

Generating Handwriting via Decoupled Style Descriptors



Atsunobu Kotani



Stefanie Tellex



James Tompkin



BROWN
Computer Science









Target

gether with the

young offende

earth, marit

Ours

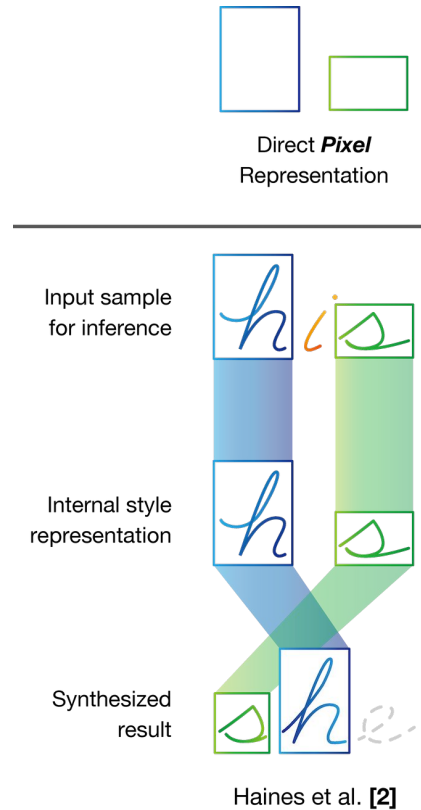
gether with the

young offende

earth marit

Handwriting generation:

- Pixel representation [2]

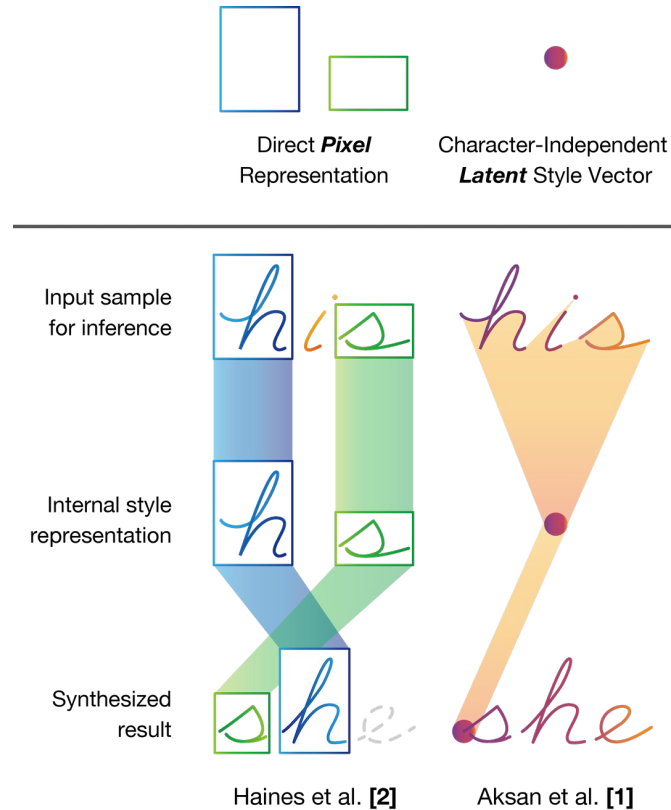


[1] Aksan, E., Pece, F., Hilliges, O.: DeepWriting: Making Digital Ink Editable via Deep Generative Modeling. SIGCHI (2018).

[2] Haines, T.S.F., Mac Aodha, O., Brostow, G.J.: My text in your handwriting. ACM Trans. Graph. 35(3) (May 2016).

Handwriting generation:

- Pixel representation [2]
- Learned by neural networks [1]



[1] Aksan, E., Pece, F., Hilliges, O.: DeepWriting: Making Digital Ink Editable via Deep Generative Modeling. SIGCHI (2018).

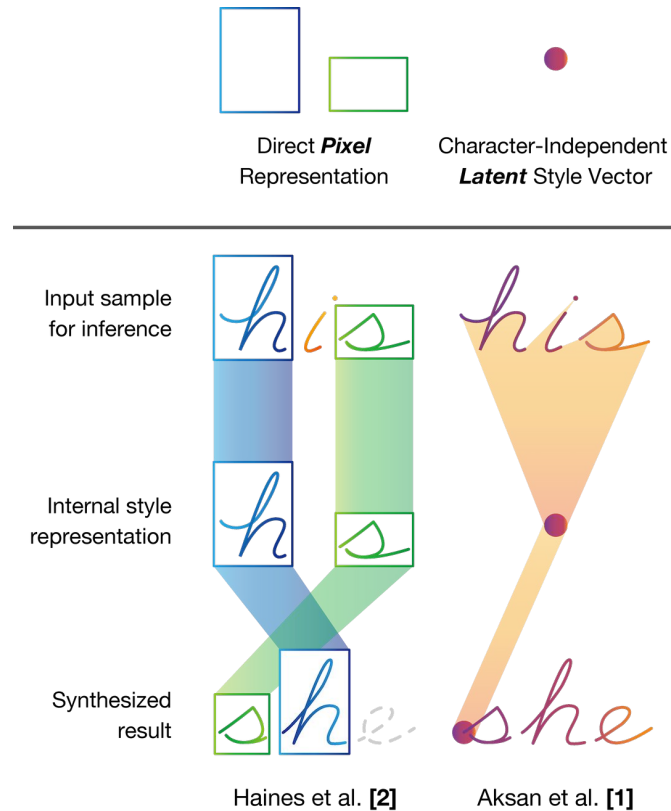
[2] Haines, T.S.F., Mac Aodha, O., Brostow, G.J.: My text in your handwriting. ACM Trans. Graph. 35(3) (May 2016).

Handwriting generation:

- Pixel representation [2]
- Learned by neural networks [1]

Limitations:

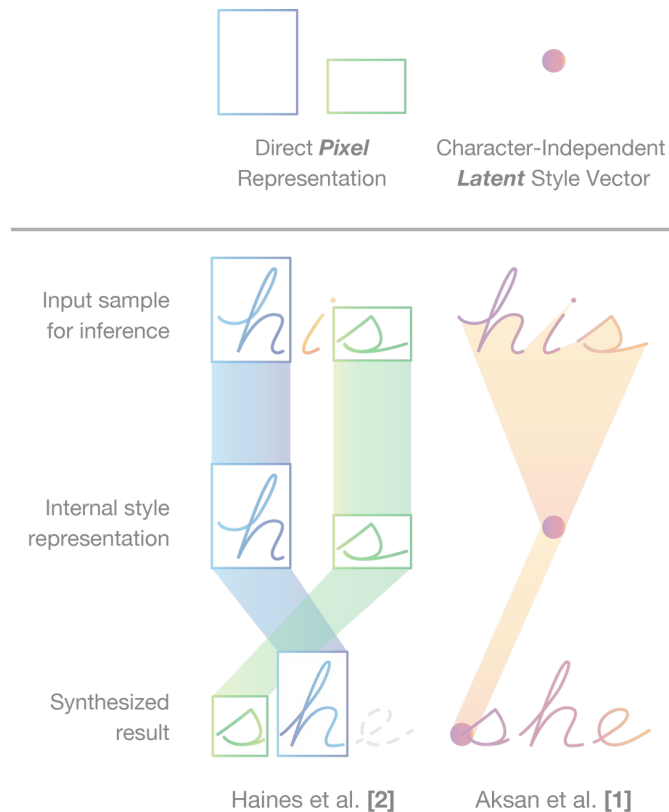
- Generating missing characters
- Generating fine details



[1] Aksan, E., Pece, F., Hilliges, O.: DeepWriting: Making Digital Ink Editable via Deep Generative Modeling. SIGCHI (2018).

[2] Haines, T.S.F., Mac Aodha, O., Brostow, G.J.: My text in your handwriting. ACM Trans. Graph. 35(3) (May 2016).

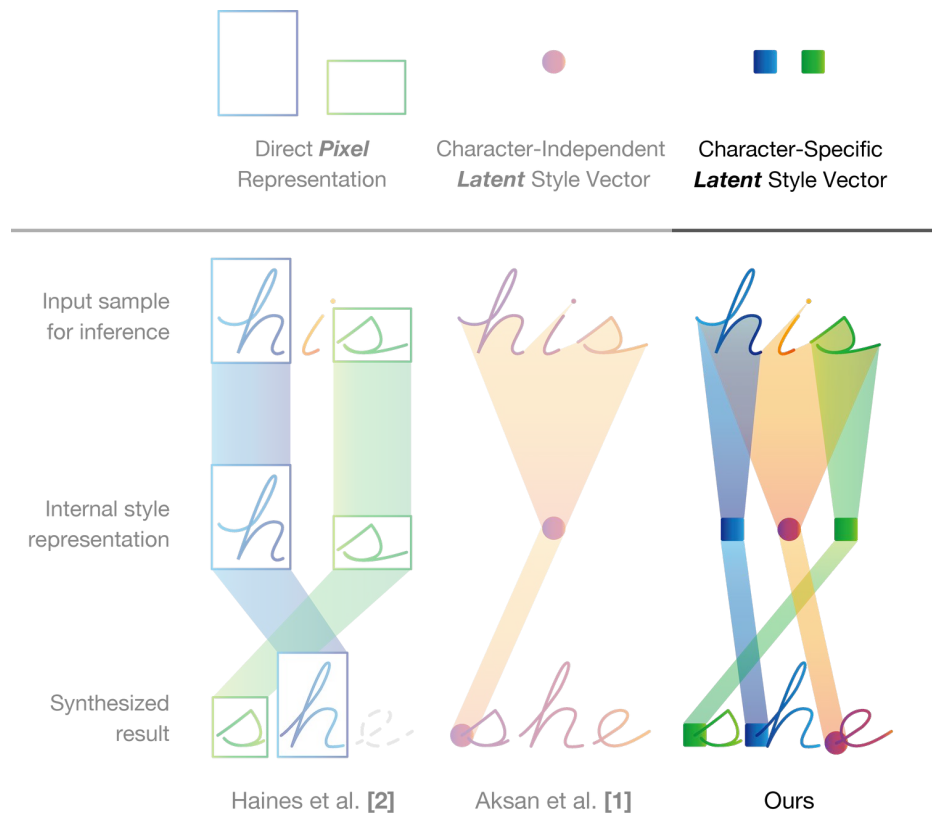
Underlying problem:
No explicit separation of writer style
from character style.



[1] Aksan, E., Pece, F., Hilliges, O.: DeepWriting: Making Digital Ink Editable via Deep Generative Modeling. SIGCHI (2018).

[2] Haines, T.S.F., Mac Aodha, O., Brostow, G.J.: My text in your handwriting. ACM Trans. Graph. 35(3) (May 2016).

We learn to decouple writer and character style into specific vectors.



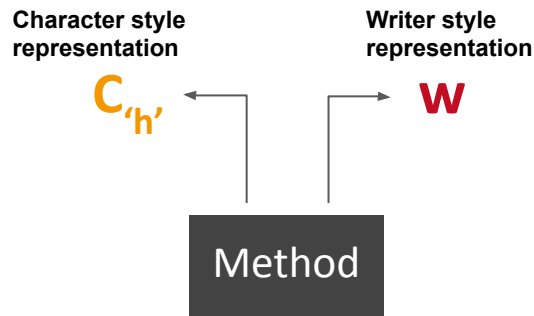
[1] Aksan, E., Pece, F., Hilliges, O.: DeepWriting: Making Digital Ink Editable via Deep Generative Modeling. SIGCHI (2018).

[2] Haines, T.S.F., Mac Aodha, O., Brostow, G.J.: My text in your handwriting. ACM Trans. Graph. 35(3) (May 2016).

Problem statement

Desired output data:

- *Writer-independent*
character style representation $C_{'h'}$
- *Character-independent*
writer style representation w



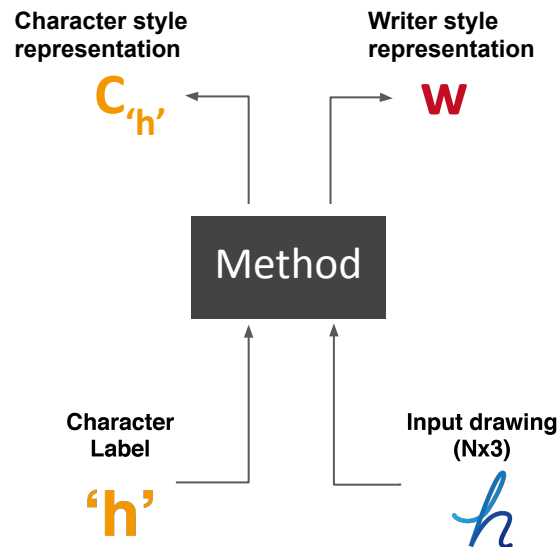
Problem statement

Desired output data:

- *Writer-independent*
character style representation $C_{\text{'h'}}$
- *Character-independent*
writer style representation w

Input data:

- Strokes as point sequences (x, y, t)
- Character labels as one-hot vectors



Decoupled Style Descriptors (DSD)

We learn a *linear* relationship between

- Writer-DSD w and
- Character-DSD C_h ,

Decoupled Style Descriptors (DSD)

We learn a *linear* relationship between

- Writer-DSD w and
- Character-DSD $C_{'h'}$

Character-DSD
(256x256 Matrix)

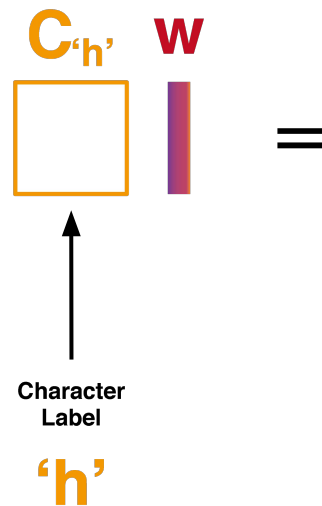


Decoupled Style Descriptors (DSD)

We learn a *linear* relationship between

- Writer-DSD w and
- Character-DSD $C_{'h'}$,

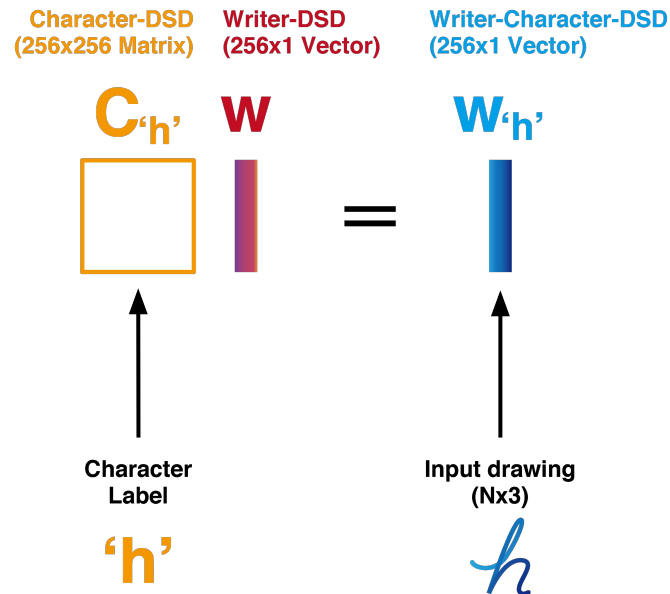
Character-DSD (256x256 Matrix) Writer-DSD (256x1 Vector)



Decoupled Style Descriptors (DSD)

We learn a *linear* relationship between

- Writer-DSD w and
- Character-DSD $C_{'h'}$

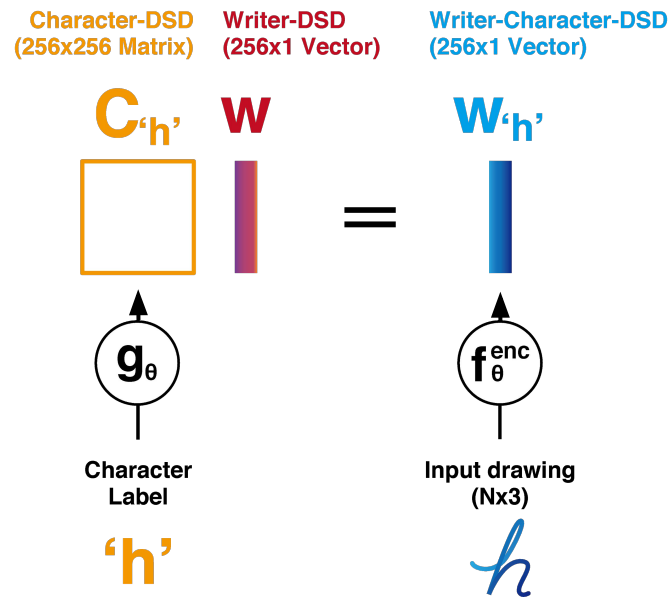


Decoupled Style Descriptors (DSD)

We learn a *linear* relationship between

- Writer-DSD w and
- Character-DSD $C_{h'}$,

through learned LSTM encoders g_{θ} and f_{θ} .



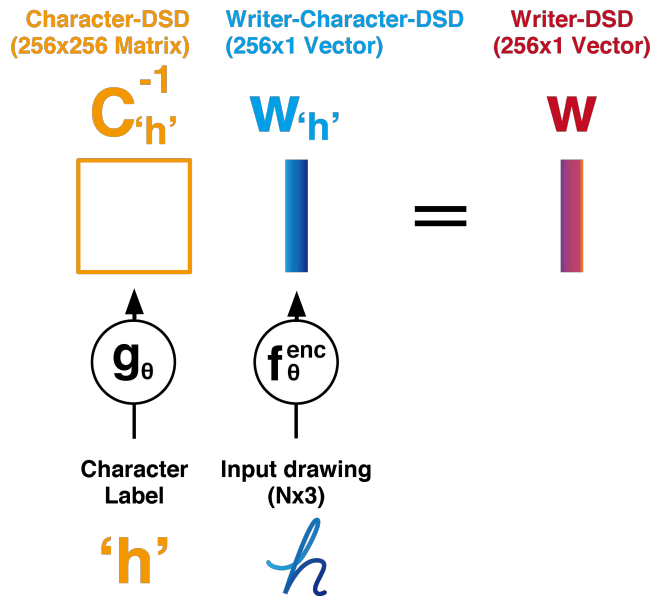
Decoupled Style Descriptors (DSD)

We learn a *linear* relationship between

- Writer-DSD w and
- Character-DSD $C_{h'}$,

through learned LSTM encoders g_{θ} and f_{θ} .

Simply invert $C_{h'}$ to recover w from $w_{h'}$.



Decoupled Style Descriptors (DSD)

We learn a *linear* relationship between

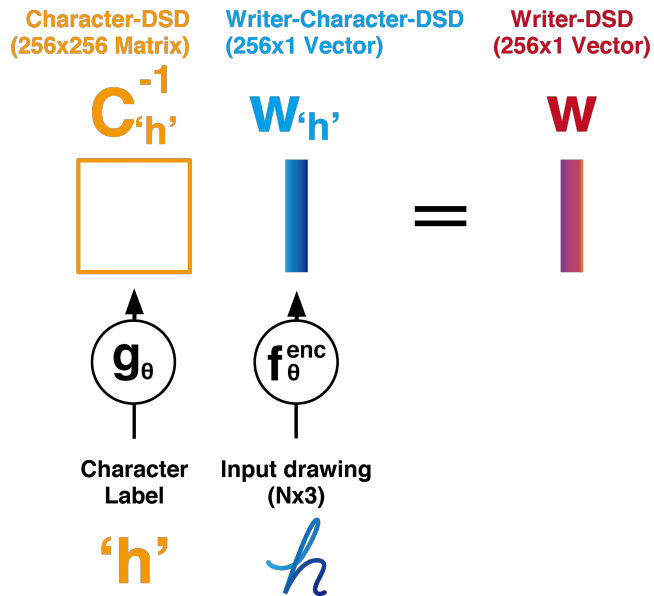
- Writer-DSD w and
- Character-DSD $C_{h'}$

through learned LSTM encoders g_{θ} and f_{θ} .

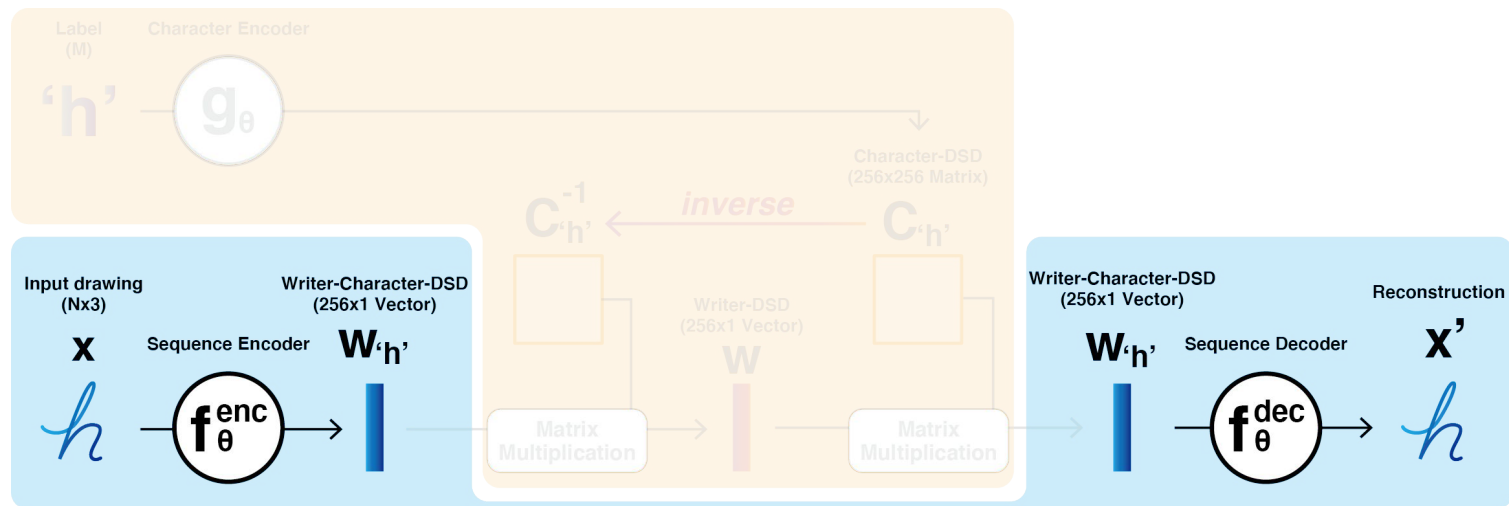
Simply invert $C_{h'}$ to recover w from $w_{h'}$.

Retains *more fine detail*.

Allows *few-shot learning* for new characters,
and *writer identification*.

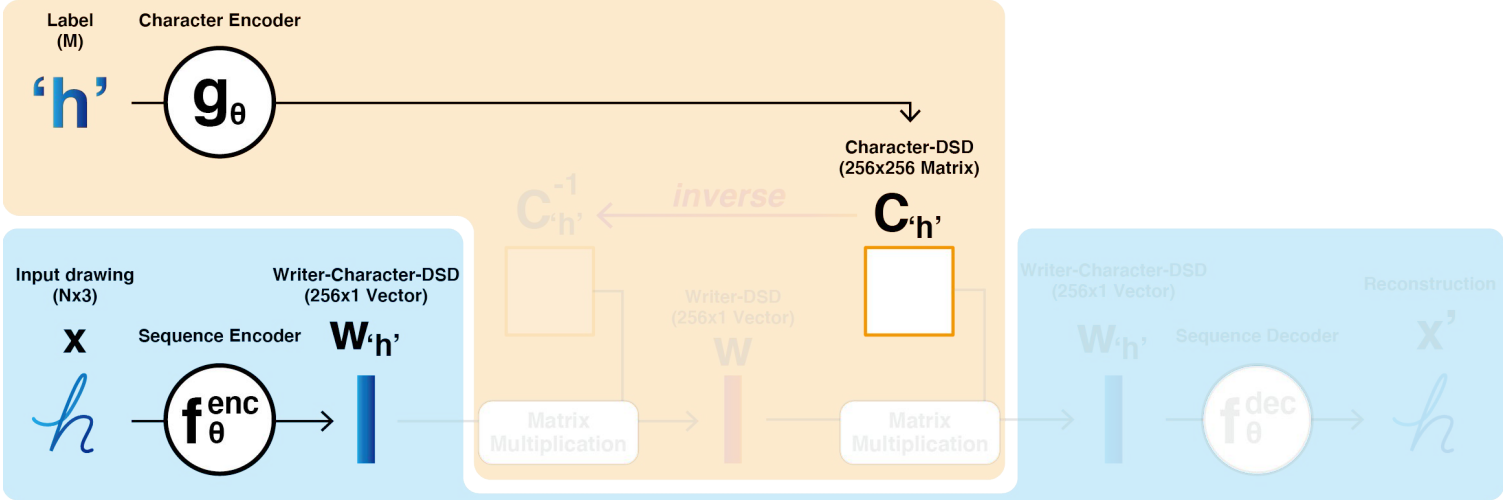


Decoupled Style Descriptors (DSD)

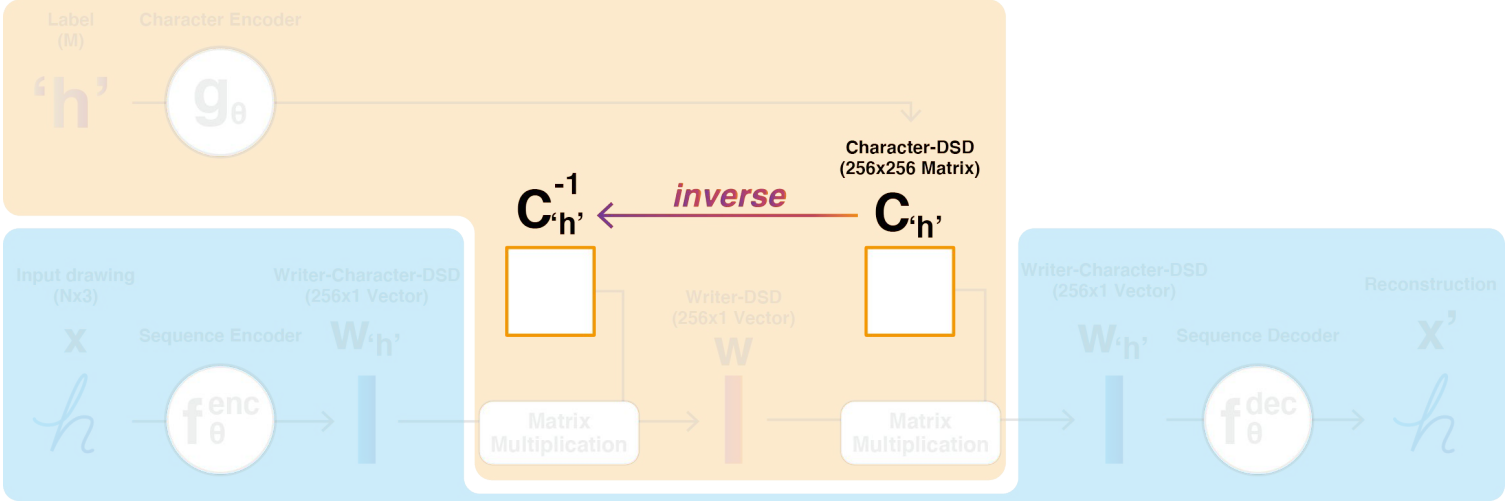


At the core, we have LSTM-based autoencoder, similar to the work by Graves [3].

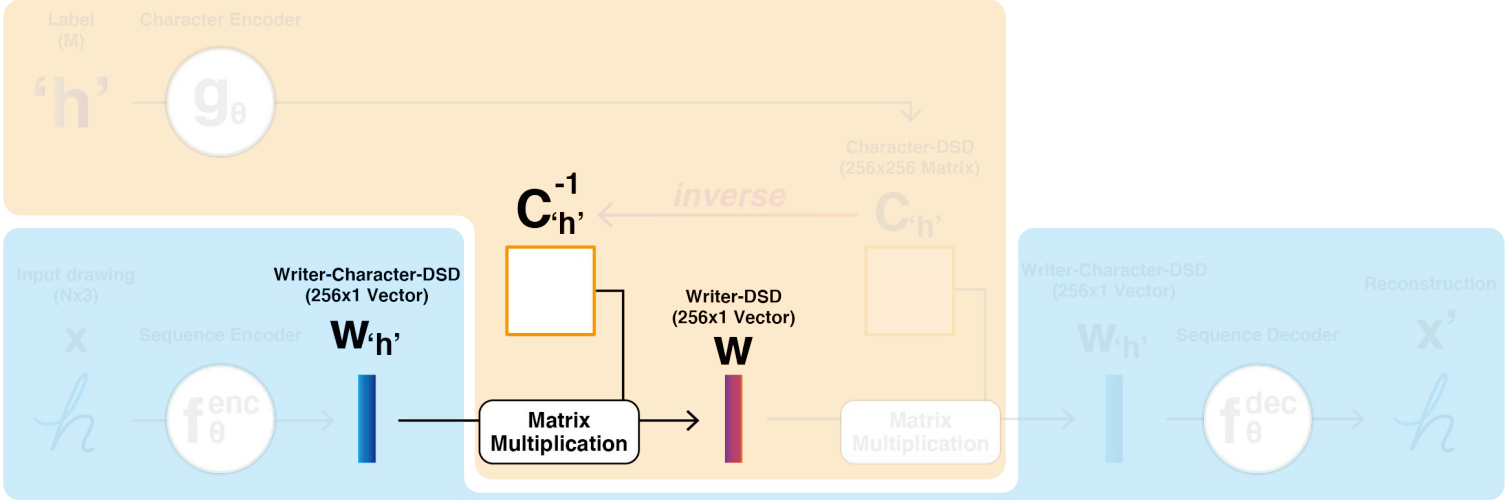
Decoupled Style Descriptors (DSD)



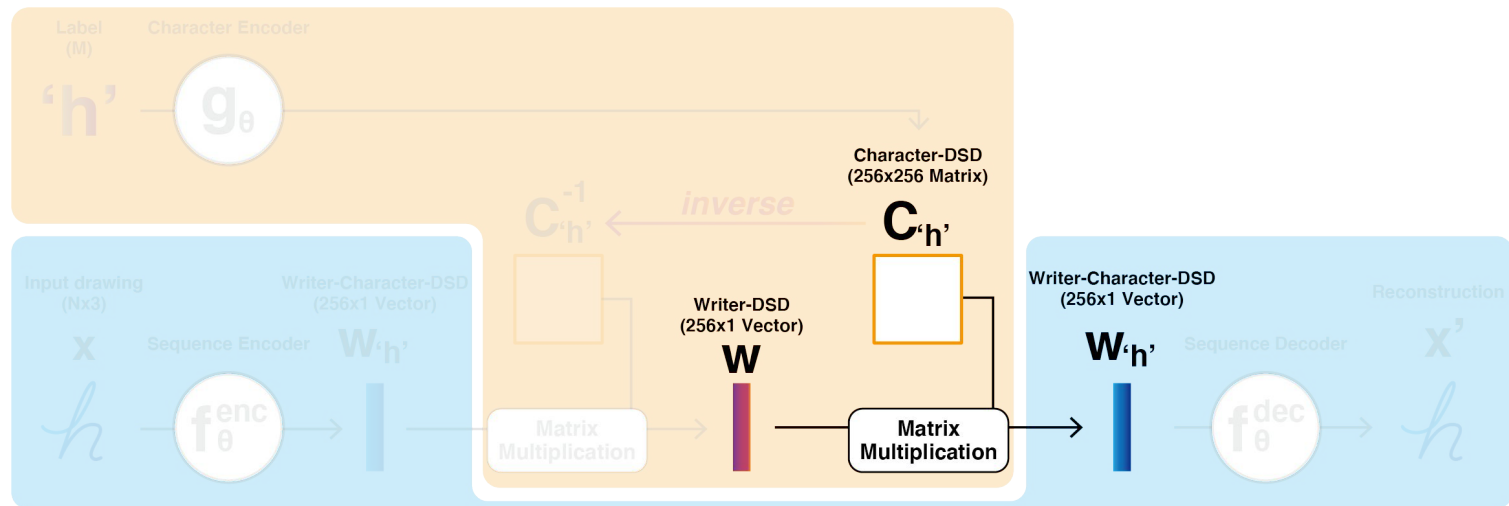
Decoupled Style Descriptors (DSD)



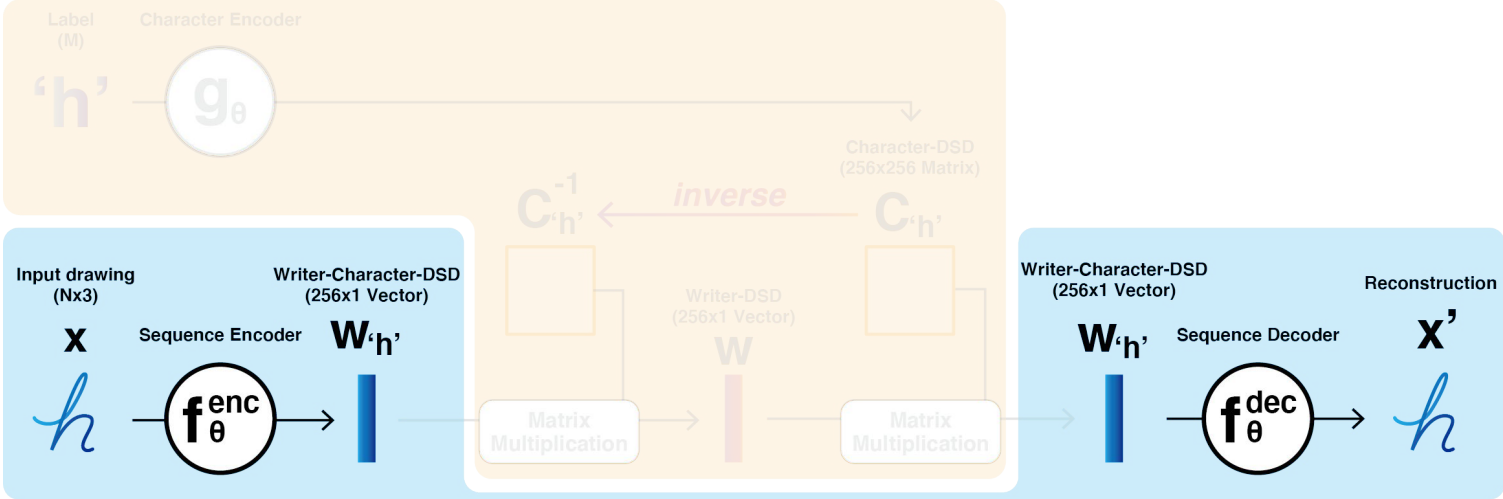
Decoupled Style Descriptors (DSD)



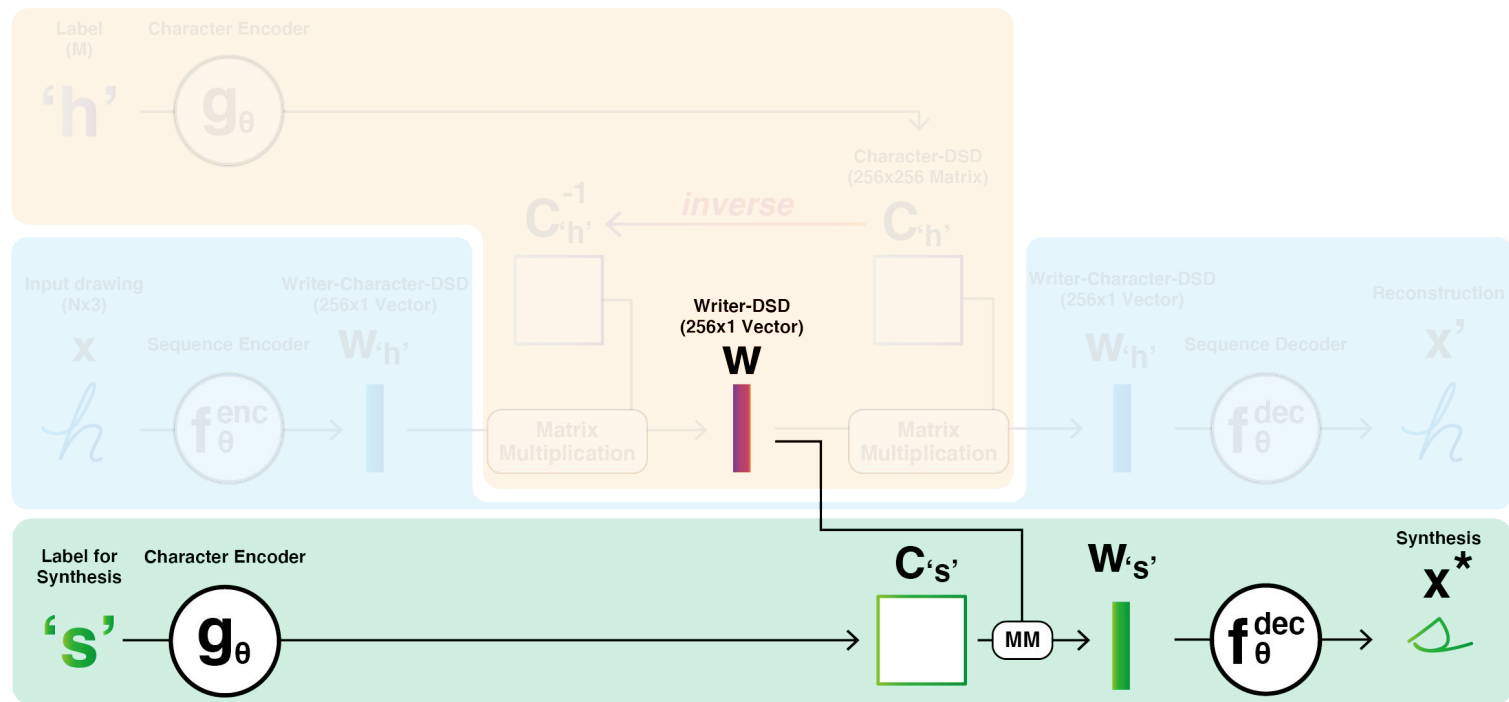
Decoupled Style Descriptors (DSD)



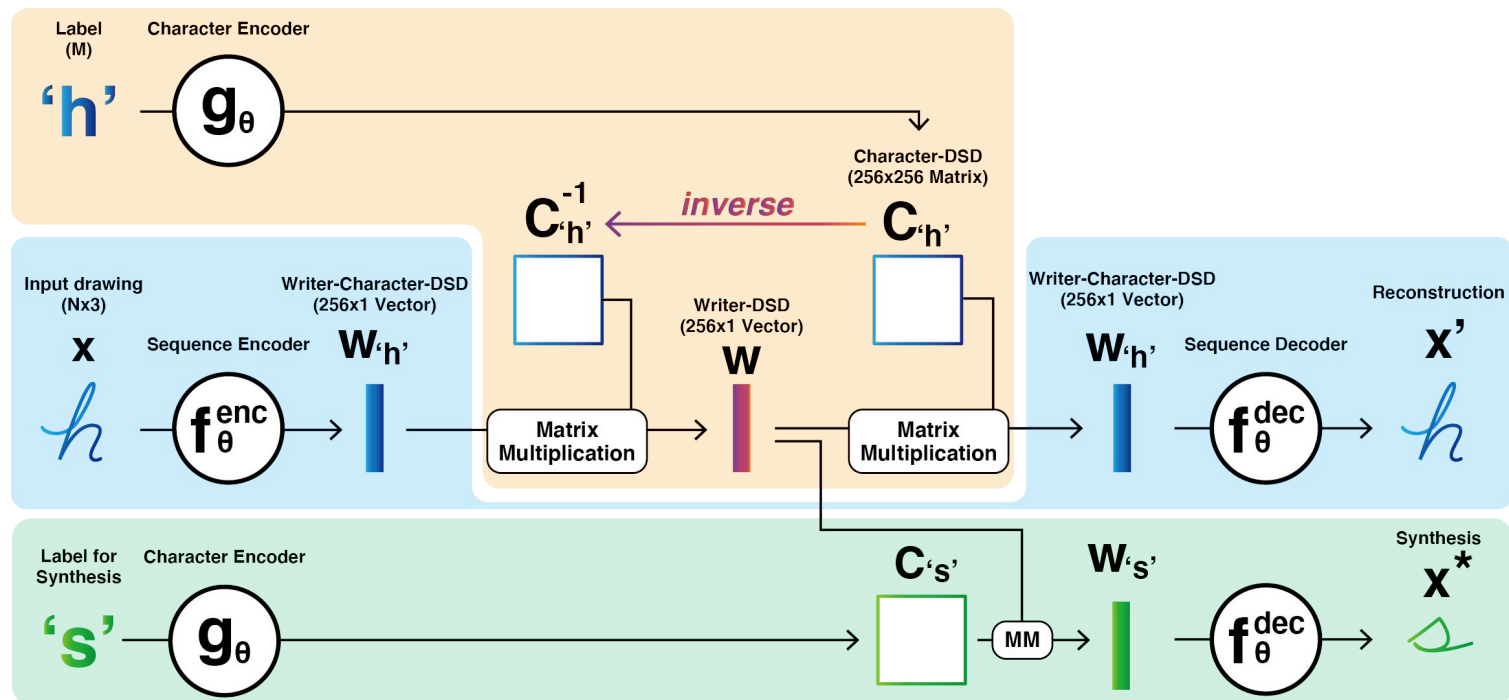
Decoupled Style Descriptors (DSD)



Decoupled Style Descriptors (DSD)



Decoupled Style Descriptors (DSD)



Subsequences rather than single characters

Writing is complex:

- Cursive
- Character pairs (ligatures)
- Delayed strokes (f, t, i, j)

Subsequences rather than single characters

Writing is complex:

- Cursive
- Character pairs (ligatures)
- Delayed strokes (f, t, i, j)

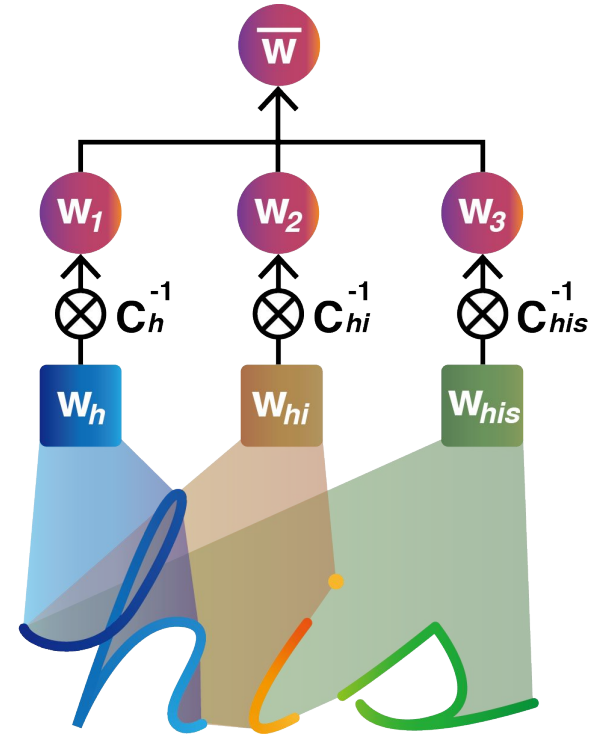
Our approach actually represents latent space of *all subsequences of characters*.

Word 'his' has representations $C_{\text{'h'}}$, $C_{\text{'hi'}}$, and $C_{\text{'his'}}$.
and $w_{\text{'h'}}$, $w_{\text{'hi'}}$, and $w_{\text{'his'}}$.

Recovering **Writer-DSD** \mathbf{w} from handwriting samples

Take mean \mathbf{w} over subsequences:

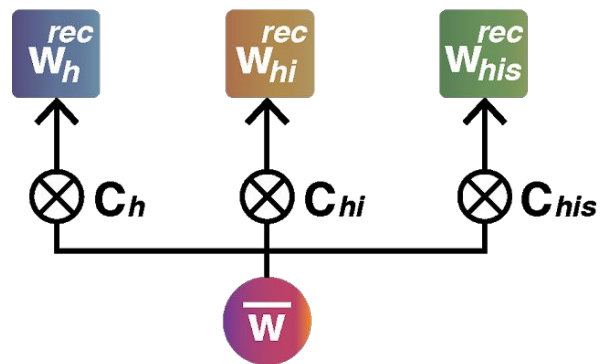
$$\bar{\mathbf{w}} = \frac{1}{M} \sum_{t=1}^M \mathbf{C}_{c_t}^{-1} \mathbf{w}_{c_t}$$



Generating handwriting using a *global* Writer-DSD w

Given a target word 'his':

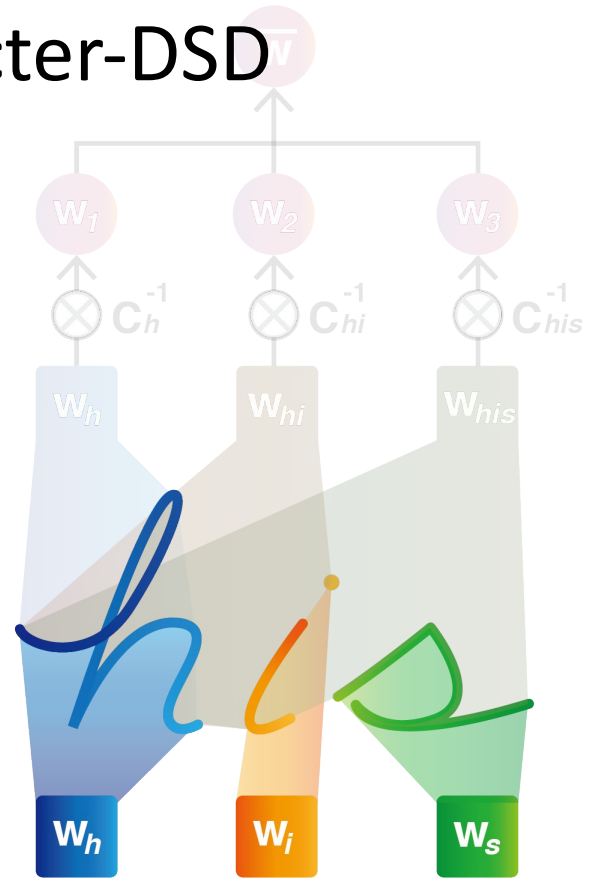
- Predict $C_{'h'}$, $C_{'hi'}$, and $C_{'his'}$
- Multiply by w to create $w_{'h'}$, $w_{'hi'}$, and $w_{'his'}$
- Decode ($w_{'h'}$, $w_{'hi'}$, $w_{'his'}$) into stroke sequence



Single-character Writer-Character-DSD

There are relatively few single-character $w_{h'}$.

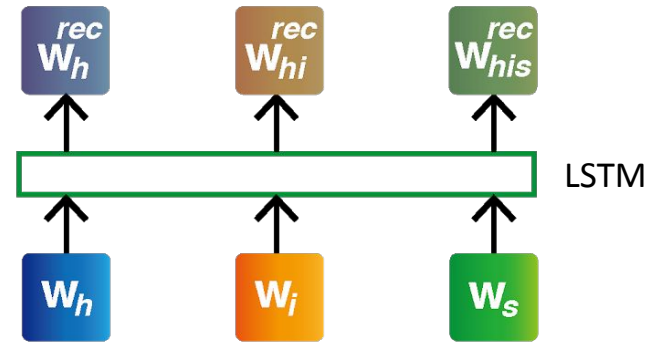
If we extract them from a writing sample, we can save them in a **database** and **sample** them during generation.



Generating handwriting using *sampled* Writer-Character-DSD

Retrieve relevant single-characters w_h , w_i , w_s .

Restore temporal dependencies via LSTM.



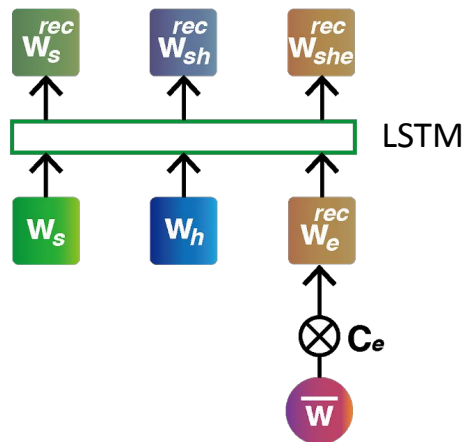
Cannot cope with any missing characters in reference handwriting samples.

Combined method with *sampling*

Reference sample: 'his'

Generation target: 'she'

- Compute mean \mathbf{w} from all substrings
- Predict $\mathbf{w}_{e'}$ with $\mathbf{C}_{e'}$ and the mean \mathbf{w}
- Extract single-character Writer-Character-DSDs
- *Restore temporal dependencies with LSTM.*

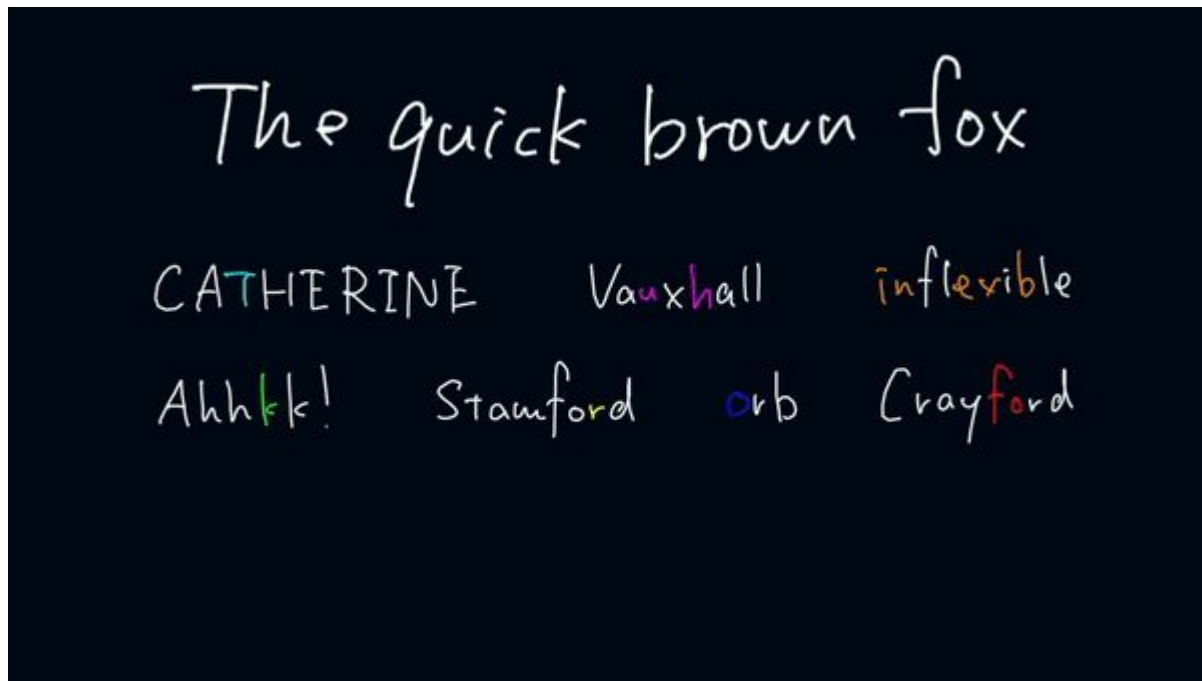


Generated Results

Colored characters match between provided writing samples and desired output.

These are generated from retrieved $w_{h'}$

Missing characters are generated from global w



Generated Results

Target Image

particularly

d to see her.

the Connecticut

 Ours w/ global DSD

particularly

d to see her

the Connecticut

 Ours w/ sampling

particularly

d to see her

the Connecticut

Target Image

gether with the

young offende

earth, marit

 Ours w/ global DSD

gether with the

young offende

earth marit

 Ours w/ sampling

gether with the

young offende

earth marit

Generated Results - Comparison

Target Image

mentioned the f gether with the ft of mainly

 **DeepWriting [1]**

mentioned the f gether with the ft of mainly

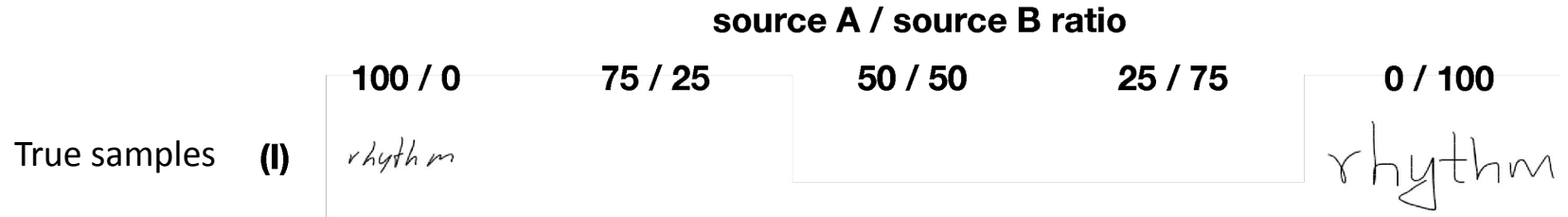
 **Ours w/ global DSD**

mentioned the f gether with the ft of mainly

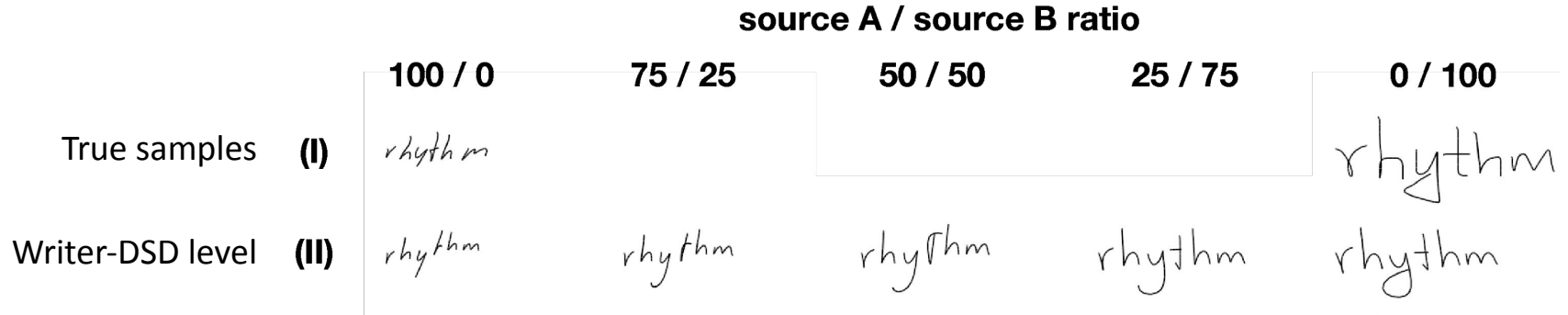
 **Ours w/ sampling**

mentioned the f gether with the ft of mainly

Interpolation



Interpolation



Interpolation

		source A / source B ratio				
		100 / 0	75 / 25	50 / 50	25 / 75	0 / 100
True samples	(I)	<i>rhythm</i>				<i>rhythm</i>
Writer-DSD level	(II)	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>
Writer-Character-DSD level	(III)	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>

Interpolation

		source A / source B ratio				
		100 / 0	75 / 25	50 / 50	25 / 75	0 / 100
True samples	(I)	<i>rhythm</i>				<i>rhythm</i>
Writer-DSD level	(II)	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>
Writer-Character-DSD level	(III)	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>	<i>rhythm</i>
Character-DSD level	(IV)	a c a a a a a a a a a c c a a a a a d d d c c c a a d d d d d c c c c d d d d d d	L L L l l t t x x x x L L L l l t t x x x x L L h l l t t x x x x L L h l l t t x x x x L L h l l t t y y y y L L h l l t t y y y y h h h h h h h y y y h h h h h h h y y y h h h h h h h y y y	R R R R R r r y k k k k R R R R R r r y k k k k R R R R R r r y k k k k R R R R R r r y k k k k R R R R R r r n n n n n R R R R R r r n n n n n m m m m n n n n n n m m m m n n n n n n m m m m n n n n n n	O O O O O w w w w w O O O O O w w w w w O O O O O w w w w w O O O O O w w w w w O O O O O w w w w w O O O O O w w w w w O O O O O q q q q q q P P P O q q q q q q P P P O q q q q q q P P P O q q q q q q	

Few-shot learning of New Characters

Writer A

Writer B

Source for W

goals politics, [A

goals politics; [A

C from 1 sample

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

C from 10 samples

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

C from 100 samples

0 1 2 3 4 5 6 7 8 9

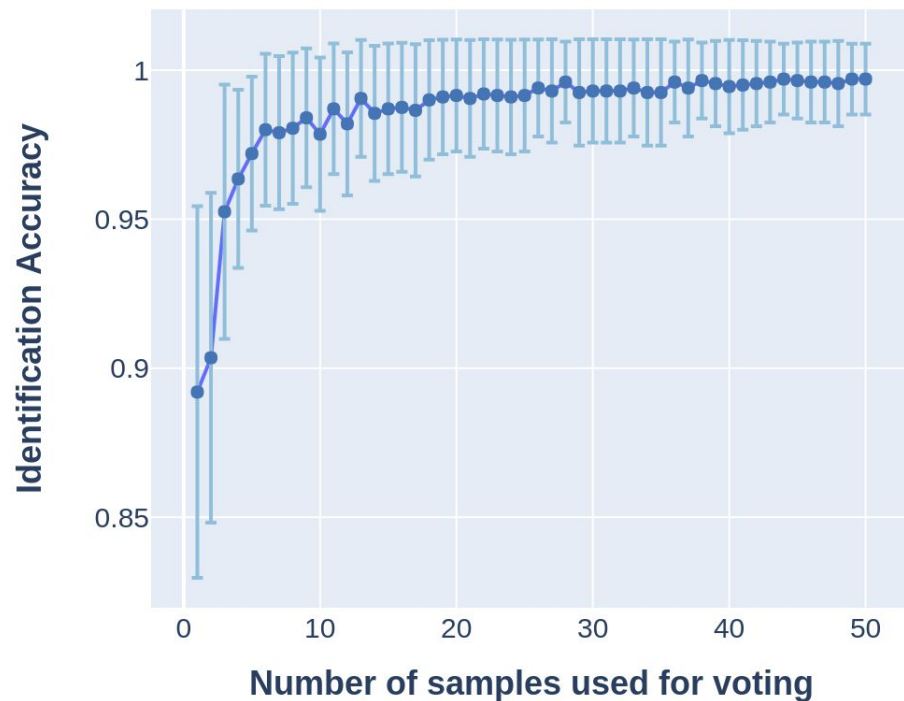
0 1 2 3 4 5 6 7 8 9

Writer Identification

In a 20-writer identification task,
our model achieved:

89.20% \pm 6.23 for **1** word sample

99.70% \pm 1.18 for **50** word samples



BRUSH dataset

BRUSH dataset contains handwriting of:

- 170 writers, 86 Latin alphabet characters
- 488 common words written by all writers
 - 99.5% coverage of two-character letter space
- + 3668 rarer words written across writers
 - 99.9% coverage in total

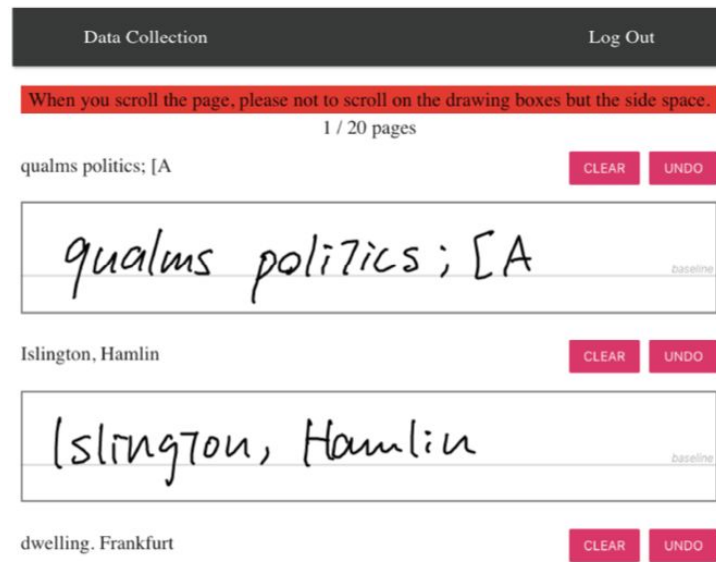
The screenshot displays the BRUSH dataset interface. At the top, there is a dark grey header with 'Data Collection' on the left and 'Log Out' on the right. Below the header is a red warning bar that reads: 'When you scroll the page, please not to scroll on the drawing boxes but the side space.' Underneath the warning bar, the page number '1 / 20 pages' is shown. The main content area features three examples of handwriting on a white background with a light blue baseline. Each example consists of a text label, a drawing box containing the handwritten text, and two buttons labeled 'CLEAR' and 'UNDO' to the right. The first example shows the text 'qualms politics; [A' with the handwritten text 'qualms politics; [A'. The second example shows the text 'Islington, Hamlin' with the handwritten text 'Islington, Hamlin'. The third example shows the text 'dwelling. Frankfurt' with the handwritten text 'dwelling. Frankfurt'.

BRUSH dataset

BRUSH dataset contains handwriting of:

- 170 writers, 86 Latin alphabet characters
- 488 common words written by all writers
 - 99.5% coverage of two-character letter space
- + 3668 rarer words written across writers
 - 99.9% coverage in total

Common word samples & character-level labels are not present in IAM dataset [4].



Generating Handwriting via Decoupled Style Descriptors

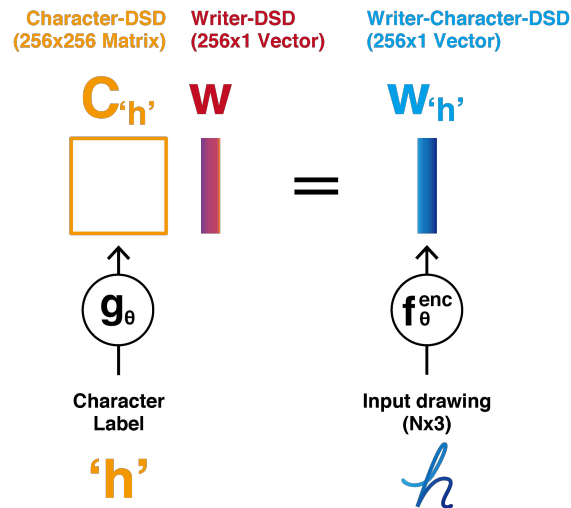
Atsunobu Kotani, Stefanie Tellex, James Tompkin



BROWN
Computer Science

Contributions:

- DSDs decouple character style from writer style.
- Allow flexible handwriting generation.
- BRUSH: new dataset for online handwriting.



Code & dataset available at
<http://dsd.cs.brown.edu>

Generating Handwriting via Decoupled Style Descriptors

Atsunobu Kotani, Stefanie Tellex, James Tompkin



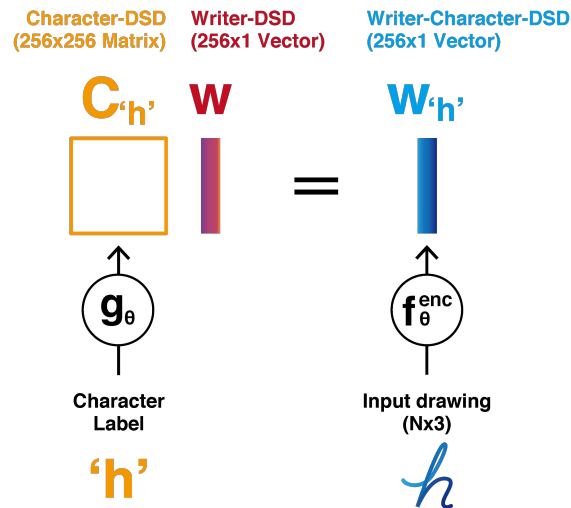
BROWN
Computer Science

Contributions:

- DSDs decouple character style from writer style.
- Allow flexible handwriting generation.
- BRUSH: new dataset for online handwriting.

Potential Extension:

- Explicit decoupling may work better than implicit decoupling / disentanglement.
- DSDs could be applied to other sequential data (e.g. speech, motion capture data)



Code & dataset available at
<http://dsd.cs.brown.edu>