
| **HiGAN+** | 50.19 | 21.92 | 43.39 | 14.21 | 0.20 | 3.12 | 0.17 |
| **HWT** | 27.83 | 15.09 | 19.64 | 11.95 | 0.15 | 2.01 | 0.06 |
| **VATr** | 30.26 | 15.81 | 22.31 | 13.37 | **0.00** | 2.19 | 0.56 |
| **VATr++** | 31.91 | 17.15 | 23.05 | 15.20 | 0.07 | 2.54 | 0.55 |
| **One-DM** | 27.54 | 10.73 | 21.39 | 6.72 | 0.10 | 2.28 | **0.00** |
| **DiffPen** | **15.54** | **6.06** | **11.55** | **3.93** | 0.06 | **1.78** | 0.53 |
| **Emuru** | 63.61 | 37.73 | 62.34 | 37.22 | 0.19 | 3.03 | 0.16 |
| | **IAM lines** | | | | | | |
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **TS-GAN** | 44.17 | 19.45 | 45.42 | 18.17 | 0.02 | 3.21 | 25.66 |
| **HiGAN+** | 74.41 | 34.18 | 77.27 | 31.24 | **0.00** | 3.25 | 63.43 |
| **HWT** | 44.72 | 30.26 | 43.49 | 31.14 | 0.33 | 2.97 | 34.31 |
| **VATr** | 35.32 | 27.97 | 33.61 | 27.81 | 0.02 | 2.37 | 28.17 |
| **VATr++** | 34.00 | 21.67 | 29.68 | 19.04 | 0.03 | 2.38 | 35.86 |
| **One-DM** | 43.89 | 21.54 | 44.48 | 20.94 | 0.13 | 2.83 | 78.42 |
| **DiffPen** | **12.89** | 6.87 | 9.73 | **4.98** | 0.03 | 2.13 | **3.27** |
| **Emuru** | 13.89 | **6.19** | **1.30** | 5.36 | 0.14 | **1.87** | 38.27 |

Table 3. Comparison on the word-level and line-level IAM datasets between Emuru and State-of-the-Art approaches trained on IAM. KID, BKID, and ΔI-LPIPS are multiplied by $10^3$ and the best performance is in bold for readability.

| | CVL | | | | | | |
|---|---|---|---|---|---|---|---|
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **TS-GAN** | 42.12 | 31.97 | 43.15 | 32.44 | 0.13 | 3.07 | 2.02 |
| **HiGAN+** | 78.44 | 39.47 | 80.39 | 36.50 | 0.12 | 3.07 | 53.91 |
| **HWT** | 31.22 | 16.73 | 26.14 | 14.44 | 0.38 | 2.59 | 10.23 |
| **VATr** | 34.40 | 24.64 | 32.21 | 25.01 | 0.06 | 2.36 | 8.77 |
| **VATr++** | 35.53 | 19.87 | 34.15 | 16.08 | 0.12 | 2.18 | 13.30 |
| **One-DM** | 60.45 | 26.58 | 64.13 | 26.76 | 0.06 | 2.66 | 88.94 |
| **DiffPen** | 40.40 | 17.50 | 38.21 | 18.30 | **0.01** | 2.99 | 51.58 |
| **Emuru** | **14.39** | **10.77** | **12.34** | **10.54** | 0.13 | **1.82** | **0.75** |
| | **RIMES** | | | | | | |
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **TS-GAN** | 109.04 | 36.39 | 132.90 | 41.64 | 0.12 | 3.26 | 93.87 |
| **HiGAN+** | 160.57 | 47.38 | 183.82 | 46.23 | 0.14 | 3.39 | 48.48 |
| **HWT** | 118.21 | 35.26 | 128.66 | 35.60 | 0.45 | 3.36 | **1.89** |
| **VATr** | 113.76 | 30.21 | 114.21 | 27.88 | 0.07 | 3.09 | 7.88 |
| **VATr++** | 110.04 | 35.61 | 104.05 | 31.90 | 0.10 | 2.83 | 26.84 |
| **One-DM** | 121.18 | 36.07 | 121.67 | 34.68 | 0.20 | 3.36 | 86.49 |
| **DiffPen** | 89.79 | 18.25 | 94.78 | 18.49 | **0.04** | 2.58 | 78.41 |
| **Emuru** | **26.93** | **13.36** | **21.19** | **9.40** | 0.25 | **2.18** | 47.96 |

Table 4. Comparison on the line-level CVL and RIMES datasets between Emuru and State-of-the-Art approaches trained on IAM. KID, BKID, and ΔI-LPIPS are multiplied by $10^3$ and the best performance is in bold for readability.

| | Karaoke Handwritten | | | | | | |
|---|---|---|---|---|---|---|---|
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **TS-GAN** | 60.30 | 12.68 | 64.80 | 6.75 | 0.05 | 4.59 | 75.30 |
| **HiGAN+** | 125.75 | 69.41 | 136.75 | 72.14 | 0.04 | 4.90 | 49.49 |
| **HWT** | 62.69 | 43.03 | 59.35 | 43.98 | 0.34 | 4.50 | 33.89 |
| **VATr** | 72.22 | 47.66 | 67.70 | 46.23 | 0.04 | 3.89 | 60.46 |
| **VATr++** | 67.16 | 46.53 | 58.57 | 42.59 | 0.01 | 3.96 | 84.57 |
| **One-DM** | 59.73 | 38.30 | 56.55 | 37.93 | 0.24 | 4.31 | 47.65 |
| **DiffPen** | 34.19 | 25.78 | 28.91 | 24.03 | 0.16 | 4.18 | 33.33 |
| **Emuru** | **13.87** | **7.99** | **9.24** | **5.37** | **0.13** | **2.24** | **0.73** |
| | **Karaoke Typewritten** | | | | | | |
| | **FID↓** | **BFID↓** | **KID↓** | **BKID↓** | **ΔCER↓** | **HWD↓** | **ΔI-LPIPS↓** |
| **TS-GAN** | 141.41 | 75.78 | 157.33 | 80.14 | 0.02 | 4.70 | 235.45 |
| **HiGAN+** | 135.34 | 63.39 | 146.34 | 65.81 | 0.03 | 5.19 | 85.45 |
| **HWT** | 72.78 | 37.40 | 62.77 | 31.51 | 0.39 | 4.57 | 138.76 |
| **VATr** | 80.38 | 41.02 | 70.46 | 37.26 | 0.04 | 4.14 | 115.80 |
| **VATr++** | 76.03 | 41.69 | 63.17 | 36.50 | **0.01** | 4.15 | 89.74 |
| **One-DM** | 70.75 | 44.06 | 60.90 | 42.78 | 0.25 | 4.80 | 119.63 |
| **DiffPen** | 78.07 | 61.16 | 67.17 | 61.05 | 0.14 | 4.71 | 187.68 |
| **Emuru** | **9.85** | **4.33** | **5.60** | **1.24** | 0.11 | **1.28** | **5.07** |

Table 5. Comparison on the line-level Karaoke dataset, separating calligraphy and typewritten styles, between Emuru and State-of-the-Art approaches trained on IAM. KID, BKID, and ΔI-LPIPS are multiplied by $10^3$ and the best performance is in bold for readability.


Figure 6. Further qualitative results on IAM words.


Figure 7. Further qualitative results on IAM lines.


Figure 8. Further qualitative results on CVL lines.


Figure 9. Further qualitative results on RIMES lines.

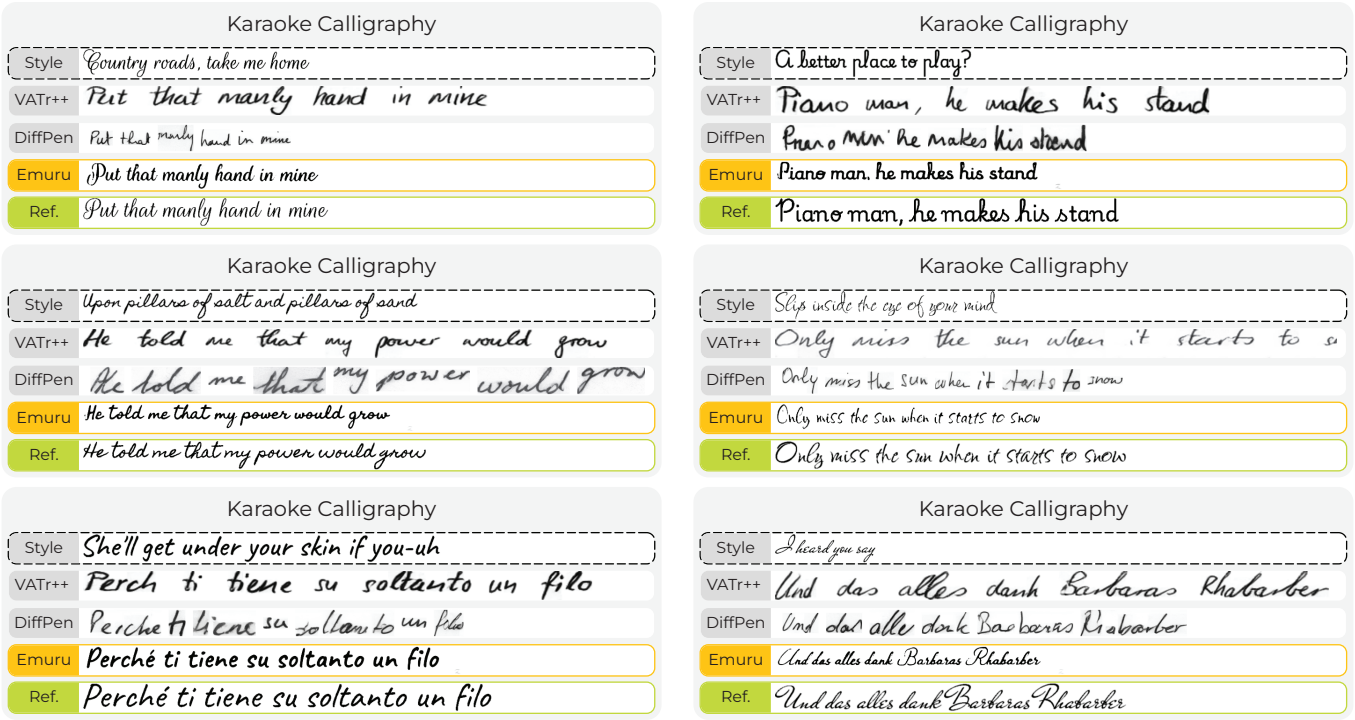


Figure 10. Further qualitative results on Karaoke Calligraphy.

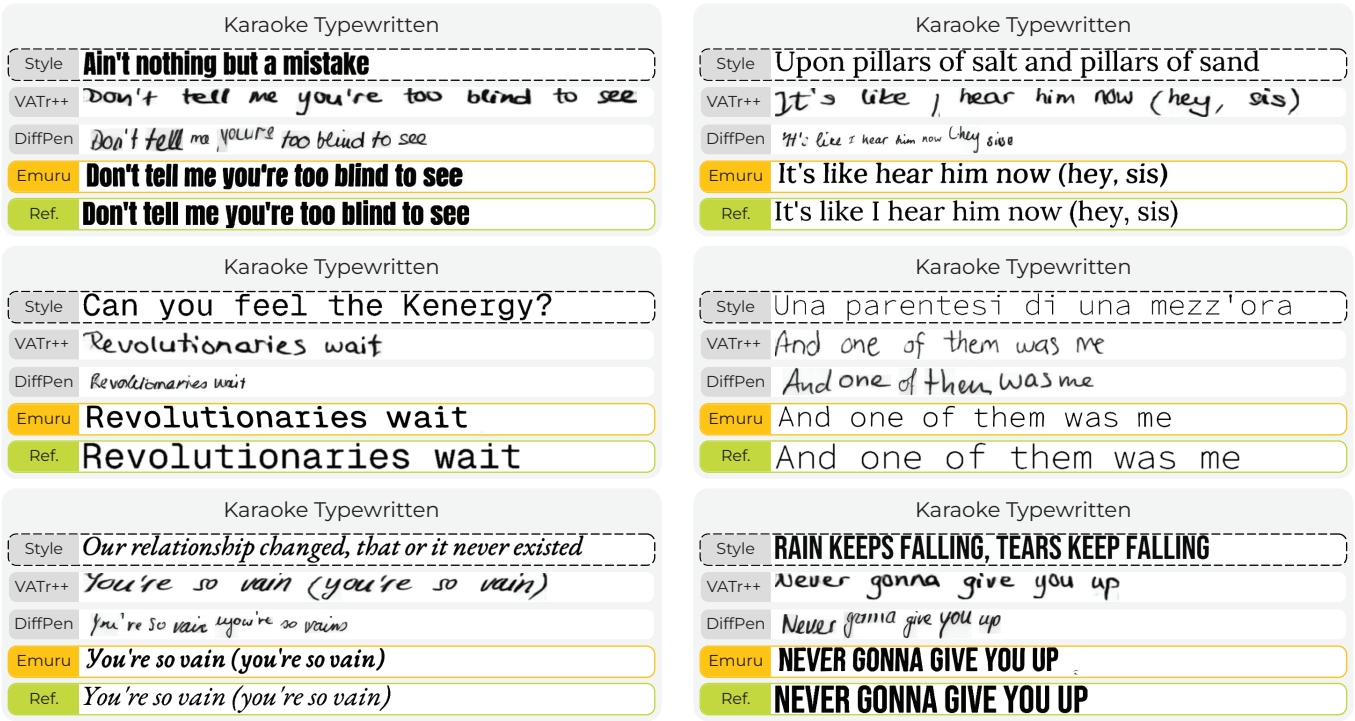


Figure 11. Further qualitative results on Karaoke Typewritten.

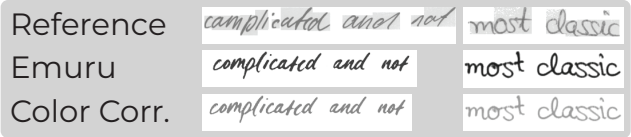


Figure 12. Thanks to the characteristics of the VAE, ink color can be easily corrected on Emuru-generated images to look more similar to the color in the reference image.

## J. Editing Application

Due to our Emuru VAE, there can be a difference in ink color between the reference image, $I_{style}$, and the Emuru-generated one, $I_{out}$. However, this difference can be easily recovered. Note that, once passed through the VAE, both $I_{style}$ and $I_{out}$ have a perfectly white background. Therefore, the background can be easily removed from both images via standard chroma keying on white. Then, the average color of the ink pixels can be computed from the reference image and directly applied to the ink pixels in the generated image (see Figure 12).

## References


[1] Emmanuel Augustin, Matthieu Carré, Emmanuèle Grosicki, J-M Brodin, Edouard Geoffrois, and Françoise Prêteux. RIMES evaluation campaign for handwritten mail processing. In *IWFHR*, 2006. 2

[2] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018. 3

[3] Gang Dai, Yifan Zhang, Quhui Ke, Qiangya Guo, and Shuangping Huang. One-DM: One-Shot Diffusion Mimicker for Handwritten Text Generation. In *ECCV*, 2024. 3

[4] Ji Gan and Weiqiang Wang. HiGAN: Handwriting Imitation Conditioned on Arbitrary-Length Texts and Disentangled Styles. In *AAAI*, 2021.

[5] Ji Gan, Weiqiang Wang, Jiaxu Leng, and Xinbo Gao. HiGAN+: Handwriting Imitation GAN with Disentangled Representations. *ACM Trans. Graphics*, pages 1–17, 2022. 3

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, 2017. 3

[7] Lei Kang, Pau Riba, Yaxing Wang, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. GANwriting: Content-Conditioned Generation of Styled Handwritten Word Images. In *ECCV*, 2020. 2

[8] Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. CVL-DataBase: An Off-Line Database for Writer Retrieval, Writer Identification and Word Spotting. In *ICDAR*, 2013. 2

[9] Praveen Krishnan, Rama Kovvuri, Guan Pang, Boris Vassilev, and Tal Hassner. TextStyleBrush: Transfer of Text Aesthetics from a Single Example. *IEEE Trans. PAMI*, 2023. 3

[10] Minghao Li, Tengchao Lv, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. TrOCR: Transformer-based optical character recognition with pre-trained models. *AAAI*, 2023. 3

[11] U-V Marti and Horst Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *IJDAR*, pages 39–46, 2002. 2

[12] Martin Mayr, Marcel Dreier, Florian Kordon, Mathias Seuret, Jochen Zöllner, Fei Wu, Andreas Maier, and Vincent Christlein. Zero-Shot Paragraph-level Handwriting Imitation with Latent Diffusion Models. *arXiv preprint arXiv:2409.00786*, 2024. 3

[13] Konstantina Nikolaidou, George Retsinas, Giorgos Sfikas, and Marcus Liwicki. DiffusionPen: Towards Controlling the Style of Handwritten Text Generation. *ECCV*, 2024. 6

[14] Konstantina Nikolaidou, George Retsinas, Giorgos Sfikas, and Marcus Liwicki. Rethinking HTG Evaluation: Bridging Generation and Recognition. *Proceedings of the European Conference on Computer Vision Workshops*, 2024. 3

[15] Vittorio Pippi, Fabio Quattrini, Silvia Cascianelli, and Rita Cucchiara. HWD: A Novel Evaluation Score for Styled Handwritten Text Generation. In *BMVC*, 2023. 3

[16] Bram Vanherle, Vittorio Pippi, Silvia Cascianelli, Nick Michiels, Frank Van Reeth, and Rita Cucchiara. VATr++: Choose Your Words Wisely for Handwritten Text Generation. *IEEE Trans. PAMI*, 2024. 3, 6

[17] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018. 3