Implicit vs. Explicit Style Transfer? A Comparison of GAN Architectures for Continuous Path Keyboard Input Modeling

Akash Mehra, Jerome R. Bellegarda, Ojas Bapat,
Hema Koppula, Rick Chang, Ashish Shrivastava, Oncel Tuzel
Apple
Cupertino, California 95014, USA
{akash_mehra, jerome, obapat, hkoppula, jenhao_chang, ashish.s, otuzel}@apple.com

***Abstract*—The success of continuous path keyboard input as an alternative text input modality requires high-quality training data to inform the underlying recognition model. In [1], we have adopted generative adversarial networks (GANs) to augment the training corpus with synthetic user-realistic paths. GAN-driven synthesis makes it possible to emulate the acquisition of enough paths from enough users to learn a model sufficiently robust across a large population. The present work studies the influence of different GAN architectures on path quality and diversity. Experiments show that explicit content/ style disentanglement resulting from separate style encoding has only a limited impact on end user perception. On the other hand, implicit and explicit style transfer paradigms are complementary in the kind of user-realistic artifacts they generate. Leveraging multiple GAN strategies thus injects more robustness into the model through broader coverage of user idiosyncrasies across a wide lexical range.**

***Index Terms*—Continuous path recognition, generative adversarial networks, style transfer, embedded devices**

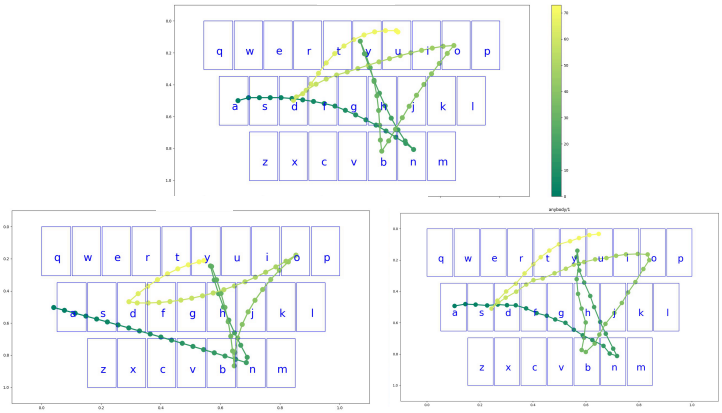

Fig. 1. *Path visualization for input word "**anybody**"—color changes from green to yellow with time. Typical user path (top), programmatically generated synthetic path (bottom left), and GAN-generated synthetic path (bottom right).*

I. Introduction

Entering text on a mobile device using continuous path input involves sliding the finger in a single continuous motion across successive keys on the keyboard until the intended word is complete [2]. After users gain proficiency with this alternative modality, they often find entering words easier and faster [3], [4]. As in standard tapping, recognition relies on statistical pattern matching enhanced with linguistic knowledge in order to predict the intended word [5].

Continuous path keyboard input has higher inherent ambiguity than tapping, because the path trace may exhibit not just local overshoots/undershoots, but also, depending on the user, substantial mid-path excursions. Deploying a robust solution requires a large amount of high-quality training data, which is difficult/expensive to collect and annotate. This situation has sparked interest into synthetic paths that could be used as proxies for real user-generated paths. In [6], for example, the authors programmatically generated plausible-looking paths by connecting the characters within a word using an algorithm that minimizes jerk [7], an approach inspired by human motor control theory (cf. [8], [9]). Typically, the parameters of such synthesis algorithms are tuned manually until generated paths look "similar enough" to real user paths (based on human judgments of a small number of paths [10]).

While credible, the ensuing paths are inherently restricted in their expressiveness and, as a result, do not fully capture the variability of user paths. To illustrate, Fig. 1 shows a typical user path (top) and synthetic path (bottom left) for the word "**anybody**." To synthetize more user-realistic training data, we have recently proposed [1] the adoption of generative adversarial networks (GANs) [11], [12]. We cast the problem as an instance of style transfer [13], where an initial synthetic path produced with simple cubic splines [8] is transformed to conform to user idiosyncrasies observed across a set of real user paths. The kind of path that results is illustrated at the bottom right of Fig. 1. GAN generation tends to more faithfully render human-like artifacts, resulting in better proxies for real user-generated paths.

In this paper, we compare the simple type of style transfer in [1] with a more explicit strategy seeking to disentangle the various factors influencing style. Of particular interest is the impact of content/style disentanglement on the quality and diversity of the generated paths, as measured downstream by path recognition accuracy at inference time. In the next two

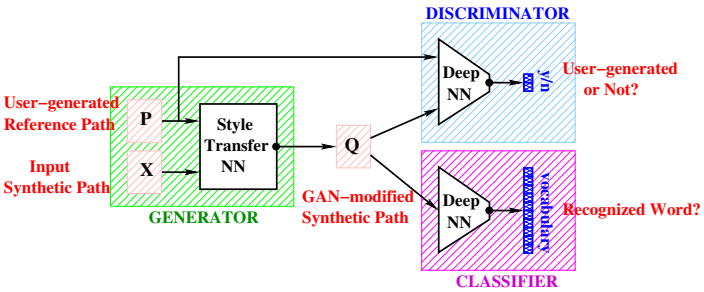

Fig. 2. *Path generation using style transfer GAN architecture of [1] (with implicit style representation).*

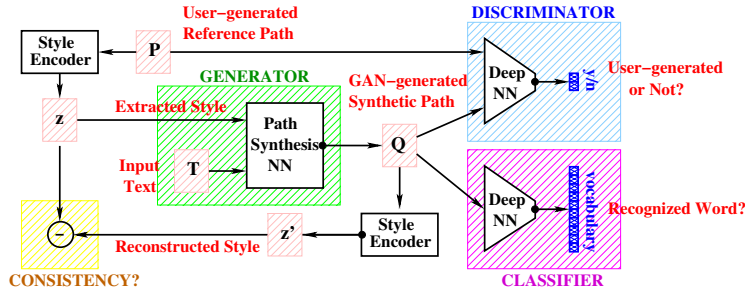

Fig. 3. *Path generation using GAN architecture with explicit style extraction and reconstruction.*

sections, we describe the two associated GAN architectures. In Sections IV and V, we discuss experimental conditions and results observed. Section VI summarizes the insights gained.

II. Implicit Style Transfer

The path synthesis approach adopted in [1] was inspired from image style transfer (cf. [14]–[16]), where the basic idea is to bias the generative model in GANs according to the desired style of image [17]. This led to the architecture illustrated in Fig. 2, where (given the inherent sequential nature of paths) both generative and discriminative models are realized via sequential neural networks (bi-LSTMs).

In Fig. 2, an initial (programmatically generated) input path X is transformed into a more realistic synthetic path Q based on a set of user-generated reference paths P, which are *collectively* representative of a *range* of observed user idiosyncrasies and/or styles. Because there is no explicit representation of such artifacts at the individual user level, the notion of style remains diffuse across the entire reference corpus, hence the terminology "implicit style transfer."

In standard multi-task fashion, every transformed path Q is also passed to a classifier which verifies that the generated path is still associated with the intended word. The network is then more likely to abstract out those discriminative elements of user generated paths that are most relevant to the current word. A suitable objective function to train the model of Fig. 2 is the typical linear interpolation of the usual GAN adversarial loss $\mathcal{K}(\mathcal{D},\mathcal{G})$ and the classification loss $\mathcal{L}(\mathcal{C})$ (where $\mathcal{D}$, $\mathcal{G}$, and $\mathcal{C}$ refer to the discriminator, generator, and classifier, respectively). In [1], we used the Connectionist Temporal Classification (CTC) loss [18] for the classifier.

Not shown on Fig. 2 is the possibility to treat the generative style transfer model as a sequence-to-sequence model (with attention) comprising an encoder and a decoder, which can expose a bottleneck layer representing the path embedding. In that case, the interpolated objective function should also include a path reconstruction loss $\mathcal{J}(\mathcal{R})$ (where $\mathcal{R}$ refers to the decoder of the sequence-to-sequence model). Such mininum mean square error (MMSE) loss is especially beneficial to assess the path distortion incurred on user-generated paths during the encoding process.

After multi-task adversarial training is complete, the discriminative model has learned to take into account user idiosyncrasies observed in the reference corpus, so the generated path Q ends up encapsulating the desired range of user behaviors, while still preserving the main characteristics of the input content X. Such realistic rendering was a major factor in the improved results we reported in [1].

III. Explicit Style Transfer

In the path synthesis approach of Fig. 2, there is no way to enforce specific constraints on path diversity or artifact coverage, both of which are largely governed by the range of behaviors observed in the set of user-generated reference paths. For more flexibility, it is desirable to explicitly specify the notion of style at the individual user level (hence the terminology "explicit style transfer"). Again, we can draw inspiration from the image synthesis literature, and more particularly the architecture known as StyleGAN [19].

The authors of [19] proposed to encode individual image "styles" into a distinct latent space ("style embedding space"), free from the restriction of following the probability density of the training data, and therefore more amenable to a systematic disentanglement of the various factors of variation. In the context of path generation, this leads to the architecture illustrated in Fig. 3, where again all models are realized with sequential neural networks.

In Fig. 3, every user-generated path P is associated with a style embedding z for that path: a point in the latent space of individual styles. That extracted style is then used to control the style of the paths generated by a text-to-path generator. In contrast with Fig. 2, programmatically generated paths X are no longer needed: instead, we now leverage the input text T directly. The steps to generate a path for T in the same style as P essentially parallel the StyleGAN approach of [19]: namely, a style encoder extracts the style embedding z from P, a content attention module provides a focused context of the input text T, and both the style embedding and the attended context are passed as input to the generator module to synthesize the resulting path Q. As before, the GAN-generated path Q is also passed to the classifier to ensure that it is indeed associated with the correct intended text T.

Explicit style encoding makes it possible to check the extent to which the style of the generated path Q conforms to the style of the reference path P. That is the purpose of the feedback loop in Fig. 3: the style encoder extracts

a style embedding z' from the GAN-generated path Q, and a consistency loss ensures that z' is suitably close to the original style z. With the addition of such MMSE loss $\mathcal{M}(\mathcal{S})$ to enforce consistency in the style embedding space $\mathcal{S}$, an interpolated objective function similar to the one in the previous section can be used to train the model. As before, it is equally possible to treat the generator as a sequence-to-sequence model exposing the path embedding, in which case a similar path reconstruction loss should be added to assess the path distortion incurred on user-generated paths during encoding.

After multi-task adversarial training is complete, the generative model has learned a disentangled representation of style and content, making it possible to generate synthetic paths Q for arbitrary input text T in any number of known user styles. In addition, the model has the ability to generate new unseen styles by interpolating between known user styles in the style embedding space $\mathcal{S}$.

IV. Testing Conditions

While the goal of [1] was to demonstrate that GANs could support training of a more robust recognition model, in this paper we want to compare the impact of different style transfer paradigms on the end user experience, as measured downstream by path recognition accuracy at inference time. Thus testing conditions need to change. In [1], we leveraged a balanced test corpus comprising the same number of user paths for each of the approximately 25k test words considered, which formed a strict subset of the set of words seen during training. Such balance was necessary to best characterize modeling performance across different training compositions. We also deliberately ignored language model rescoring since every word was recognized in isolation.

In constrast, for this paper we collected a separate corpus of about 100k user paths associated with actual user content generated by consenting participants in their normal usage of messaging and social media apps. Thus, words now appear according to their natural frequency distribution in the language. This offers the opportunity to measure Top-1 recognition accuracy after taking into account the influence of the language model, thus capturing overall end user impact. By construction, this test set still spans approximately the same number of words (25k), but this time it includes words not seen during training, such as rare names, acronyms, and abbreviations: for example, "*NBA*" and "*btw*".

Training conditions were closely aligned with [1]. We relied on the set of 2.2M user-generated paths referenced in [1] as **U2**, which covers 55k English words collected from a diversity of users in a variety of conditions. Specifically, 665 users (roughly half males, half females) ranging in age from 18 to 70 years produced paths on 6 layouts with various screen sizes. Thus each participant generated on average 3300 paths. Approximately half of the paths were generated using the thumb finger, with the other half generated with the index finger. In line with [20], the participants were chosen to attain a proportion of left-handed users of about 20%.

As in [1], we then leveraged the reference paths from **U2** within the architecture of Fig. 2 to perform implicit style transfer from initial synthetic paths obtained via cubic splines [8]. This led to the generation of a set **G2** of 2.2M GAN-modified paths. Finally, we performed explicit style transfer using the architecture of Fig. 3, again using reference paths from **U2**. This led to a comparable set **H2** of 2.2M style-disentangled GAN-generated paths. Neither set contained data generated for words outside the 55k inventory.

For comparison, we also trained on a larger set **U5** of 5M user-generated paths (comprising the 2.2M original user paths). In addition, with the help of a wide-coverage English lexicon, we generated an alternative set **H2*** of 2.2M style-disentangled paths, this time designed to cover words both inside and outside the 55k inventory, thus providing training data for words for which no user data is available.

V. Experimental Results

The results of our experiments are summarized in Table I. We report Top-1 recognition accuracies for the entire test set (column "All Words"), as well as separate Top-1 accuracies for the subset of paths associated with the 55k words seen in training (column "55k-Words"). Comparing the latter accuracies ($\approx$ 95%) with results reported in [1] ($\approx$ 65%) reflects the considerable influence of the language model.

Clearly, bringing to bear linguistic resources also reduces the differences observed in [1] between different training compositions, to the point that they become harder to perceive for the end user. In practice, linguistic constraints act as a counterbalance to low diversity in the original data collection, regardless of the GAN ability to capture relevant user artifacts. Still, GAN generation seems to help in modeling words for which no user data is available. Future work will further investigate the matter on low-resource languages, for which zero- or few-shot learning is required.

Table I also shows that, after LM, the two types of GANs do not yield material differences. Thus, the degree of content/style disentanglement that can be achieved with explicit style encoding has only a limited impact on end user perception. Close examination of individual paths, however, reveals some complementarity in the kind of user artifacts generated by implicit and explicit style transfer. Through a systematic analysis of error differences obtained across training compositions, we observed that GAN augmentation involving both **G2** and **H2**

TABLE I
Recognition results for different training compositions, on 104,057 test paths spanning 24,926 words.

Training Composition	Top-1 Accuracy After LM	
	All Words	55k-Words
U2	91.9 %	94.8 %
U2+G2	91.6 %	94.6 %
U2+H2	91.7 %	94.5 %
U2+G2+H2	91.8 %	94.6 %
U5	92.1 %	94.9 %
U5+H2*	92.4 %	94.9 %

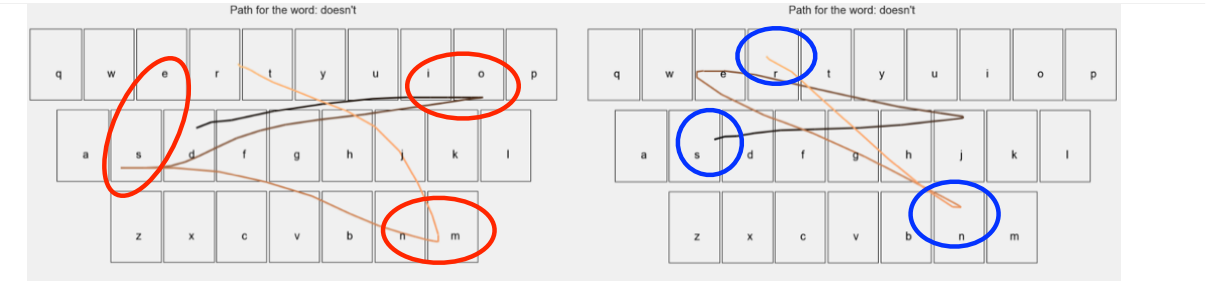

Fig. 4. *Two observed test paths with markedly different styles for input word "**doesn't**".*

achieves broader coverage of user idiosyncrasies than either of them alone. This greater robustness can perhaps be traced to the propensity of different GAN frameworks to encapsulate rarely observed user artifacts in various proportion to their actual frequency. This observation is especially salient with smaller sets of user-generated paths.

Figs. 4–6 illustrate this complementarity using two different renditions of the word "**doesn't**" extracted from the test set, which both led to an error with **U2** training alone. The left path in Fig. 4 was correctly recognized with both **U2+G2** and **U2+G2+H2** trainings, largely because in this case augmenting the training corpus with **G2** makes the system more resilient to the sharp angle in "'**o**", omission of "**e**", and overshoot of "**n**" (red ellipses). In contrast, the right path in Fig. 4 was correctly recognized with both **U2+H2** and **U2+G2+H2** trainings, because in this case augmenting the training corpus with **H2** makes the system more resilient to the overshoots of "**d**" and "**t**", and the loopy style of "**n**" (blue ellipses).

Such resilience is readily explained by looking at how the two different GAN augmentation strategies cover these various idiosyncrasies. Fig. 5 depicts four training exemplars extracted from **G2**, featuring sharp angles in "'**o**", omissions of either "**e**" or "**s**" and near overshoots of "**n**" (red ellipses). In contrast, Fig. 6 depicts four training exemplars extracted from **H2**, featuring consistent overshoots of "**d**" and instances of loopy style—though not always on "**n**" (blue ellipses). This case study illustrates the combinatorial complexity of generating all possible artifacts at every point of every path for every possible word. Multiple GAN strategies inherently help render a greater diversity of artifacts.

VI. Conclusion

This paper has studied the influence of different multi-task adversarial architectures on the quality and diversity of GAN-generated paths, looking more particularly at the impact of implicit vs. explicit content/style disentanglement. We observed that after linguistic resources are applied, explicit style extraction does not confer material benefits over implicit style transfer. On the other hand, close examination of individual paths revealed that the two generative models are complementary in the kind of user-realistic artifacts that they are able to produce. Explicit style transfer allows finer control of local behavior, while implicit style transfer renders observed attributes more diffusely across paths. Such complementarity is especially beneficial for expanding lexical coverage by generating paths with a diverse spread of more effective artifacts. Leveraging both implicit and explicit style transfer can thus inject more robustness into the model through a broader coverage of user idiosyncrasies across a wide lexical range.

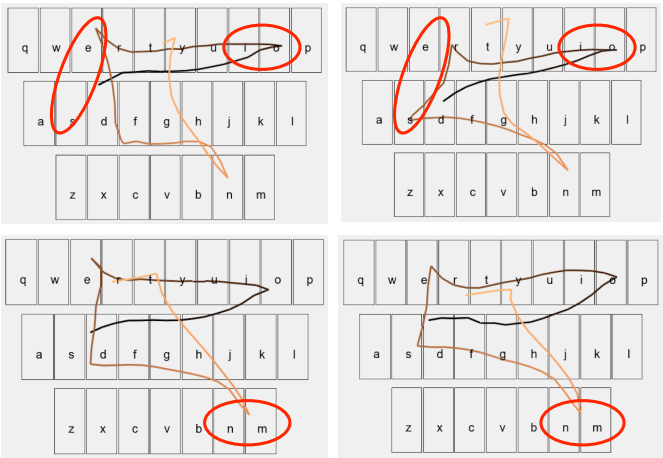

Fig. 5. *Explicit style transfer training instances (from **G2**).*

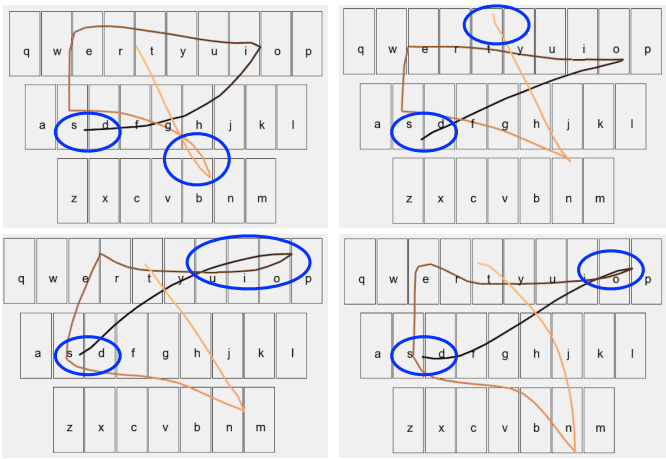

Fig. 6. *Implicit style transfer training instances (from **H2**).*

REFERENCES

[1] A. Mehra, J.R. Bellegarda, O. Bapat, P. Lal, and X. Wang, "Leveraging GANs to Improve Continuous Path Keyboard Input Models," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Barcelona, Spain, May 2020.

[2] S. Zhai and P.-O. Kristensson, "The Word-Gesture Keyboard: Reimagining Keyboard Interaction," in *Communications of the ACM*, Vol. 55, No. 9, pp. 91–101, 2012.

[3] S. Zhai and P.-O. Kristensson, "Shorthand Writing on Stylus Keyboard," in *Proc. ACM Conf. Special Interest Group on Computer–Human Interaction*, pp. 97–104, 2003.

[4] S. Reyal, S. Zhai, and P.-O. Kristensson, "Performance and User Experience of Touchscreen and Gesture Keyboards in a Lab Setting and in the Wild," in *Proc. 33rd Ann. ACM Conf. on Human Factors in Computing Systems*, pp. 679–688, 2015.

[5] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *IEEE Trans. Systems, Man, and Cybernetics*, Part C: Applications and Reviews, Vol. 37, No. 3, pp. 311–324, 2007.

[6] O. Alsharif, T. Ouyang, F. Beaufays, S. Zhai, T. Breuel, and J. Schalkwyk, "Long Short Term Memory Neural Network for Keyboard Gesture Decoding," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Brisbane, Australia, Sept. 2015.

[7] P. Quinn and S. Zhai, "Modeling Gesture–Typing Movements," *Human–Computer Interaction*, pp. 1–47, 2016.

[8] T. Flash and N. Hogan, "The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model," in *J. Neuroscience*, Vol. 5, pp. 1688–1703, 1985.

[9] J. Müller, A. Oulasvirta, and R. Murray-Smith, "Control Theoretic Models of Pointing," *ACM Trans. Computer–Human Interaction*, Vol. 24, No. 4, Aug. 2017.

[10] S. Zhai, J. Kong, and X. Ren, "Speed–Accuracy Tradeoff in Fitts' Law Tasks on the Equivalency of Actual and Nominal Pointing Precision," *Int. J. Human–Computer Studies*, Vol. 61, No. 6, pp. 823–856, 2004.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proc. Neural Information Processing Systems*, Dec. 2014.

[12] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," *arXiv:1606.03498v1*, Jun. 2016.

[13] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from Simulated and Unsupervised Images through Adversarial Training," in *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 2107–2116, 2017.

[14] L.A. Gatys, A.S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *arXiv:1508.06576v2*, Sept. 2015.

[15] J. Johnson, A. Alahi, and F.F. Li, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," *arXiv:1603.08155v1*, Mar. 2016.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros, "Image-to-Image Translation with Conditional Adversarial Nets," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2017.

[17] L.A. Gatys, A.S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

[18] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proc. ACM Int. Conf. Machine Learning*, pp. 369–376, 2006.

[19] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *arXiv:1812.04948v3*, March 2019.

[20] S. Azenkot and S. Zhai, "Touch Behavior with Different Postures on Soft Smartphone Keyboards," in *Proc. 14th ACM Int. Conf. Human-Computer Interaction with Mobile Devices and Services*, pp. 251–260, 2012.