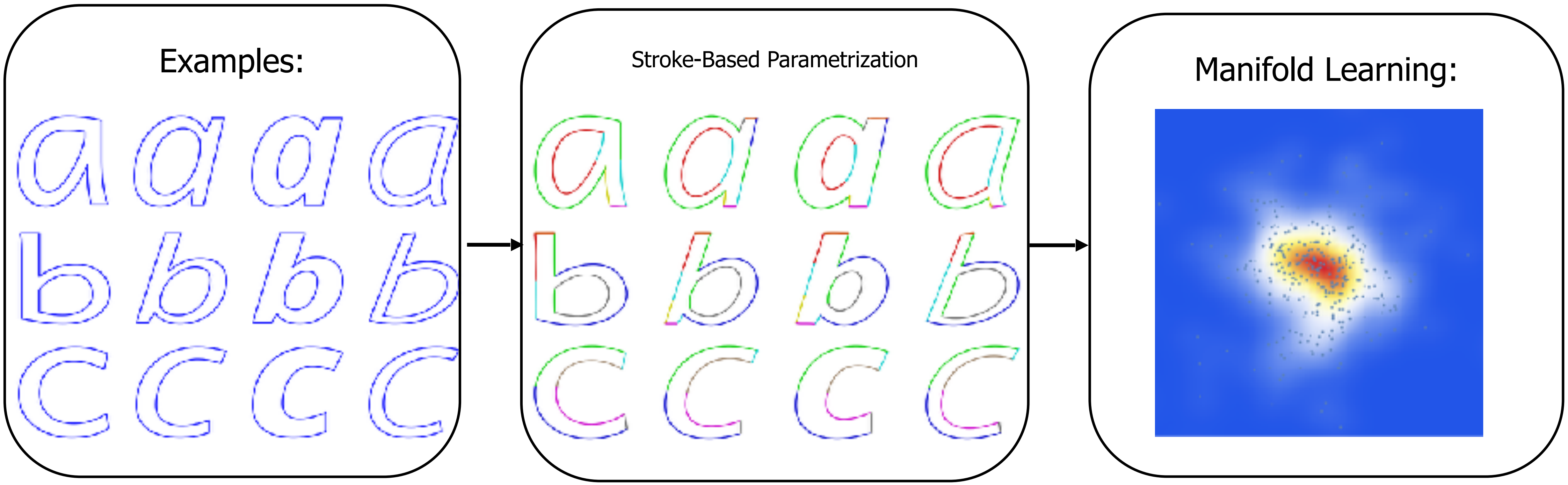
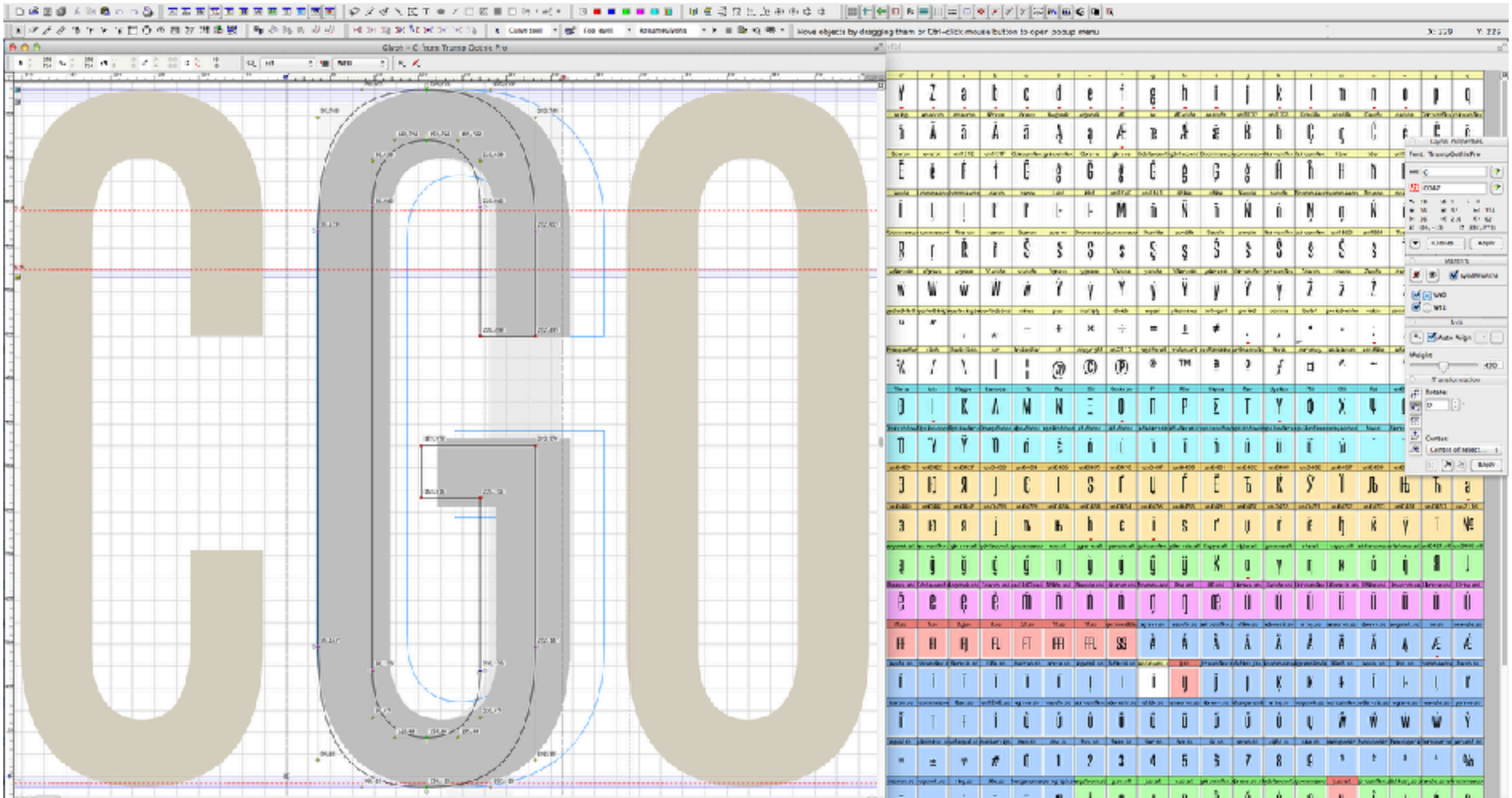
Learning A Stroke-Based Representation for Fonts

E. Balashova,[1] A. H Bermano,[1] V. G. Kim,[2] S. DiVerdi,[2]
A. Hertzmann,[2] T. Funkhouser[1]
1 - Princeton University
2 - Adobe Research

Goal: Stroke-Based Font Representation Suitable for Learning



Motivation



[https://www.fontlab.com]

Motivation

Motivation


h a n d g z o v e b c f i j k m p q r s t u w x y z
h a n d g z o v e b c f i j k m p q r s t u w x y z

Desired Font Representation Characteristics

Desired Font Representation Characteristics

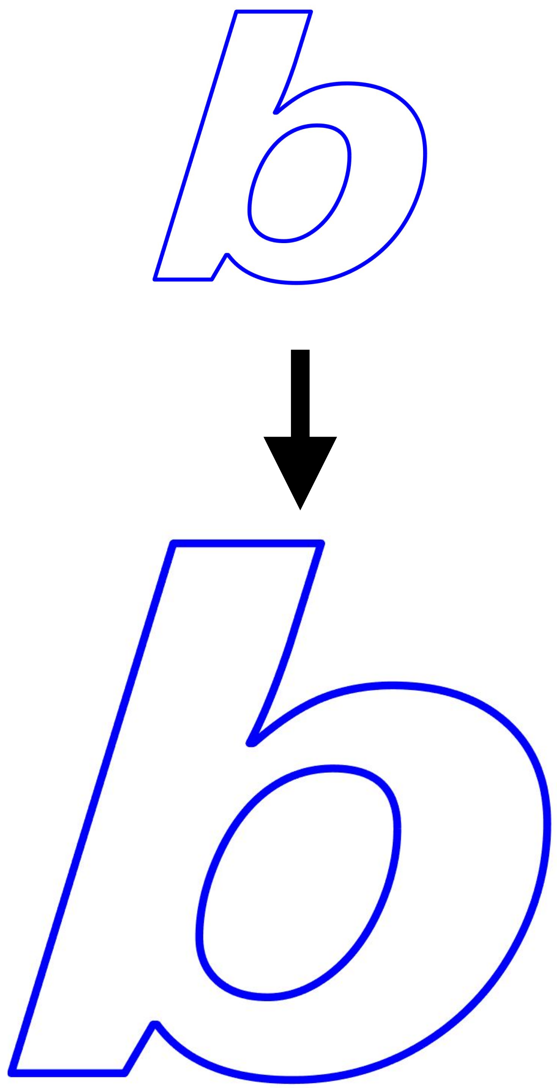
- Detail-preserving

Desired Font Representation Characteristics

- Structure-aware

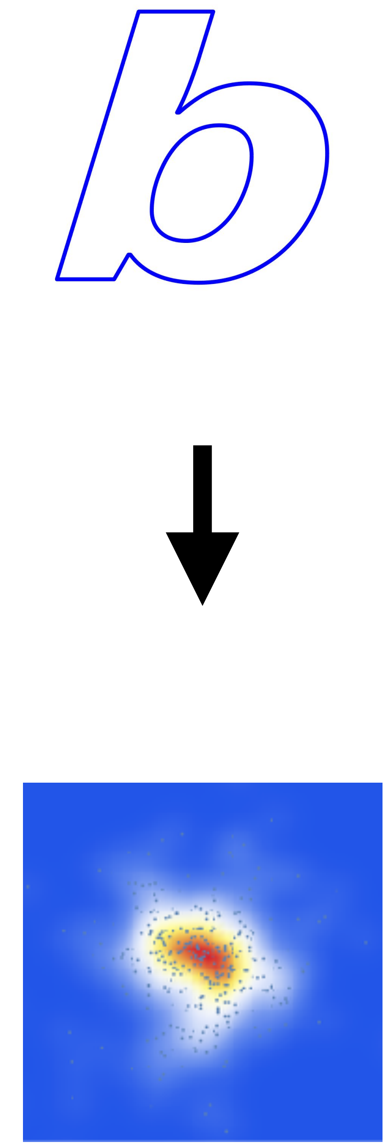
Desired Font Representation Characteristics

- Scales without artifacts



Desired Font Representation Characteristics

- Suitable for learning



Font Representations

- Raster - Based
- Contour - Based

Font Representations

Raster - Based

Limitations

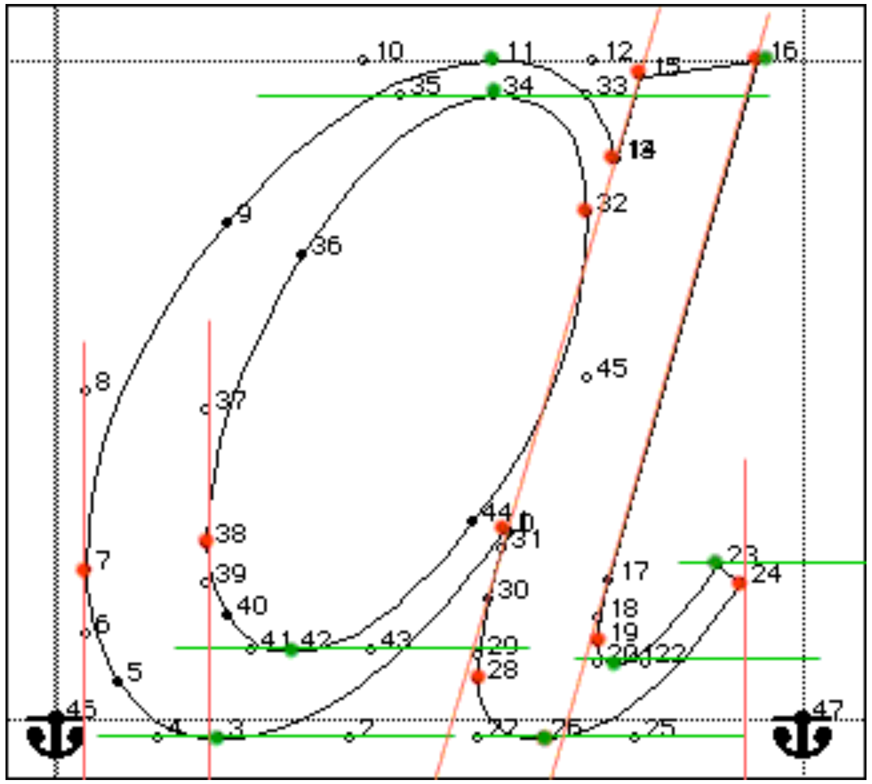
- Scales with artifacts
- Not detail-preserving
- Not structure-aware

Font Representations

Contour-Based

Font Representations

Bezier-Based



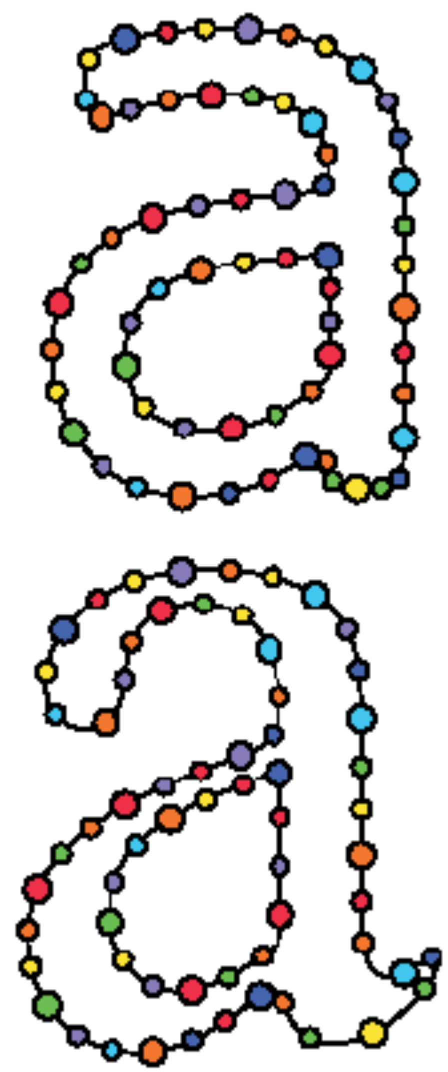
[Microsoft Docs]

Limitations

- Not suitable for learning

Font Representations

Consistent Outline-Based



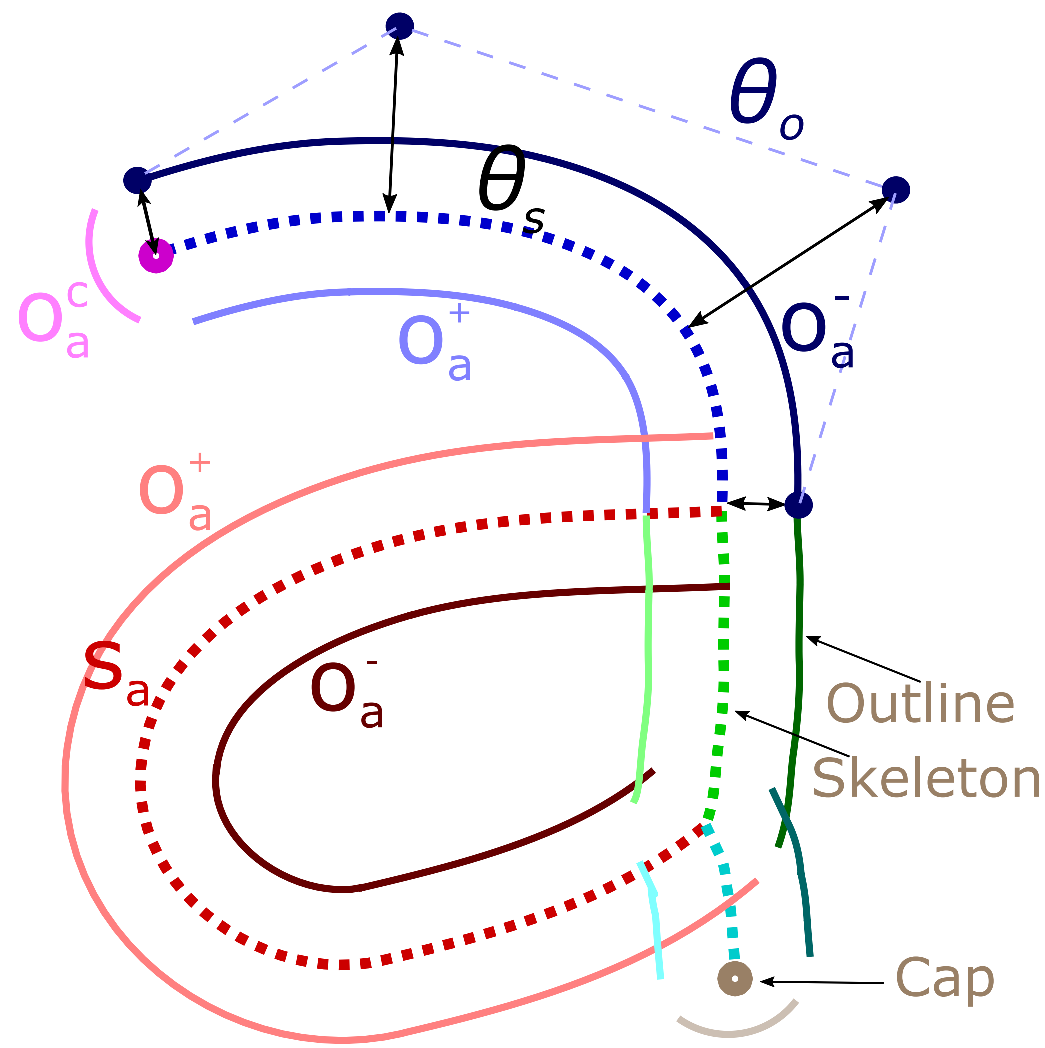
[Campbell '14]

Limitations

- Not structure-aware

Font Representations

Proposed Representation



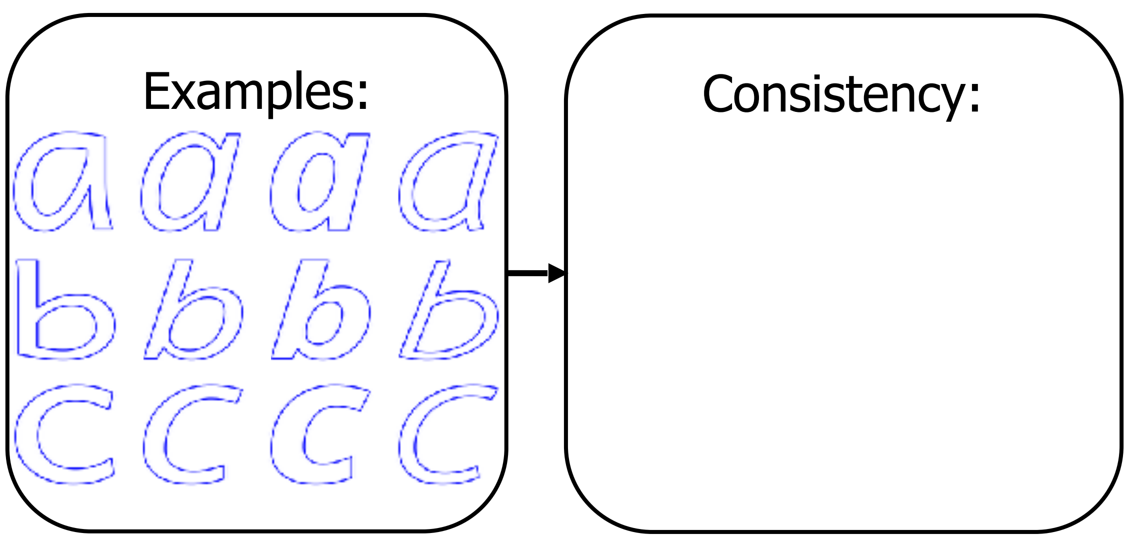
Part-Aware

Skeleton-Based Representation

Characteristics

- Detail-preserving
- Structure-aware
- Scales without artifacts
- Suitable for learning

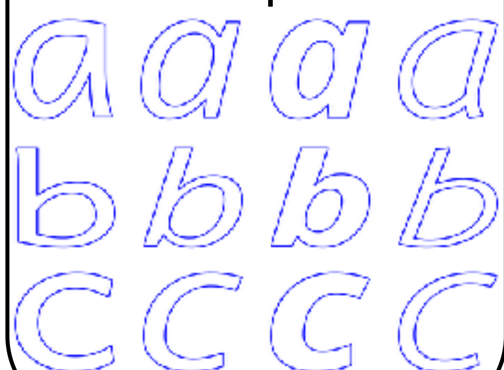
Our Approach



Our Approach

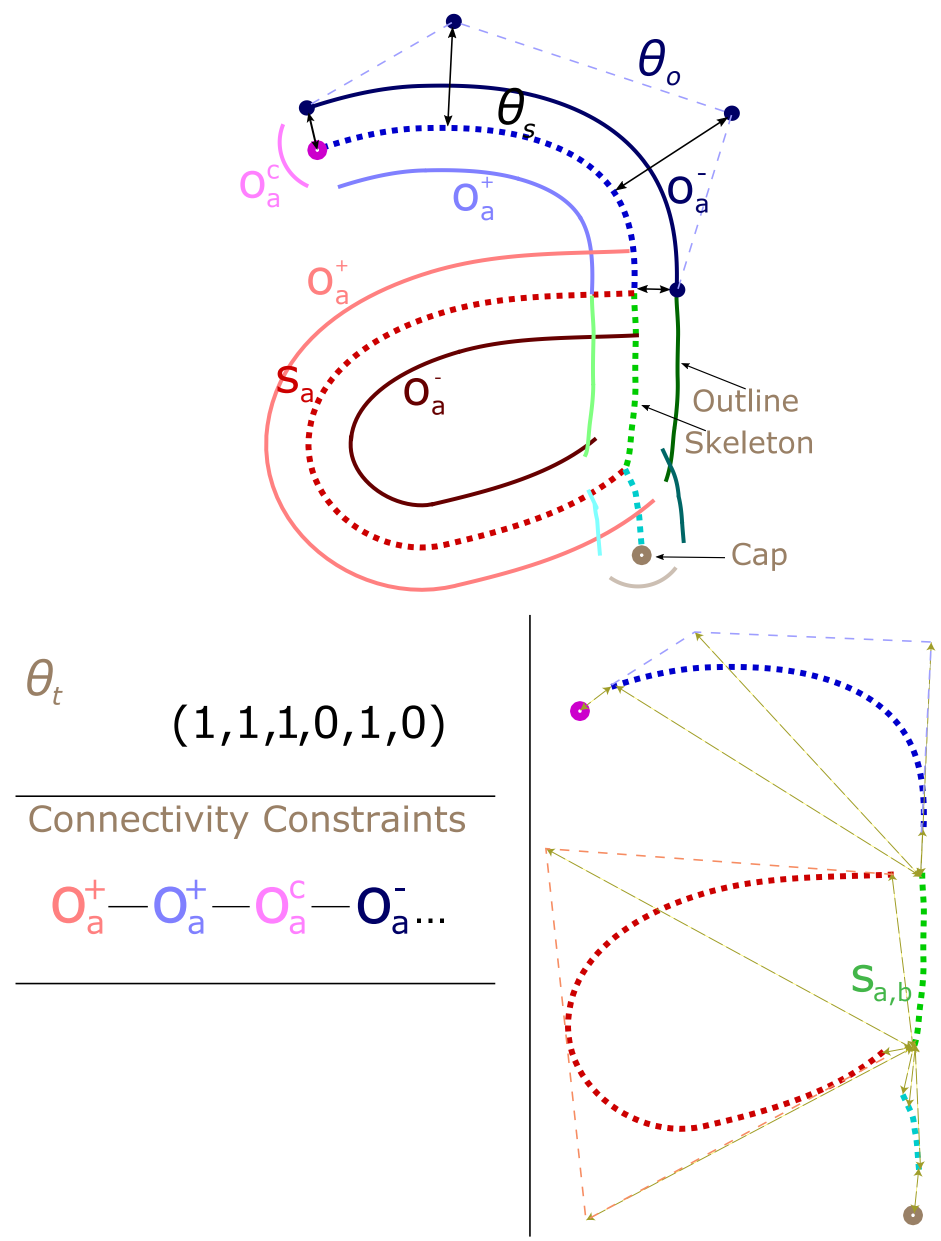
Template Fitting

Examples:



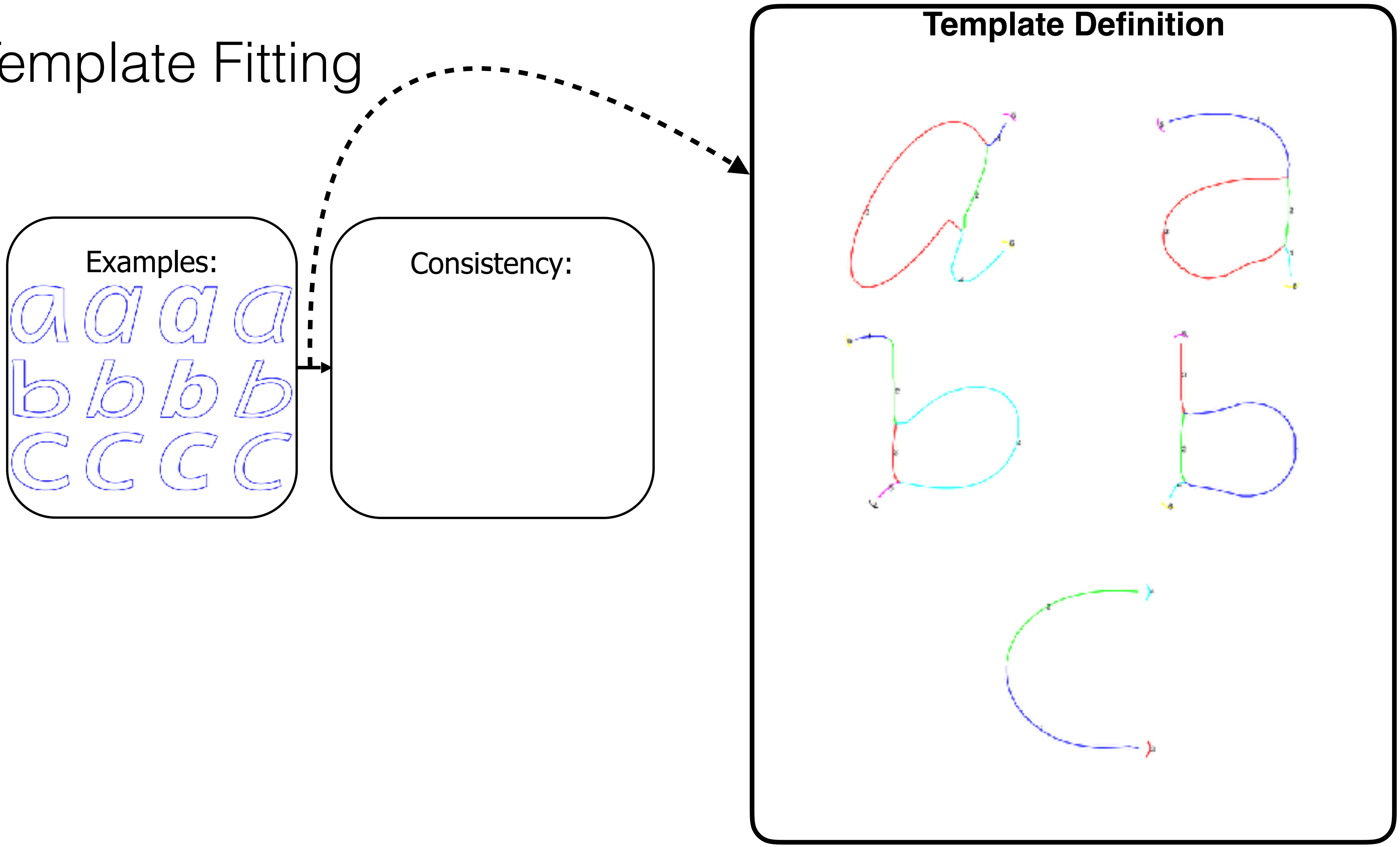
Consistency:

Template Definition

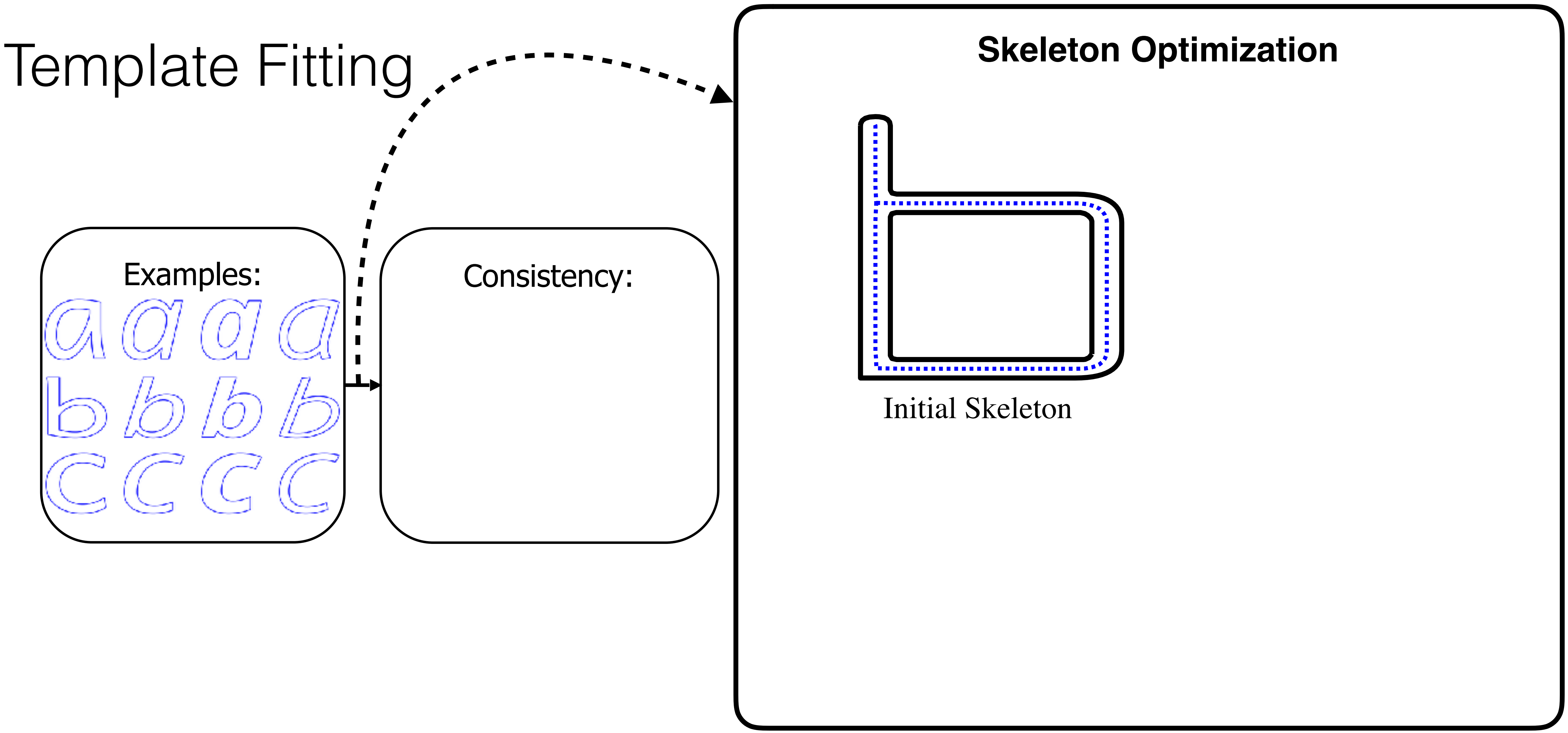


Our Approach

Template Fitting

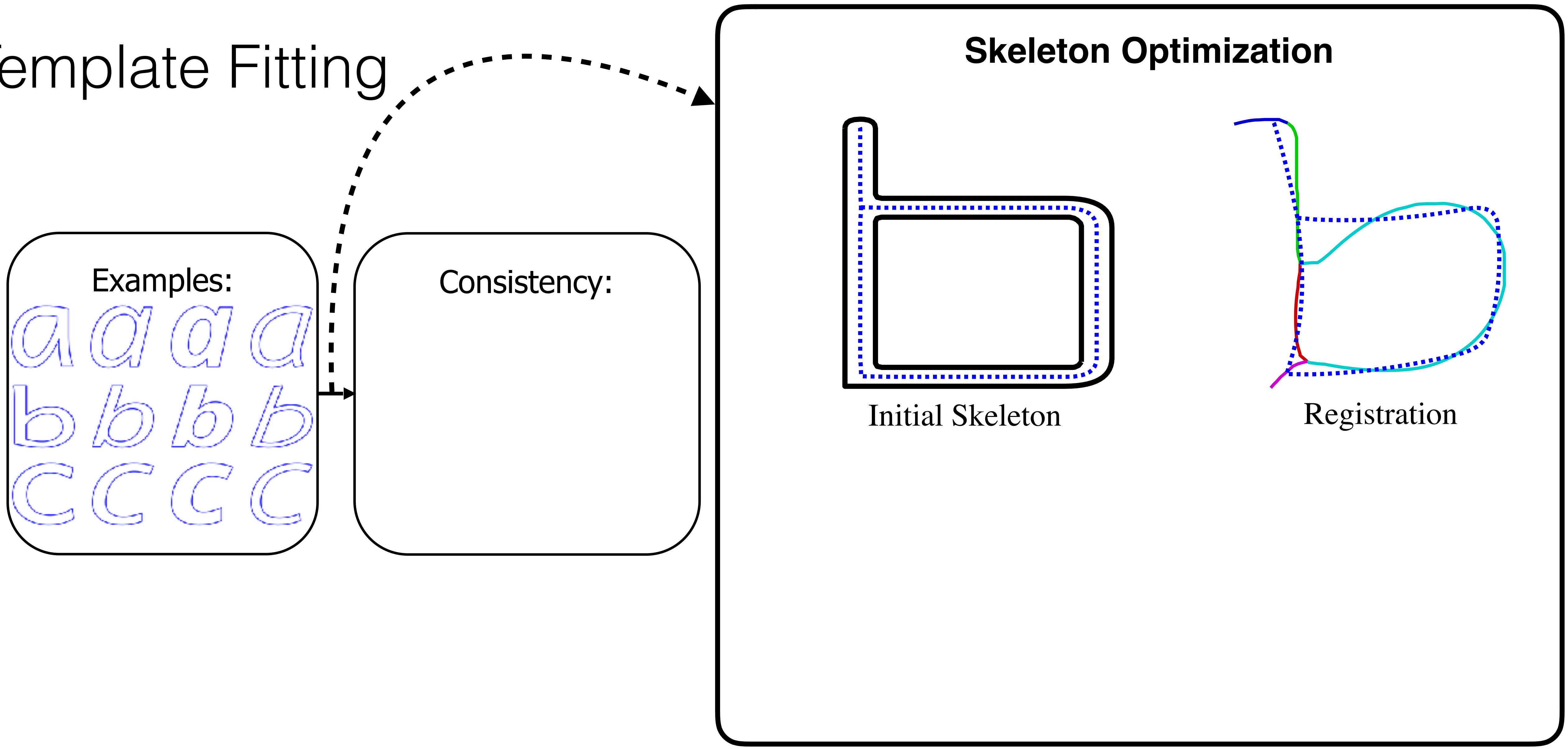


Our Approach

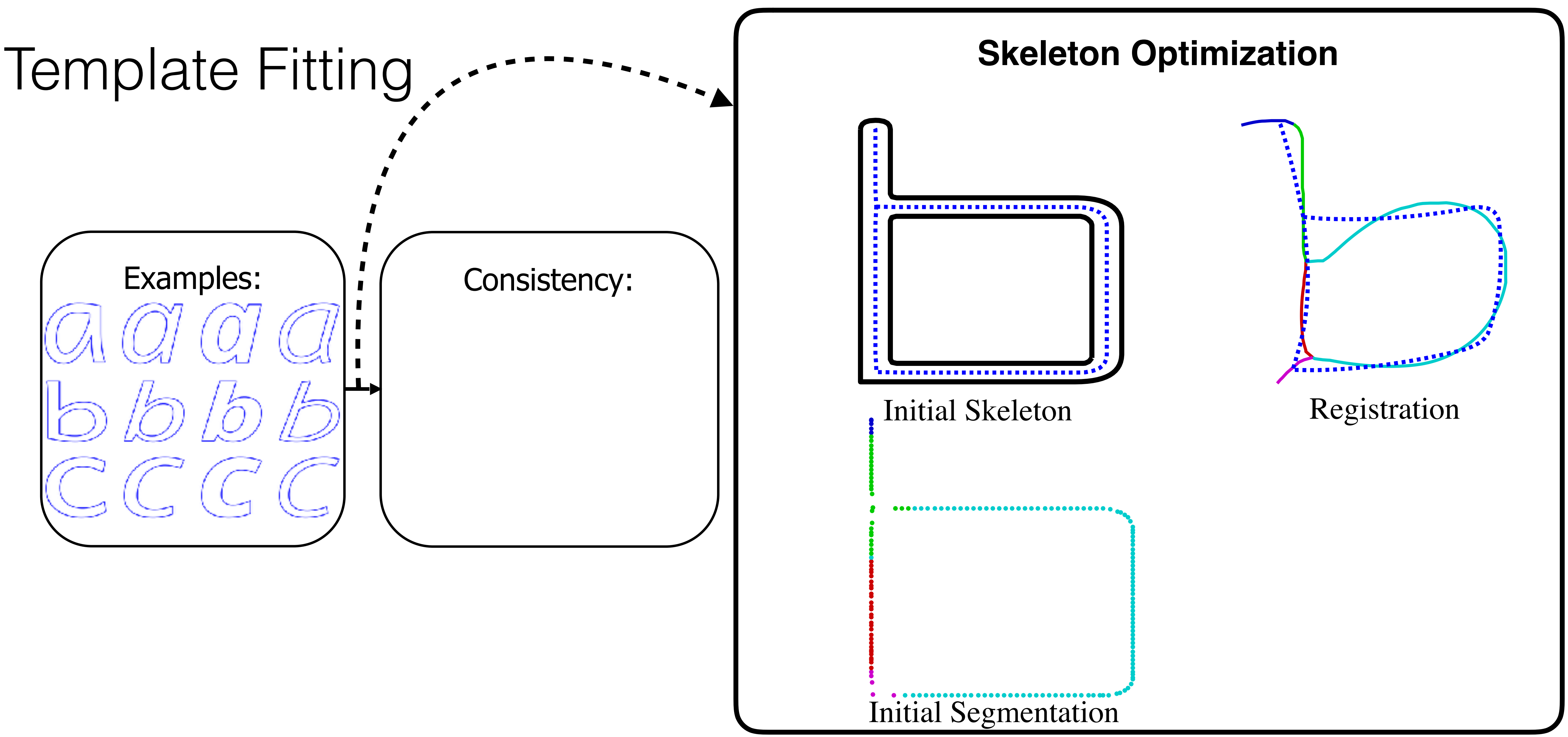


Our Approach

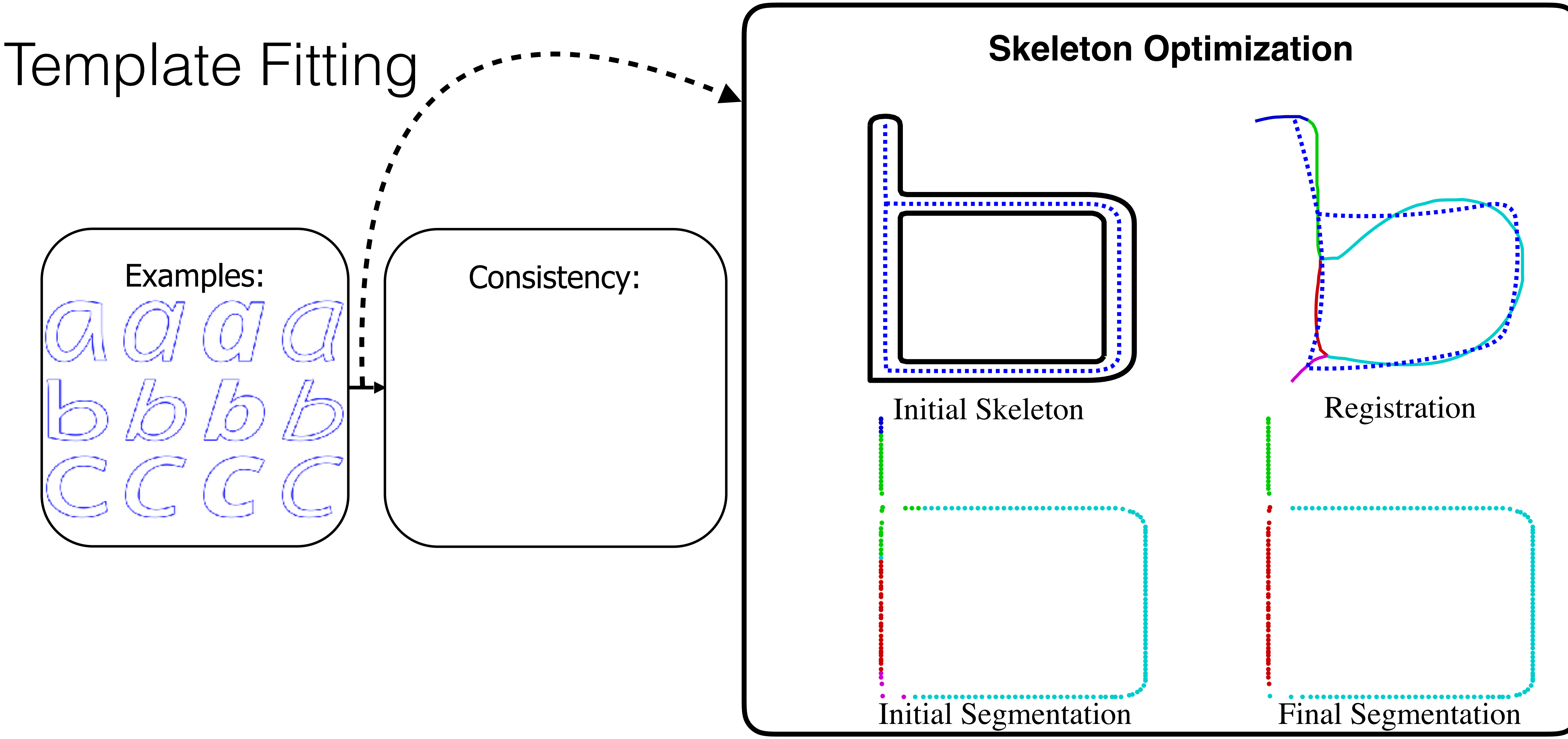
Template Fitting



Our Approach



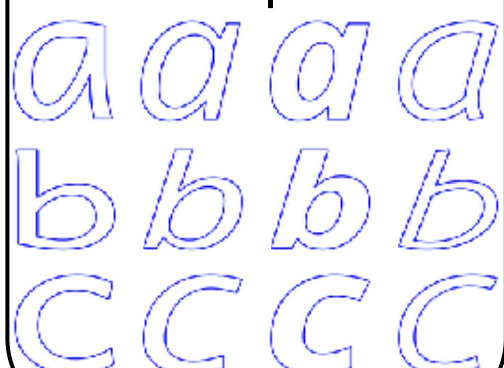
Our Approach



Our Approach

Template Fitting

Examples:



Consistency:

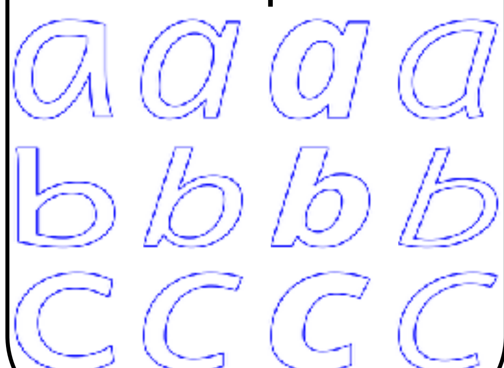
Outline Optimization

$$E = E_{\text{corr}} + E_{\text{ft}}$$

Our Approach

Template Fitting

Examples:



Consistency:

Outline Optimization

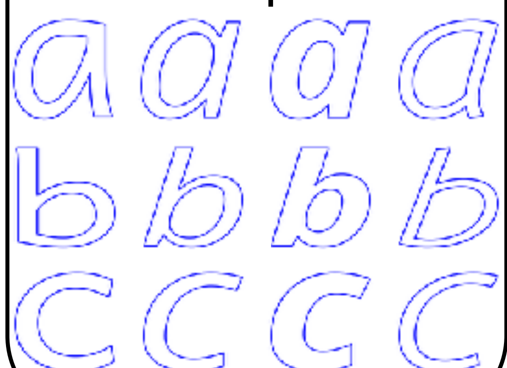
$$E = E_{\mathrm{corr}} + E_{\mathrm{ft}}$$

$$E_{\mathrm{corr}}(G_{c,f}, \Theta) = \sum_{x \in P(G_{c,f})} h_{\sigma_{\mathrm{corr}}}\left(D(x, O_c(\Theta))\right) + \sum_{y \in P(O_c(\Theta))} h_{\sigma_{\mathrm{corr}}}\left(D(y, G_{c,f})\right)$$

Our Approach

Template Fitting

Examples:



Consistency:

Outline Optimization

$$E = E_{\mathrm{corr}} + E_{\mathrm{ft}}$$

template parameters

Gaussian kernel

generated outline

$$E_{\mathrm{corr}}(G_{c,f}, \Theta) = \sum_{x \in P(G_{c,f})} h_{\sigma_{\mathrm{corr}}}\left(D(x, O_c(\Theta))\right) + \sum_{y \in P(O_c(\Theta))} h_{\sigma_{\mathrm{corr}}}\left(D(y, G_{c,f})\right)$$

GT outline

dense curve sampling

normal

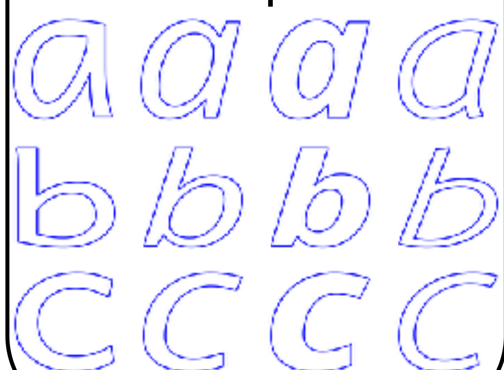
$$x = [q_X, q_Y, w_{\mathrm{n}} n_X(q), w_{\mathrm{n}} n_Y(q)]$$

coordinates normal weight

Our Approach

Template Fitting

Examples:



Consistency:

Outline Optimization

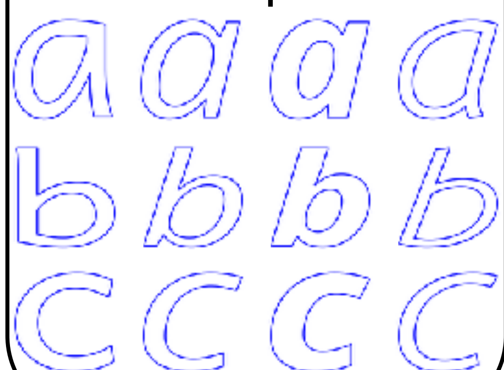
$$E = E_{\text{corr}} + E_{\text{ft}}$$

$$E_{\text{ft}}(G_{c,f}, \Theta) = \sum_{x \in F(O_c(\Theta))} D_{\text{curve}}(x, F_{\tau_{\text{ft}}}(G_{c,f}))$$

Our Approach

Template Fitting

Examples:



Consistency:

Outline Optimization

$$E = E_{\text{corr}} + E_{\text{ft}}$$

template parameters

arc-length intrinsic distance

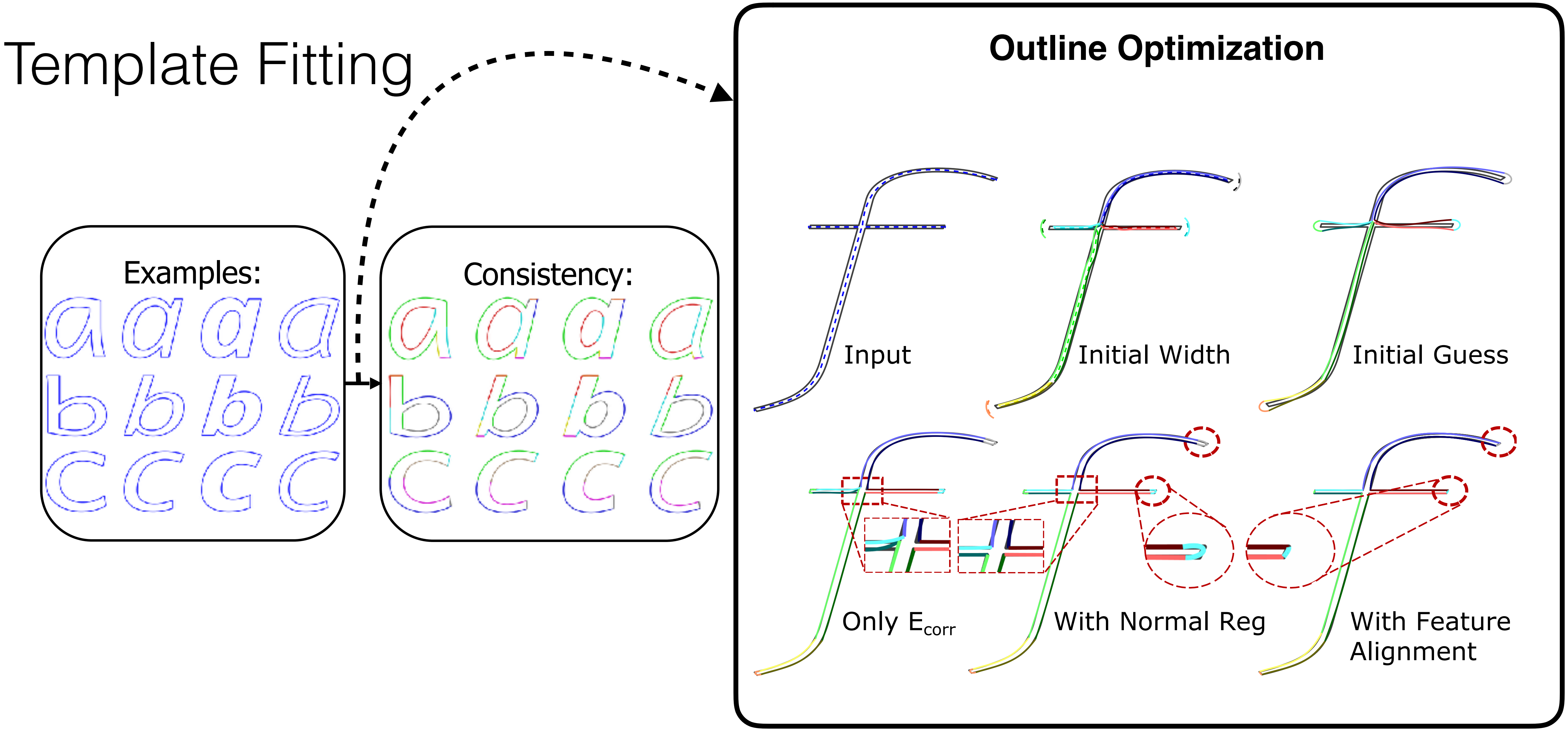
$$E_{\text{ft}}(G_{c,f}, \Theta) = \sum_{x \in F(O_c(\Theta))} D_{\text{curve}}(x, F_{\tau_{\text{ft}}}(G_{c,f}))$$

GT outline

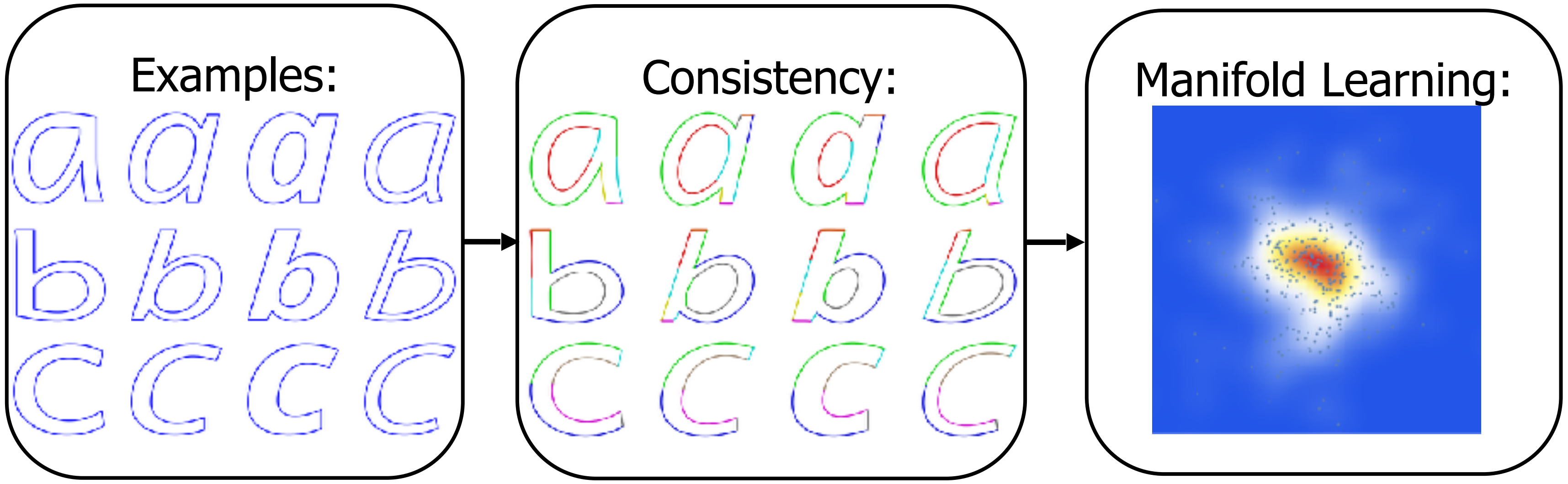
set of feature points

subset of junction points near feature point

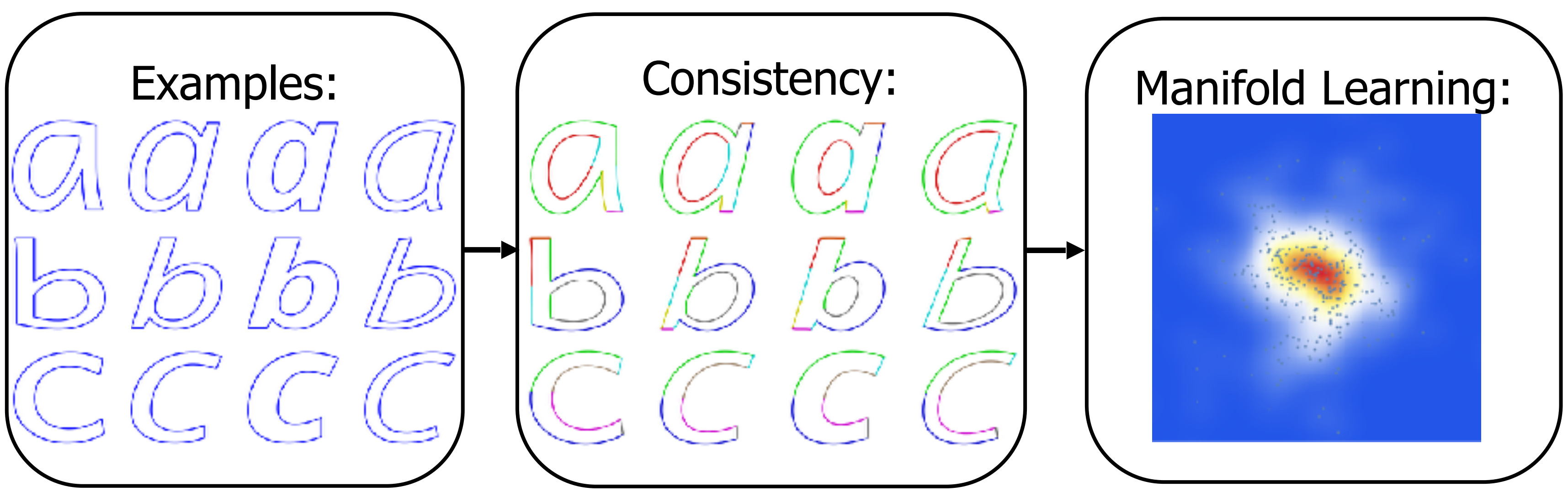
Our Approach



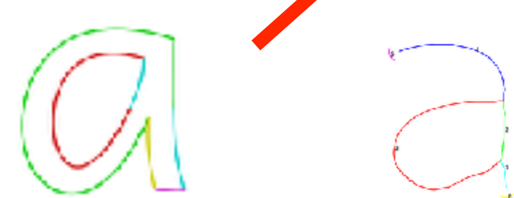
Our Approach



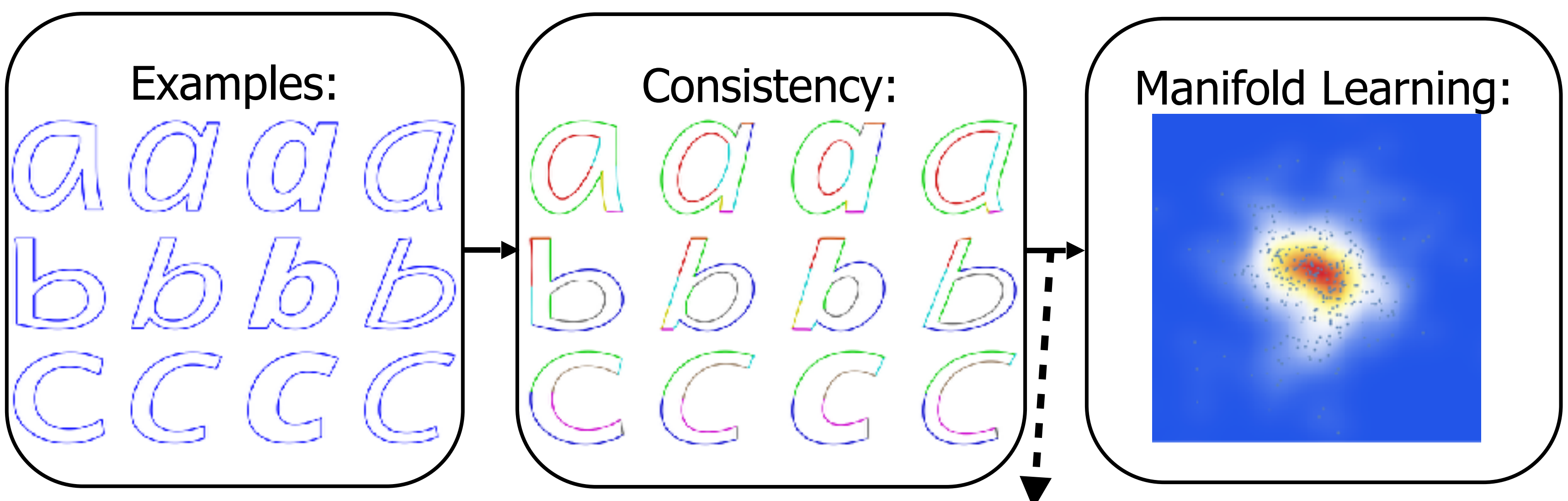
Our Approach



$$f = [f_{a_1} \overset{\text{NaN}}{\cancel{f_{a_2}}} \dots \qquad]$$



Our Approach



EM-PCA

[Row98]

Input Vectors

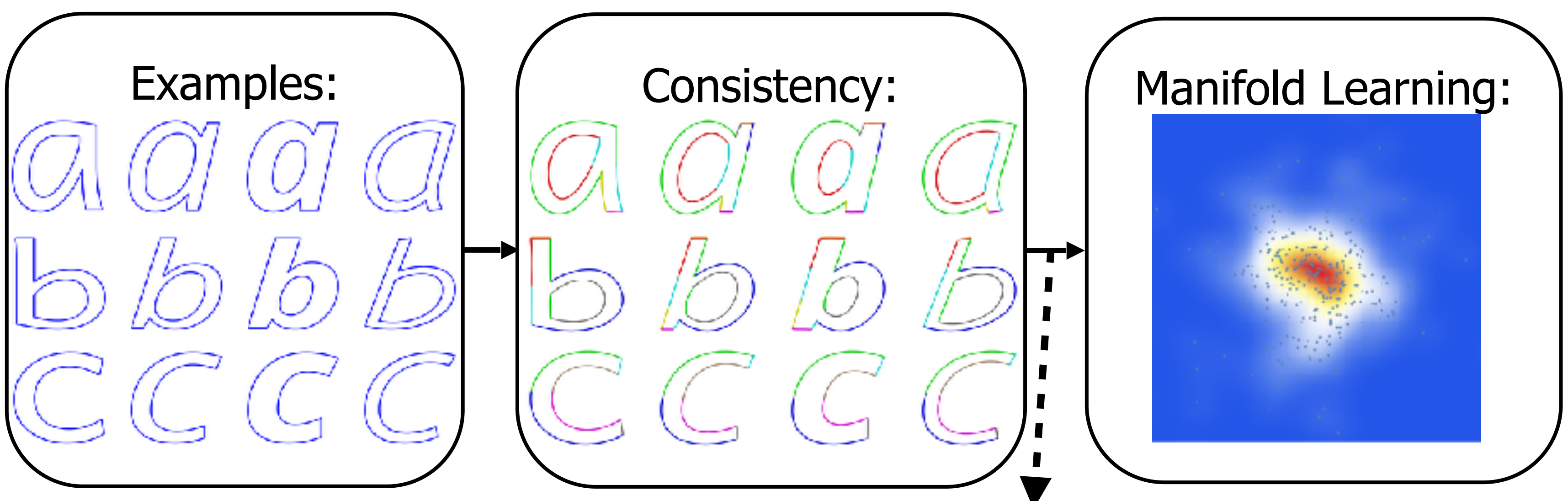
Latent coordinates

$$Y = CX + V$$

Transformation Matrix

Noise

Our Approach



EM-PCA

[Row98]

Input Vectors — Latent coordinates

$$Y = CX + V$$

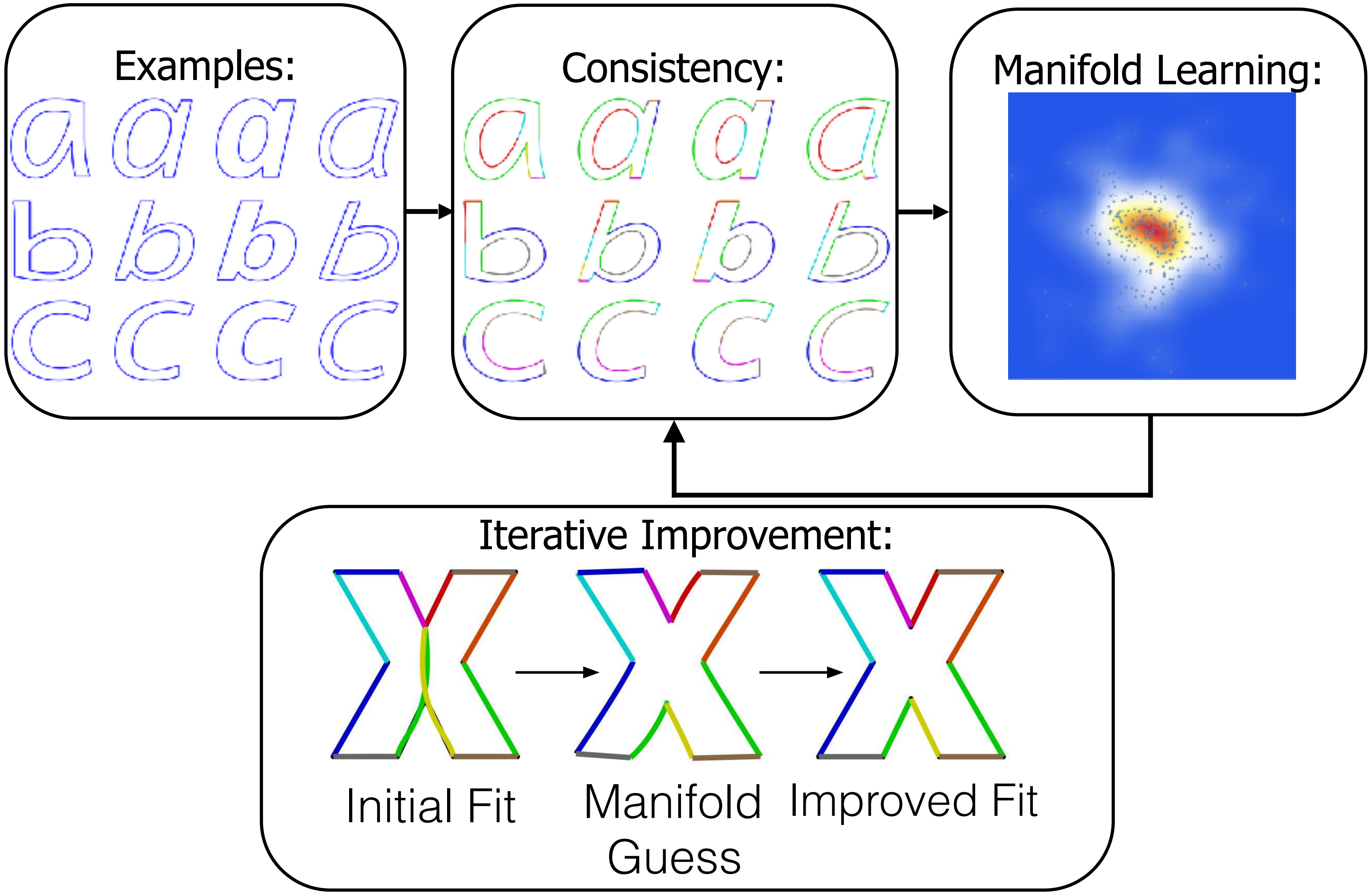
Transformation Matrix — Noise

Learning Steps:

E-step: $X = (C^T C)^{-1} C^T Y$

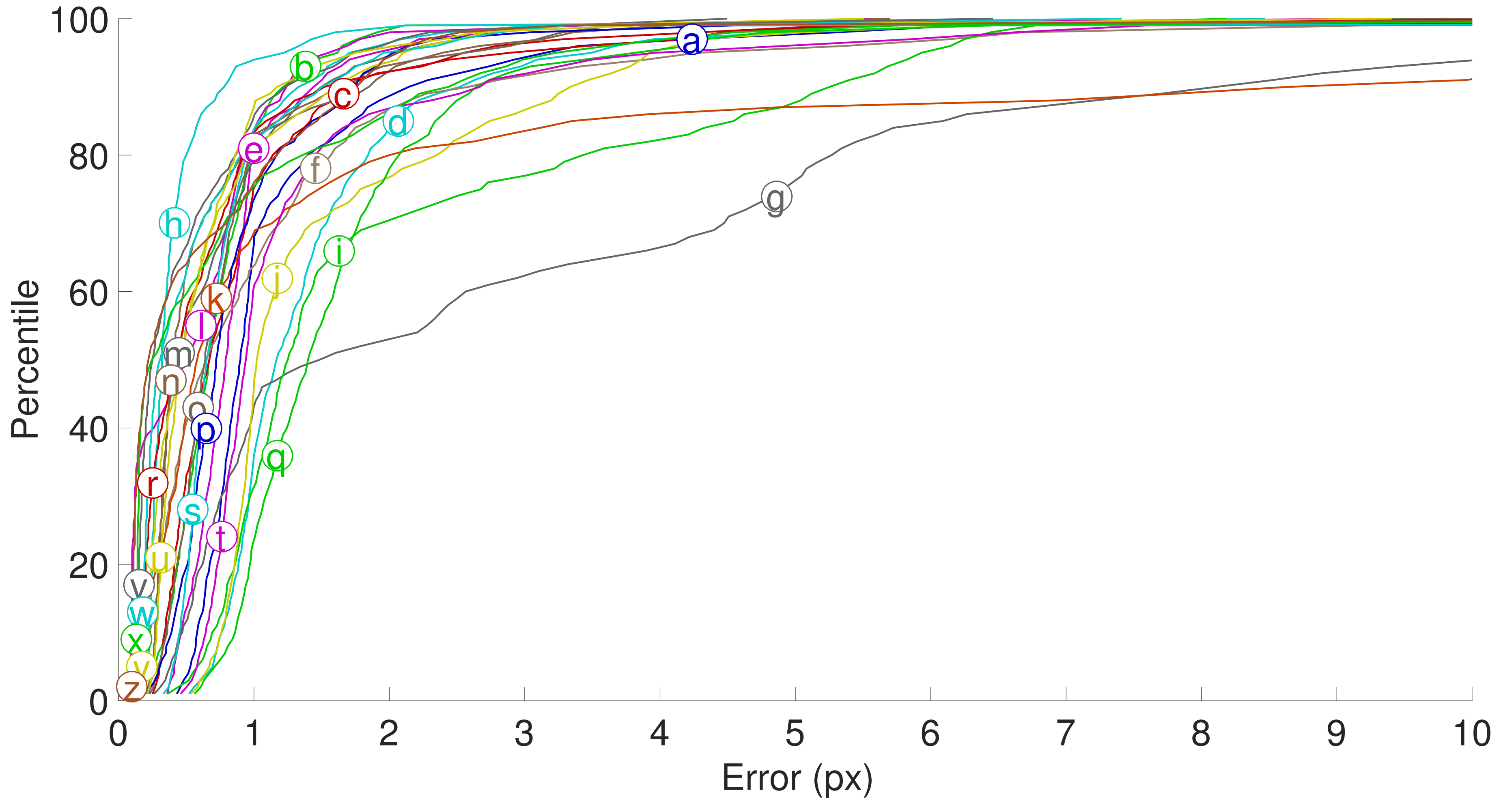
M-step: $C^{new} = YX^T (XX^T)^{-1}$

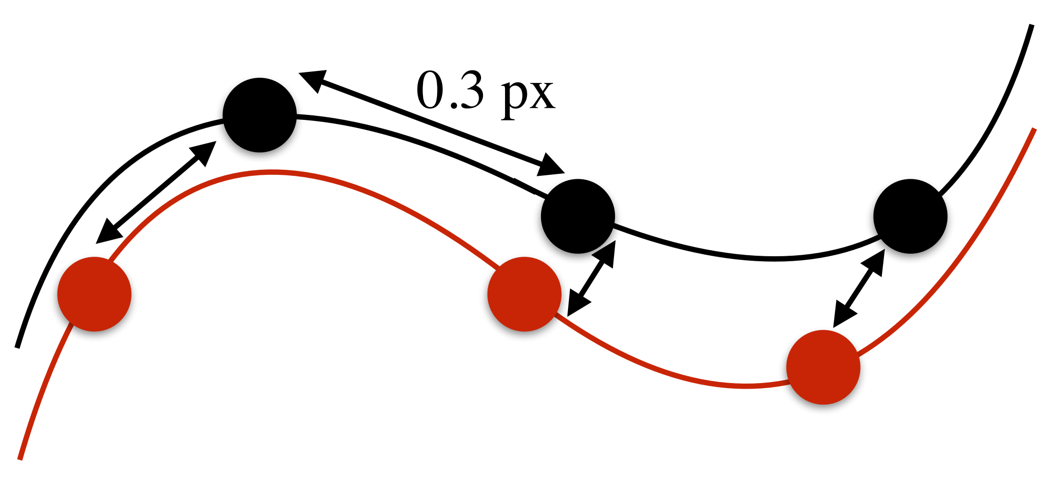
Our Approach



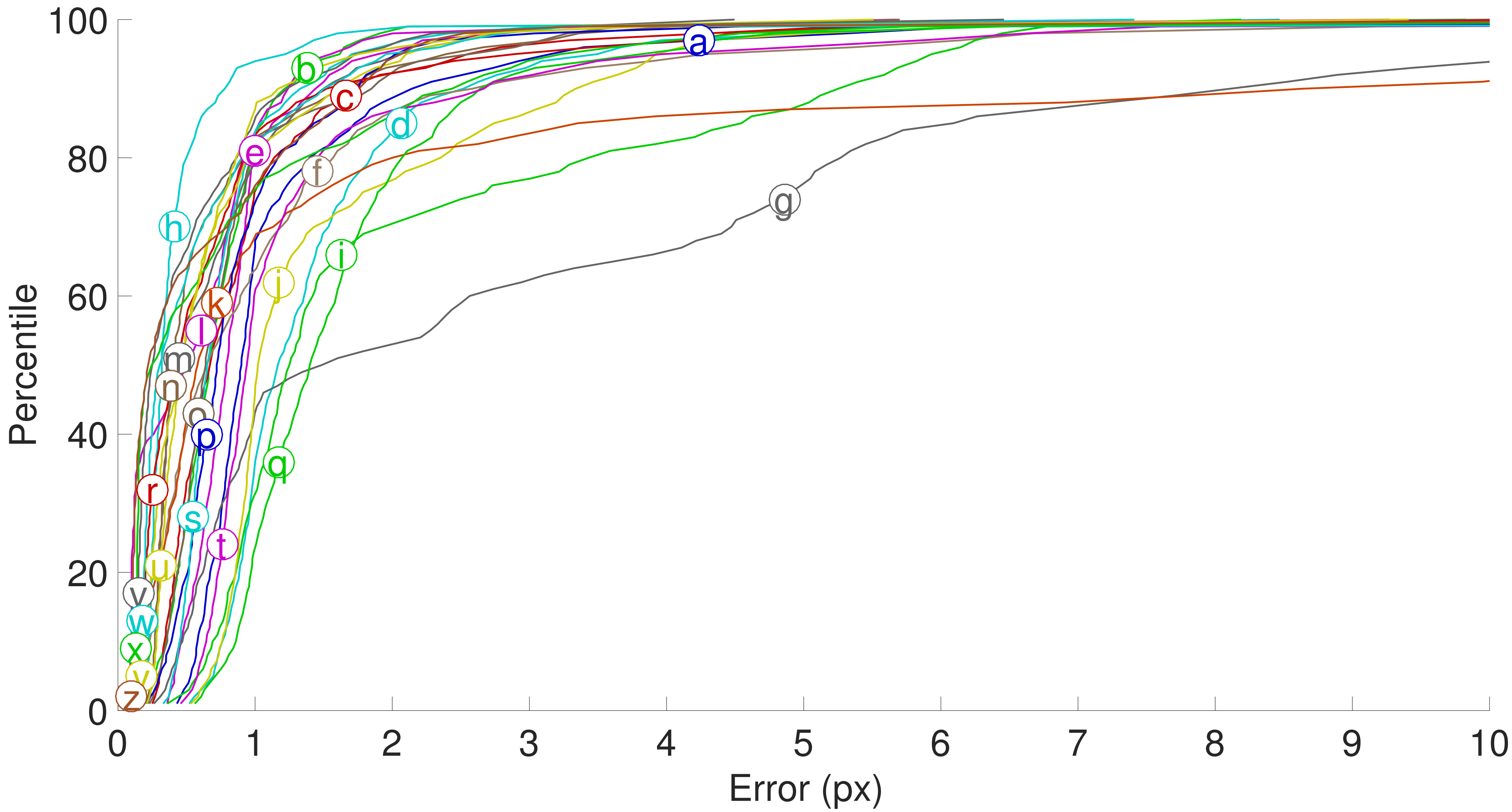
Fitting Results

Fitting Results

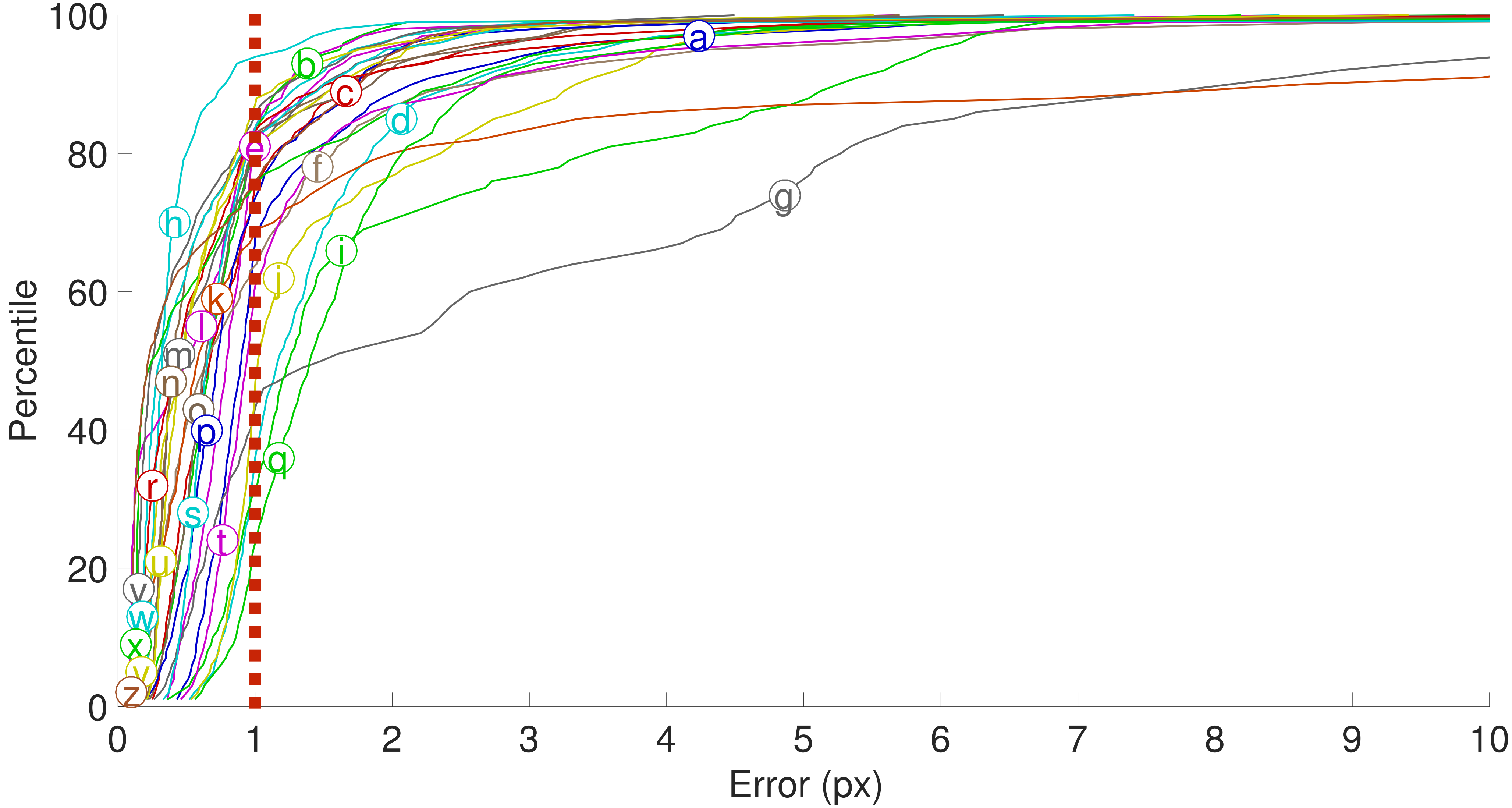




Fitting Results

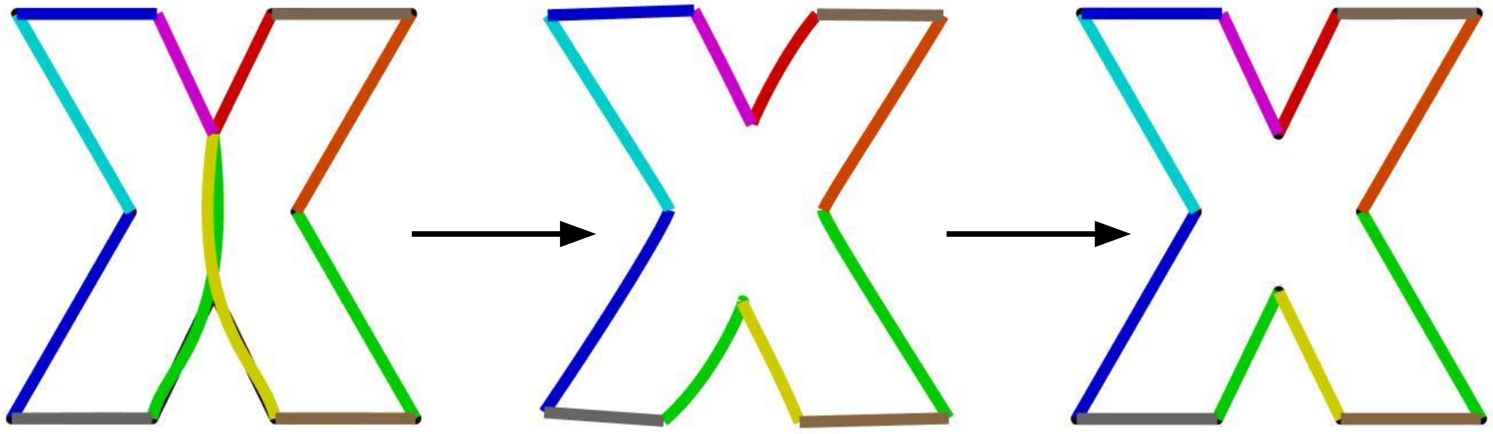


Fitting Results



Iterative Improvement Evaluation

Iterative Improvement Evaluation

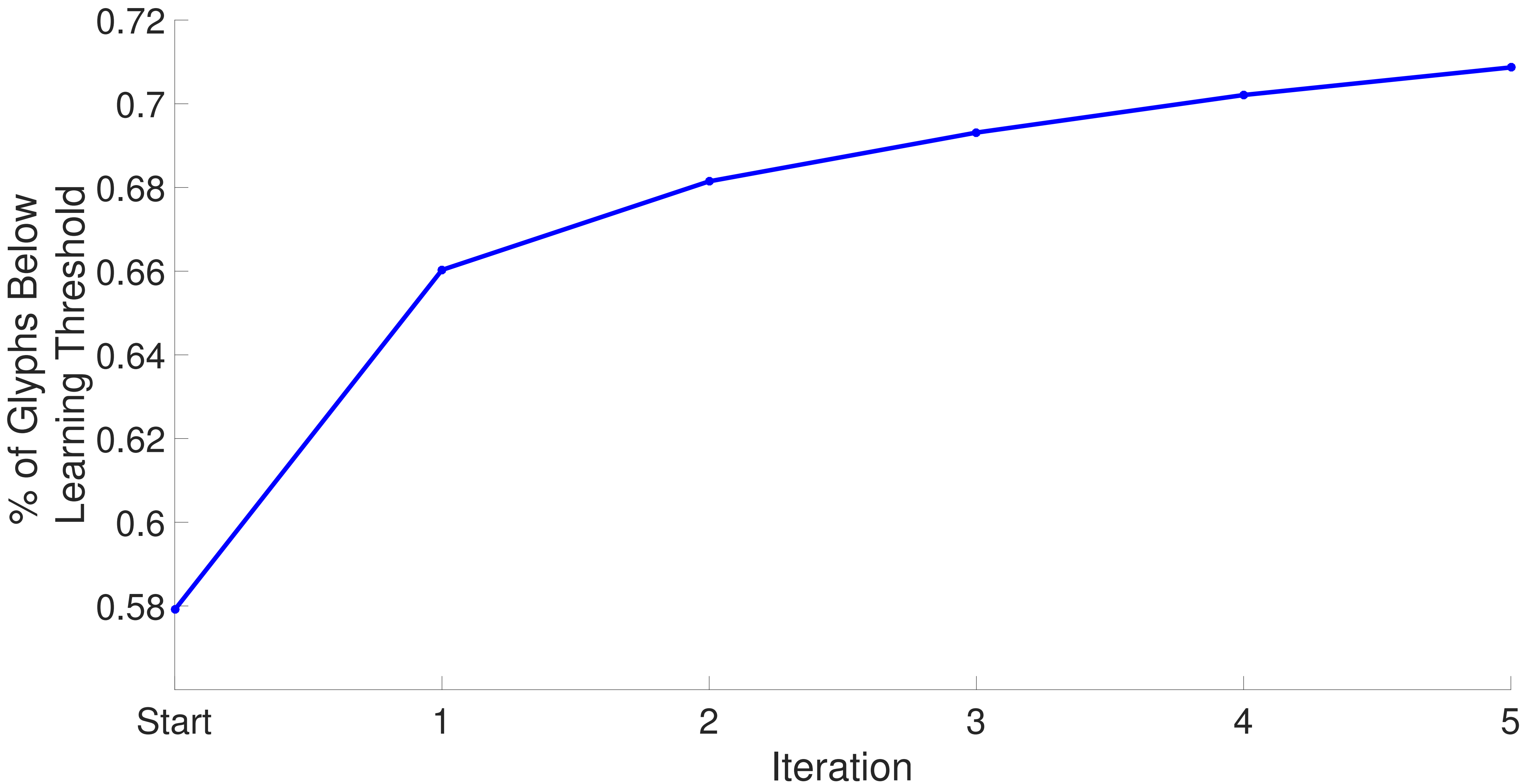


Initial Fit

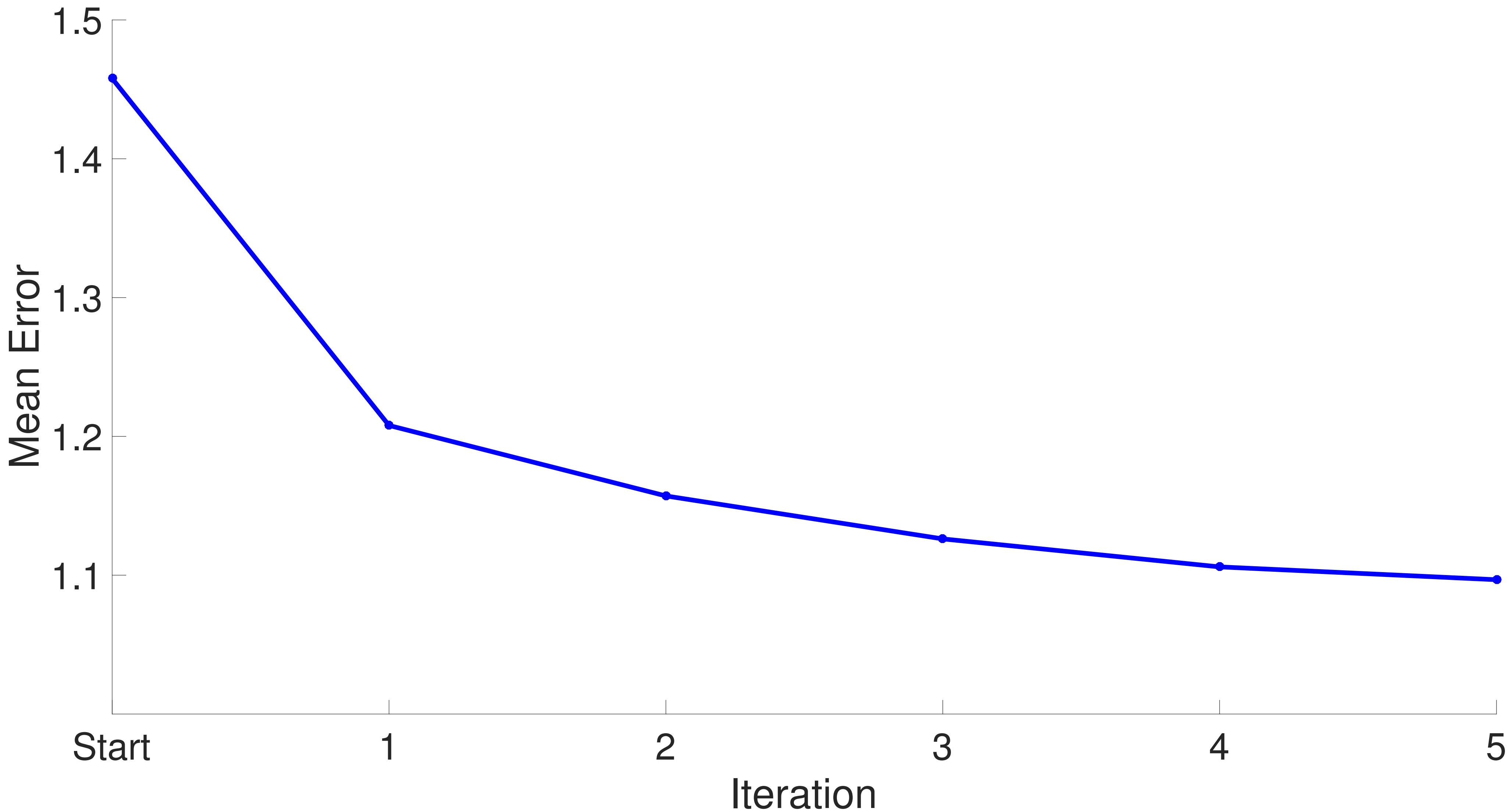
Manifold Guess

Improved Fit

Iterative Improvement Evaluation



Iterative Improvement Evaluation

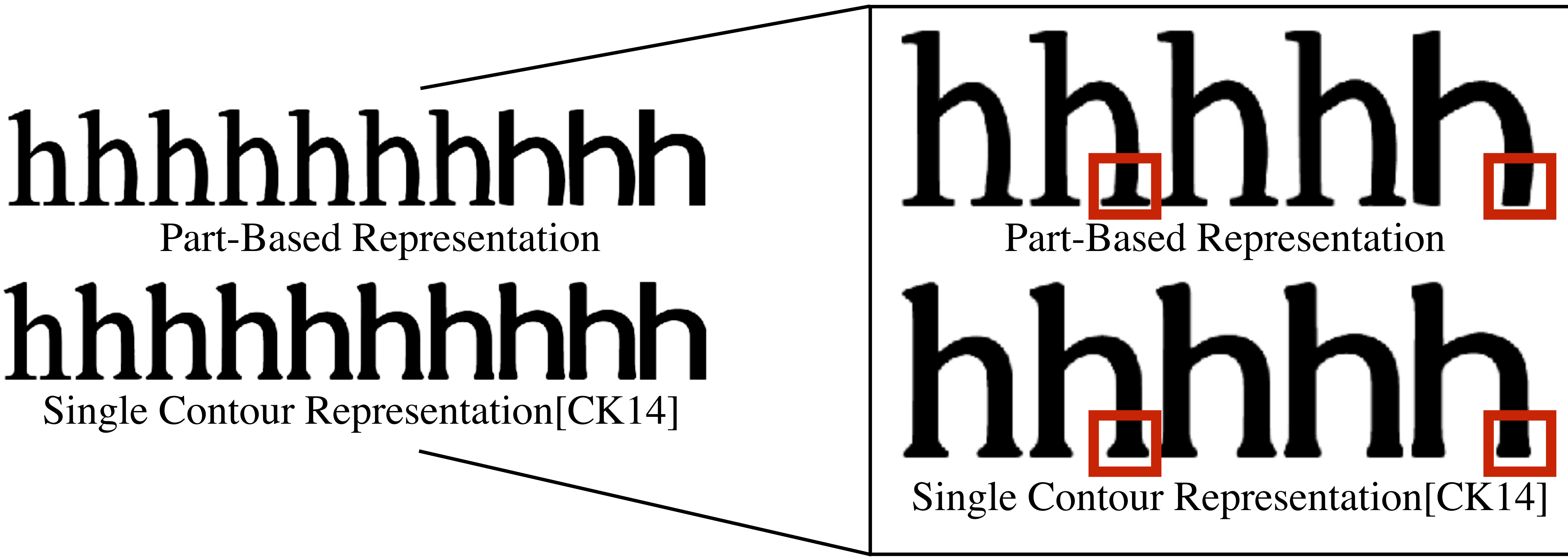


Comparison to Non-Part-Aware Approach

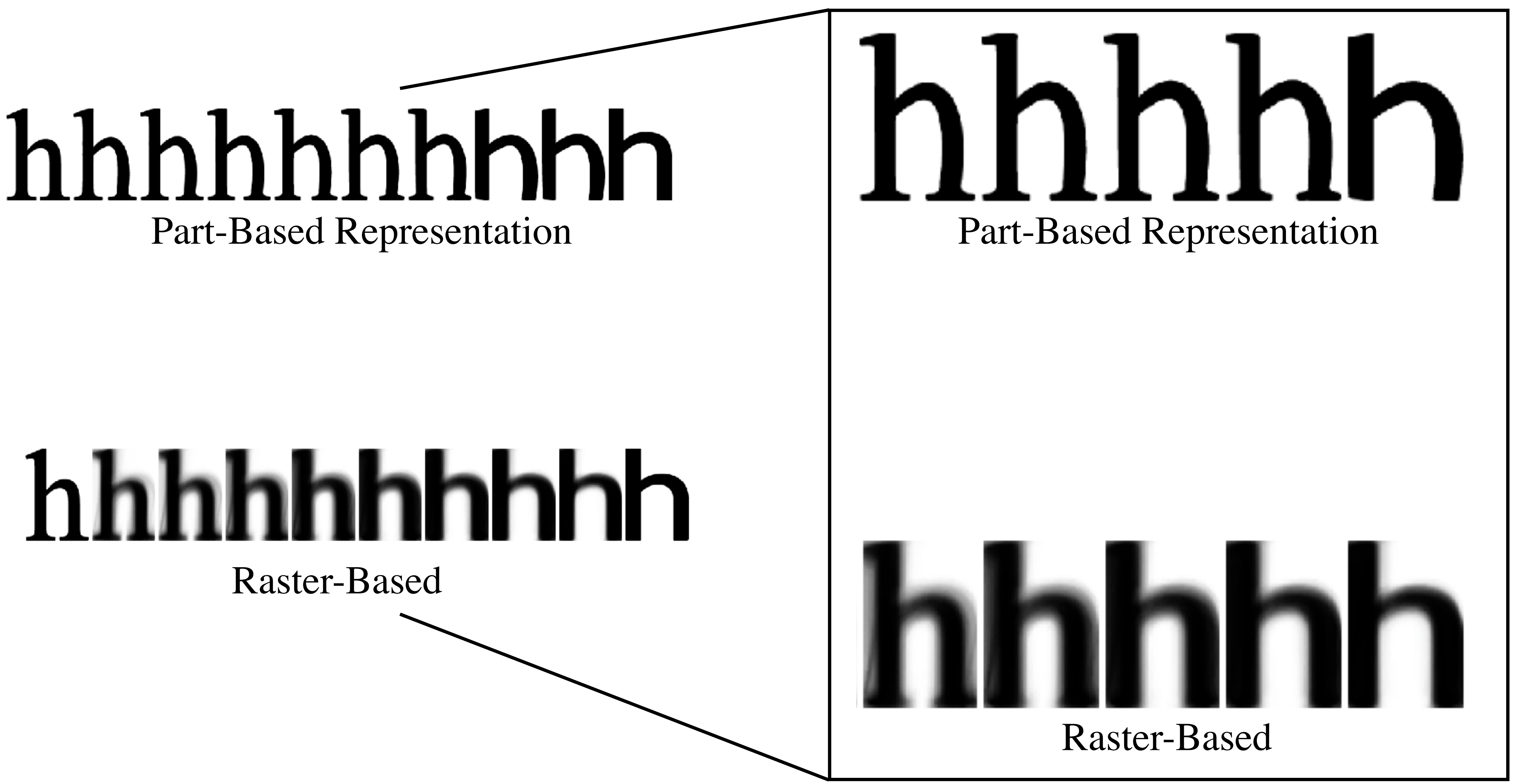
Comparison to Non-Part-Aware Approach

Part-Based Representation

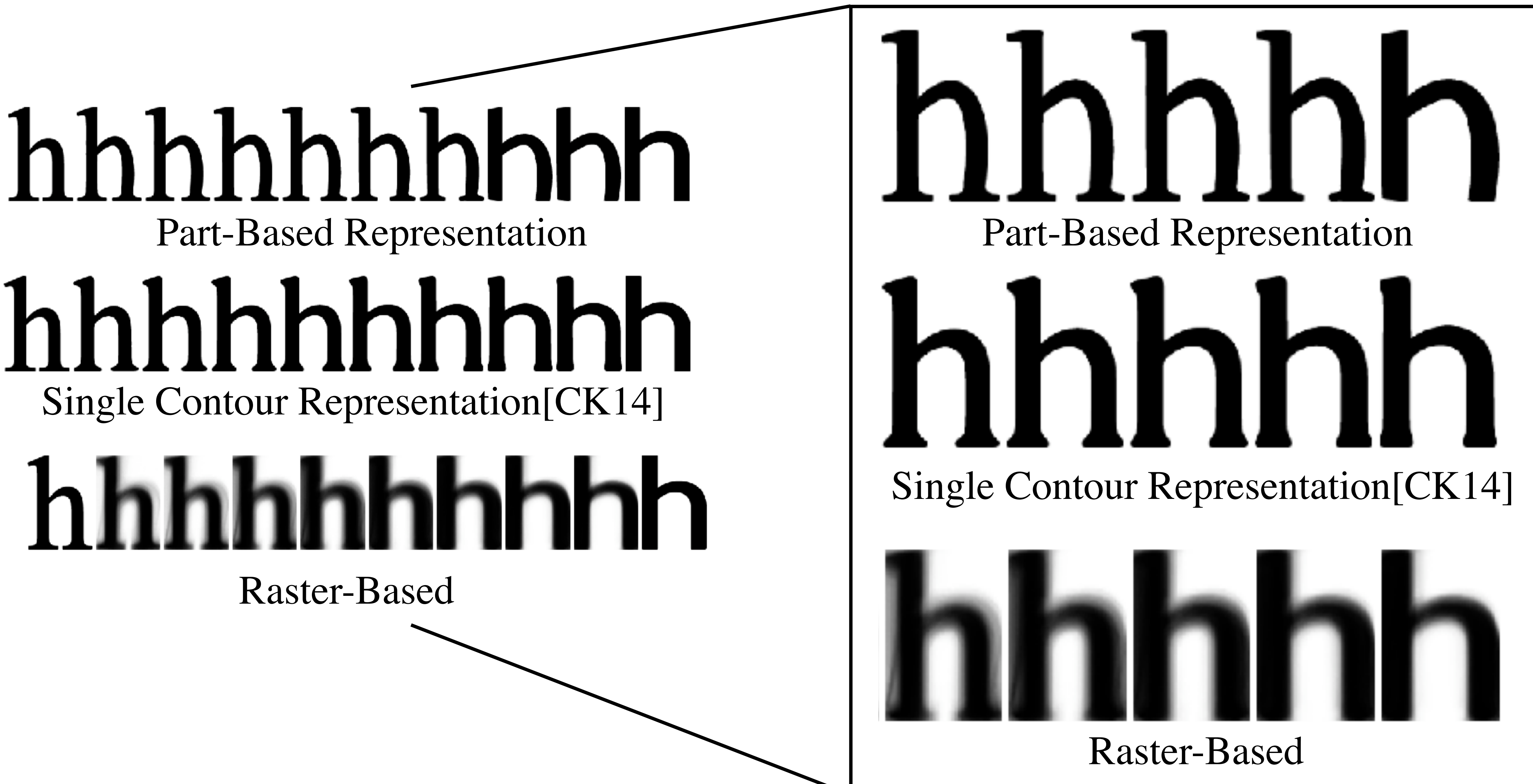
Comparison to Non-Part-Aware Approach



Comparison to Non-Part-Aware Approach

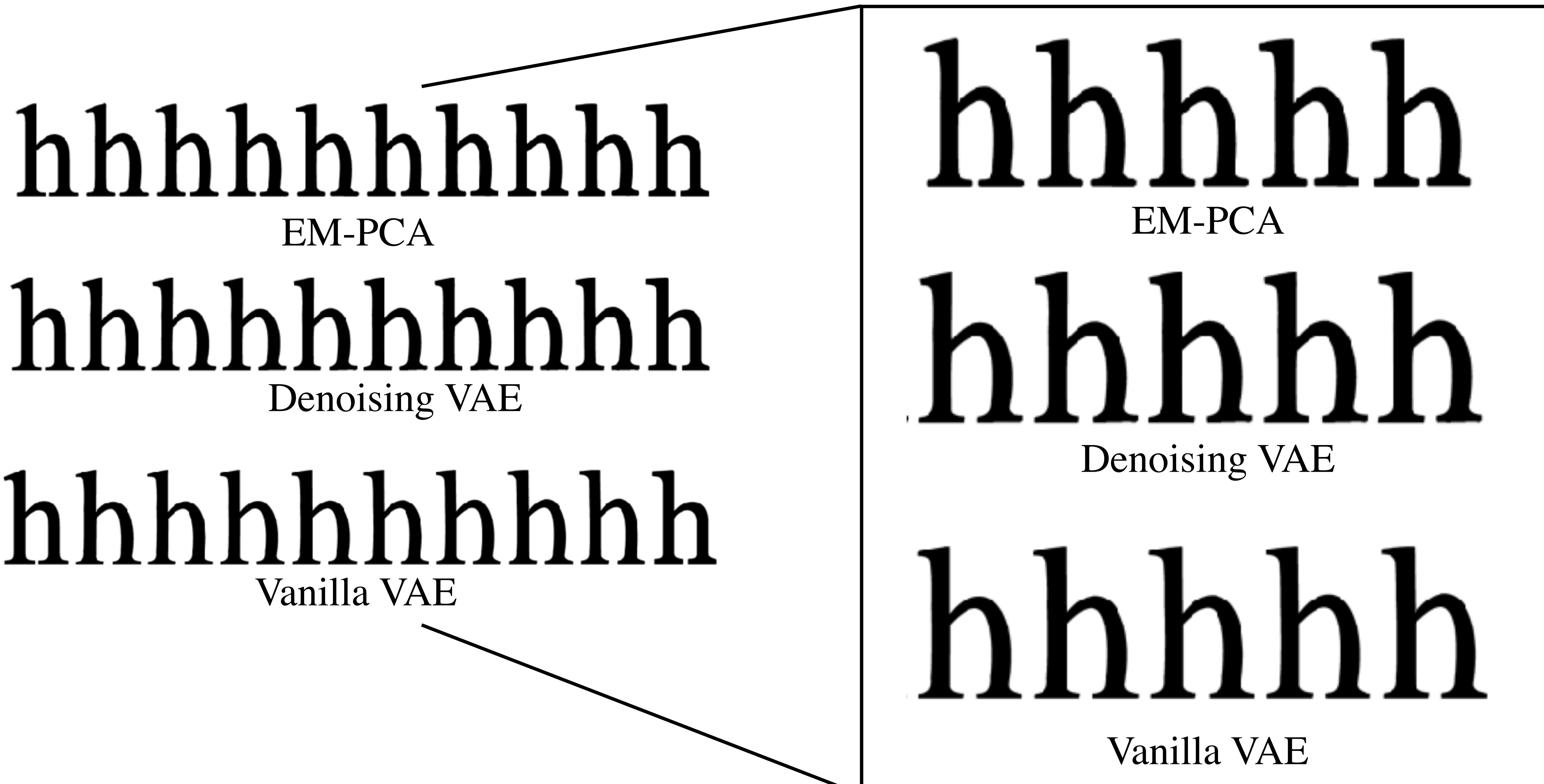


Comparison to Non-Part-Aware Approach

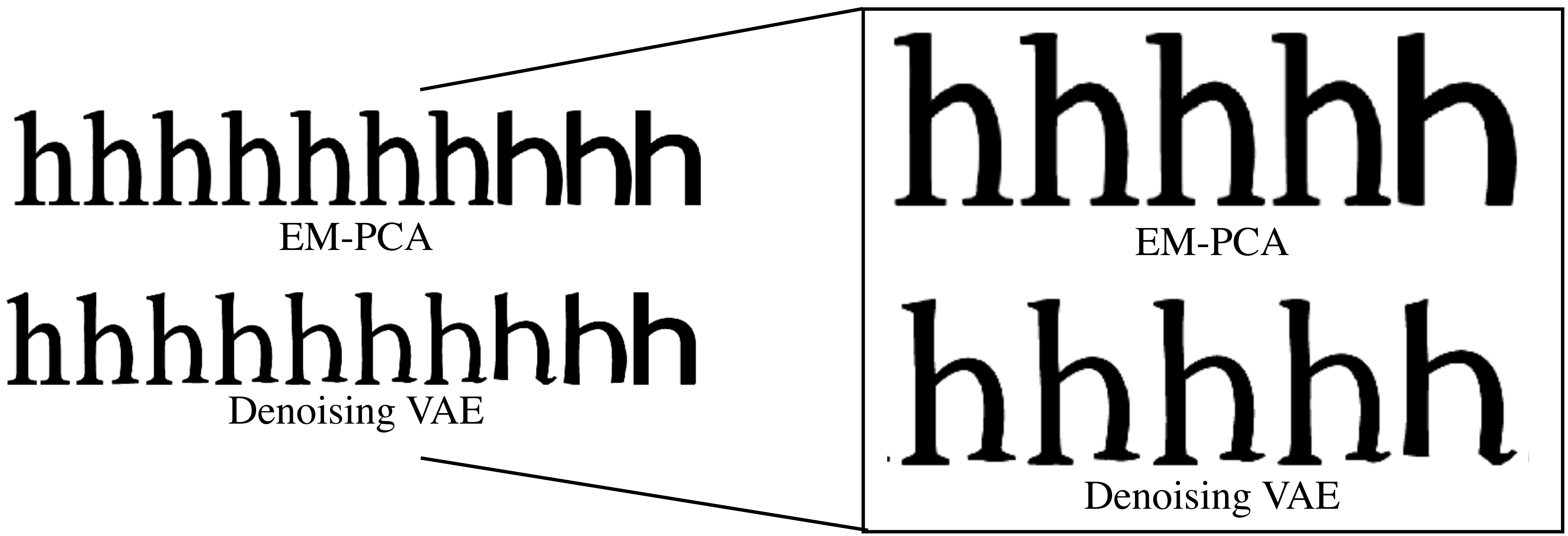


Generative Model Comparison

Generative Model Comparison



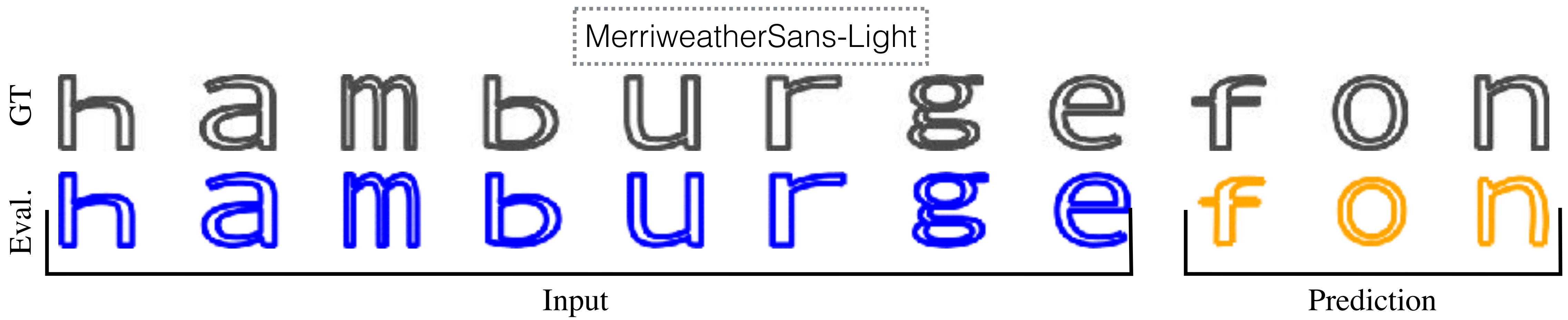
Generative Model Comparison

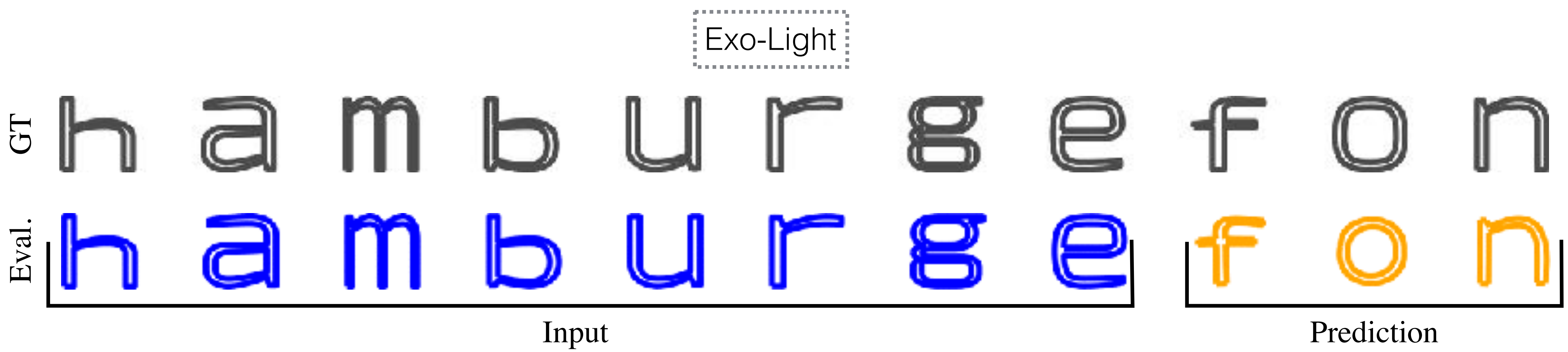


Applications

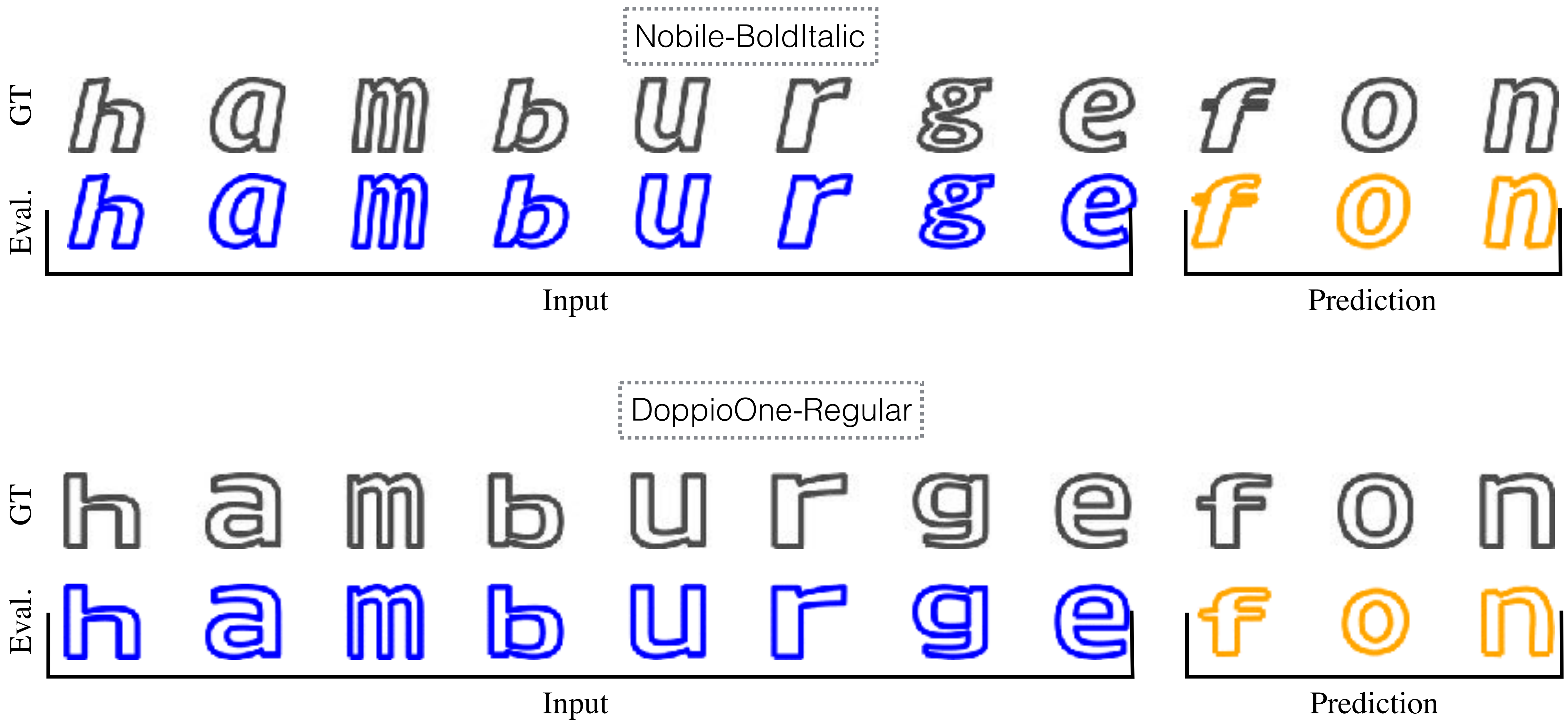
Style Completion

Style Completion: Light Fonts

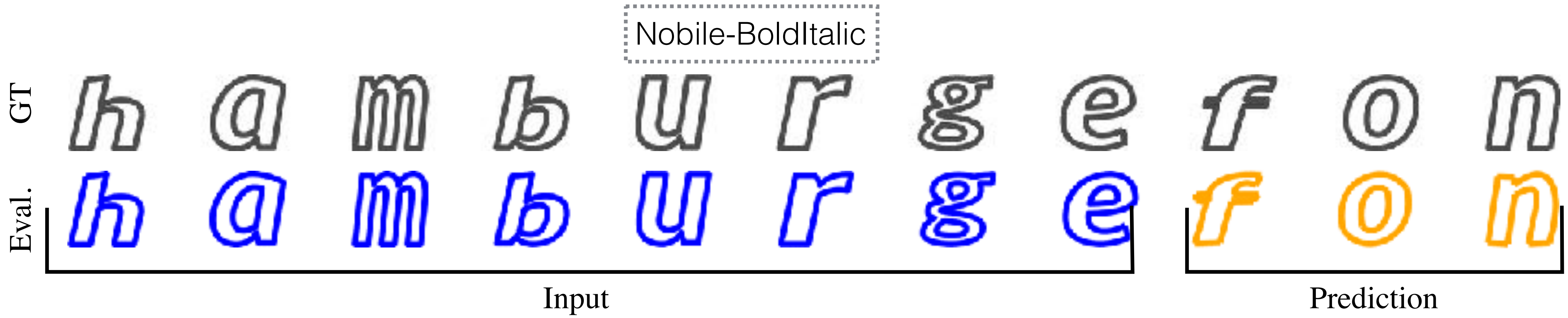


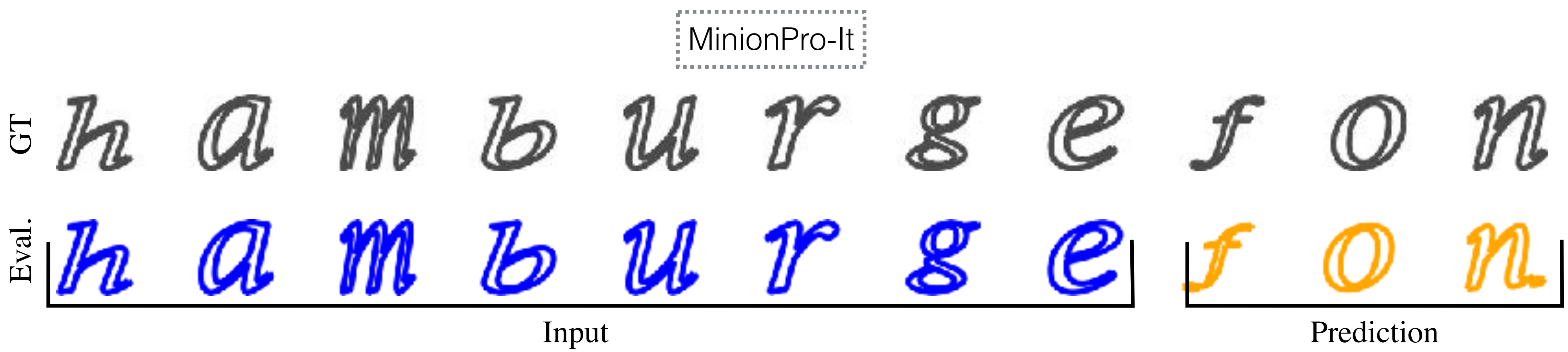


Style Completion: Regular/Bold Fonts

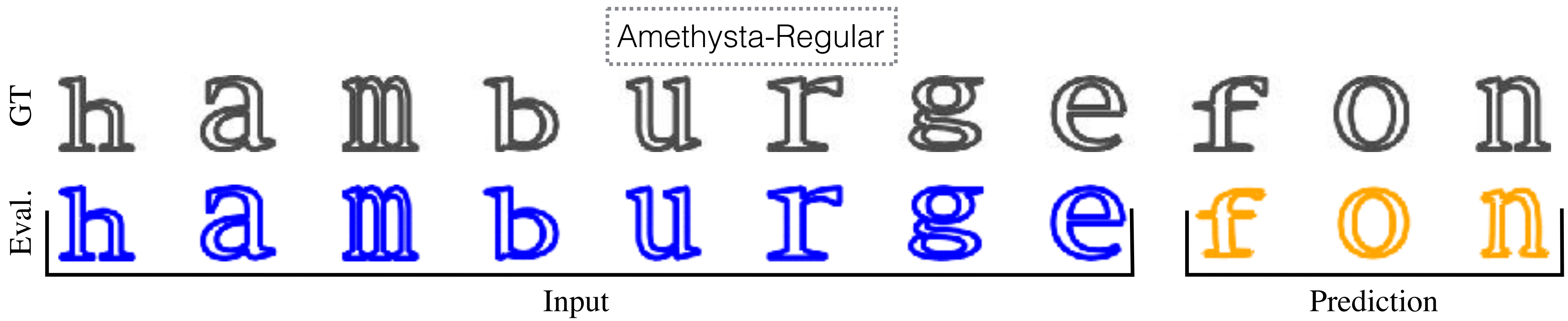


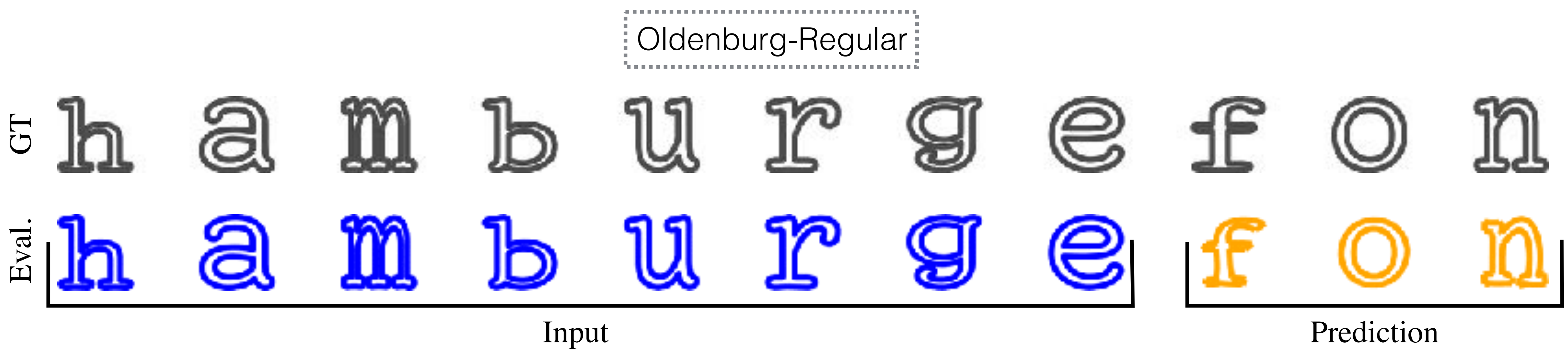
Style Completion: Italic Fonts



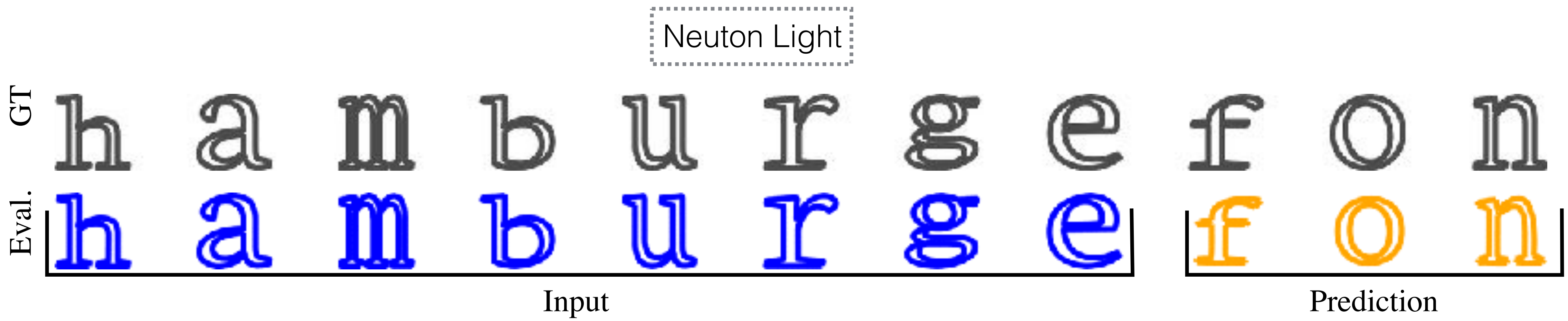


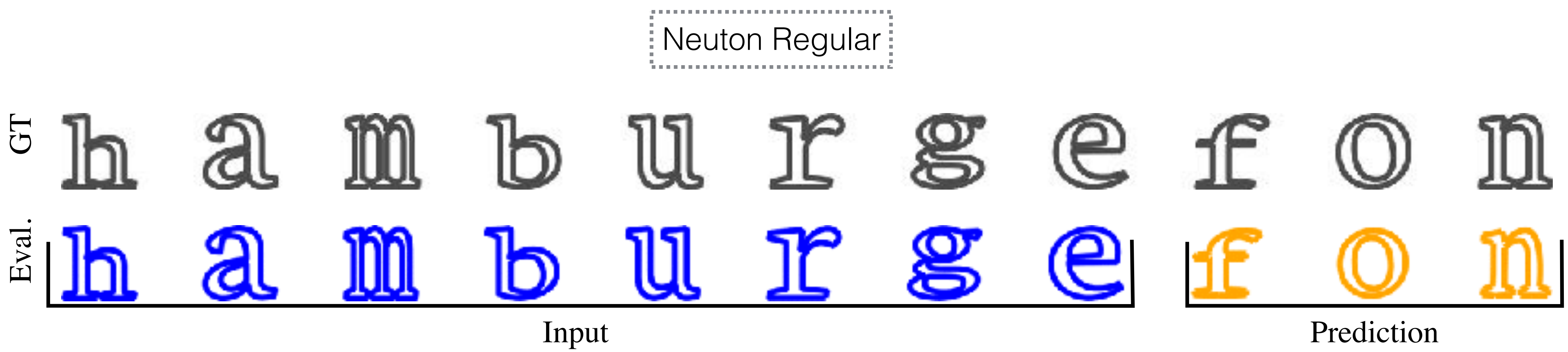
Style Completion: Serif Fonts





Style Completion: Serif Fonts





Style Completion

Style Completion

Style Completion

Style Completion

Style Completion

Style Completion

Style Completion

Style Completion

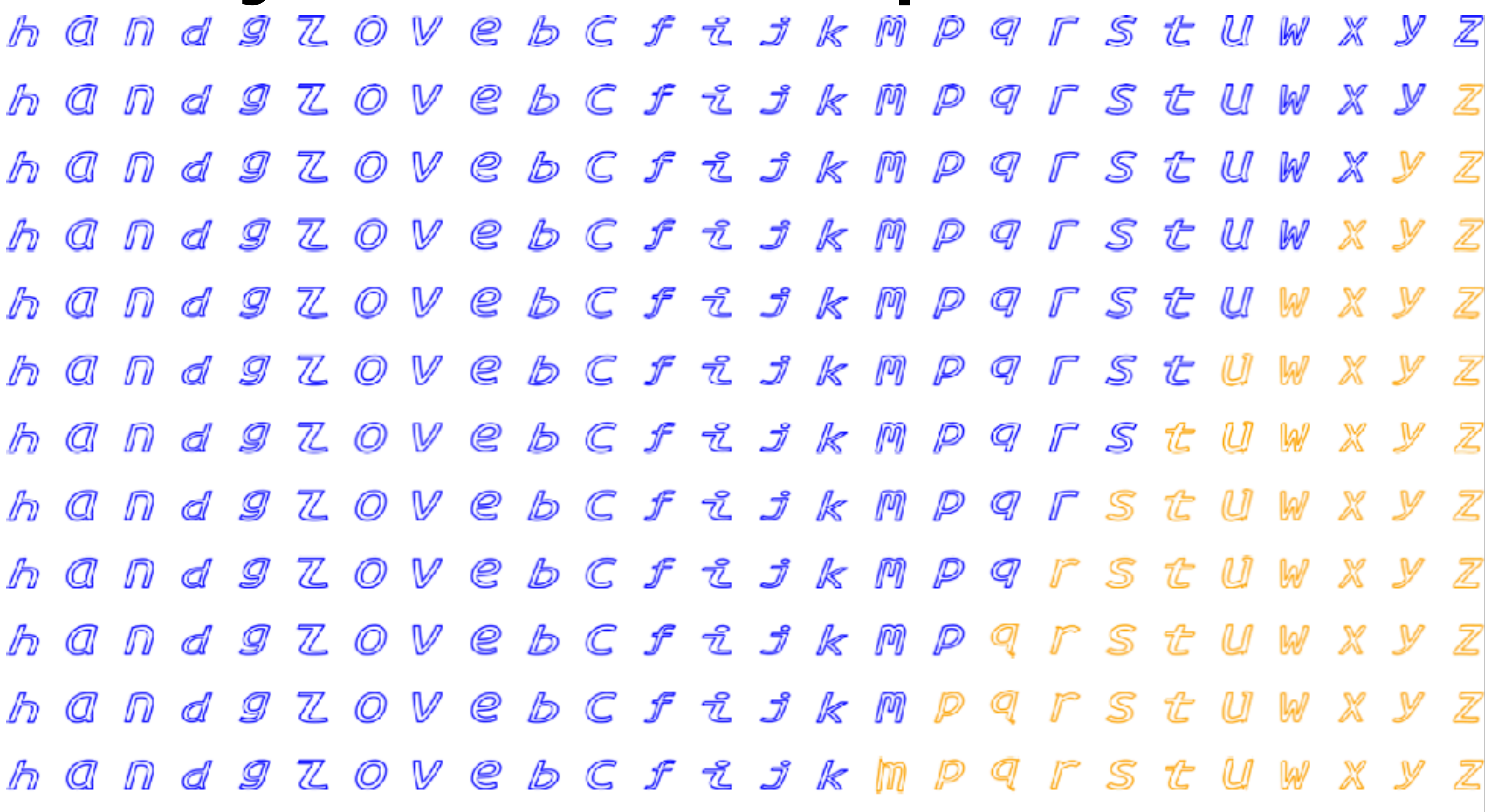
Style Completion

Style Completion

Style Completion

Style Completion

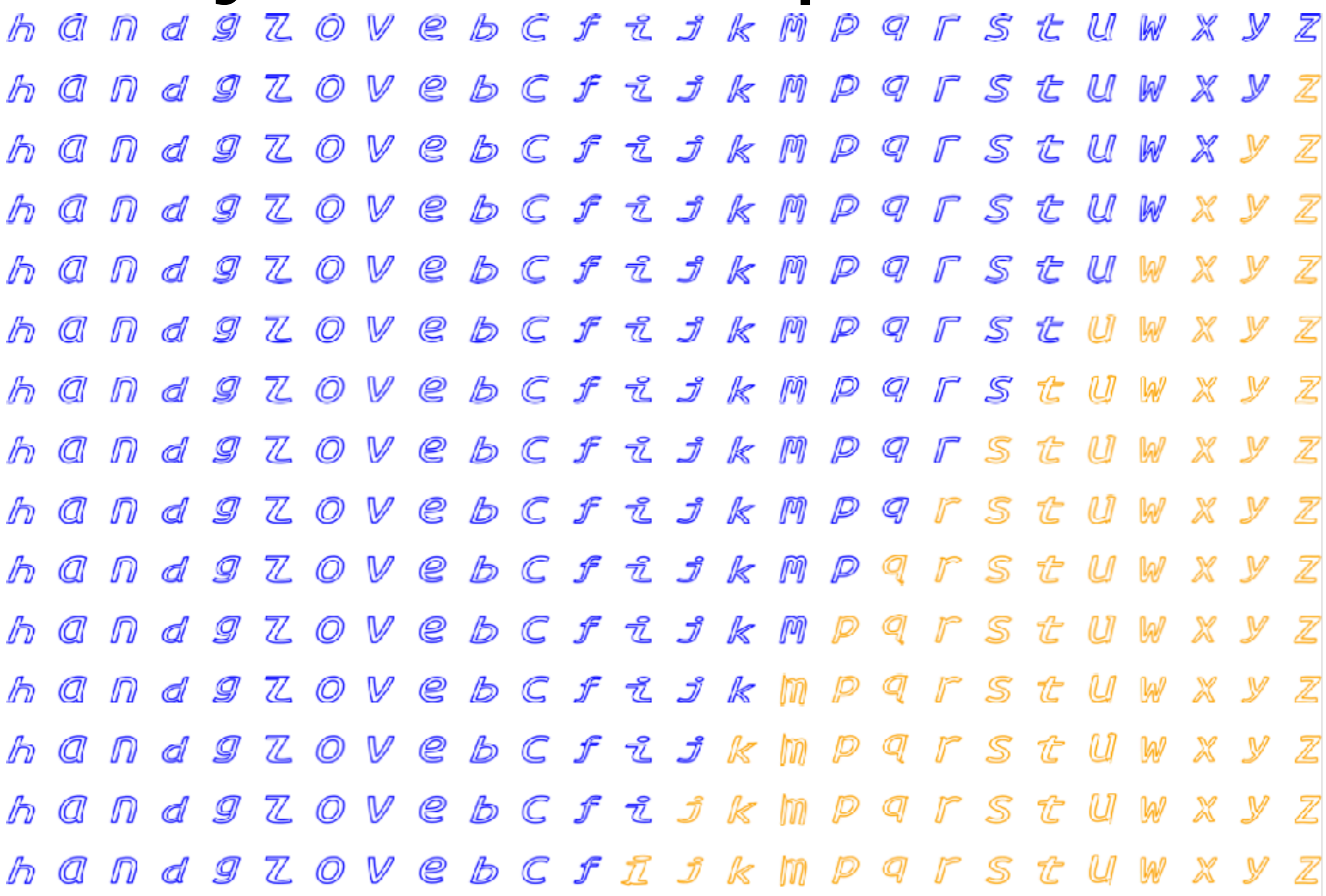
Style Completion



Style Completion

Style Completion

Style Completion



Style Completion

Style Completion

Style Completion

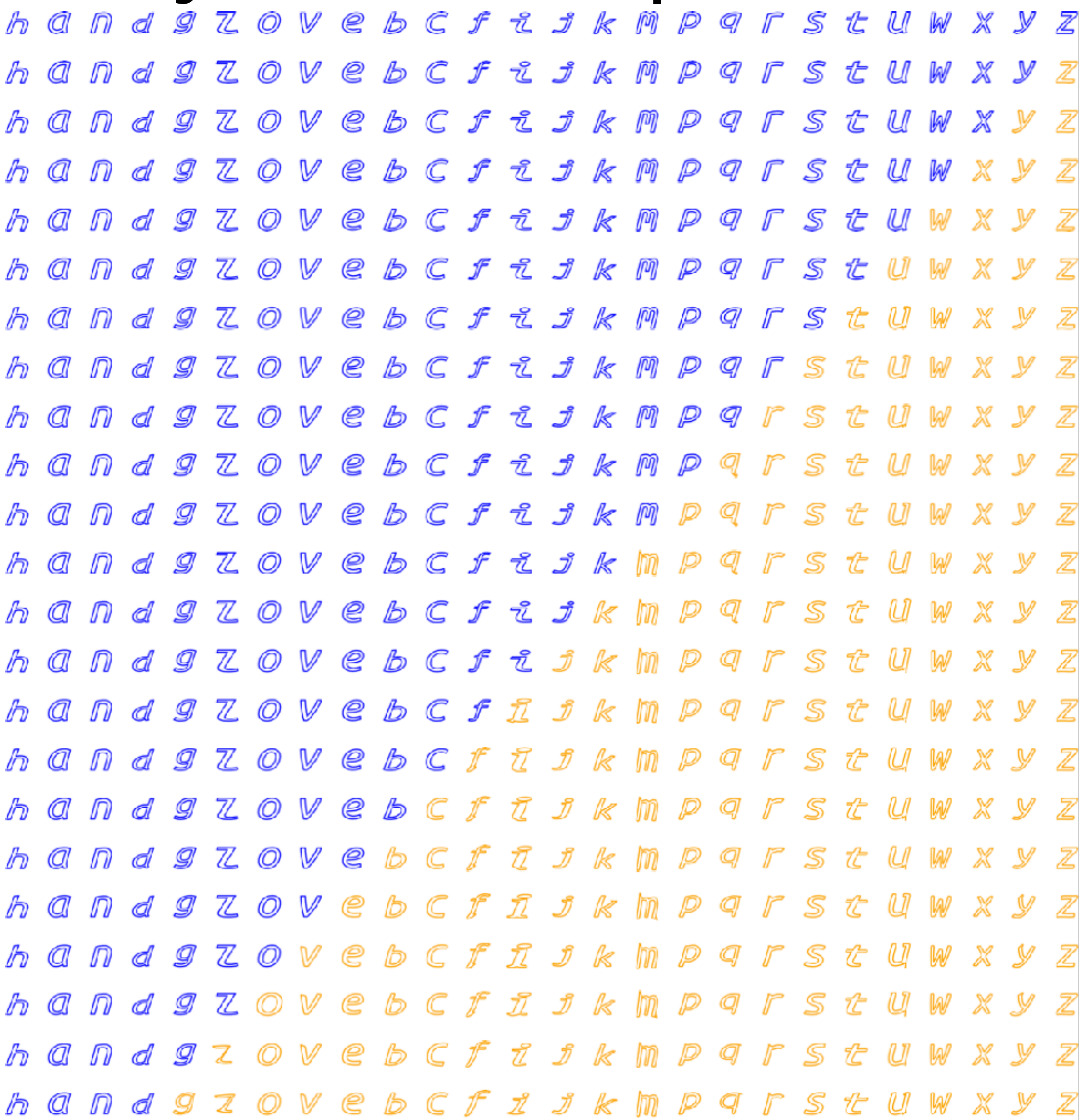
Style Completion

Style Completion

Style Completion

Style Completion

Style Completion



Style Completion

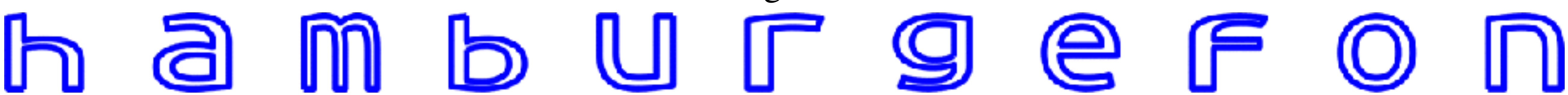
Style Completion

Font Retrieval

Font Retrieval

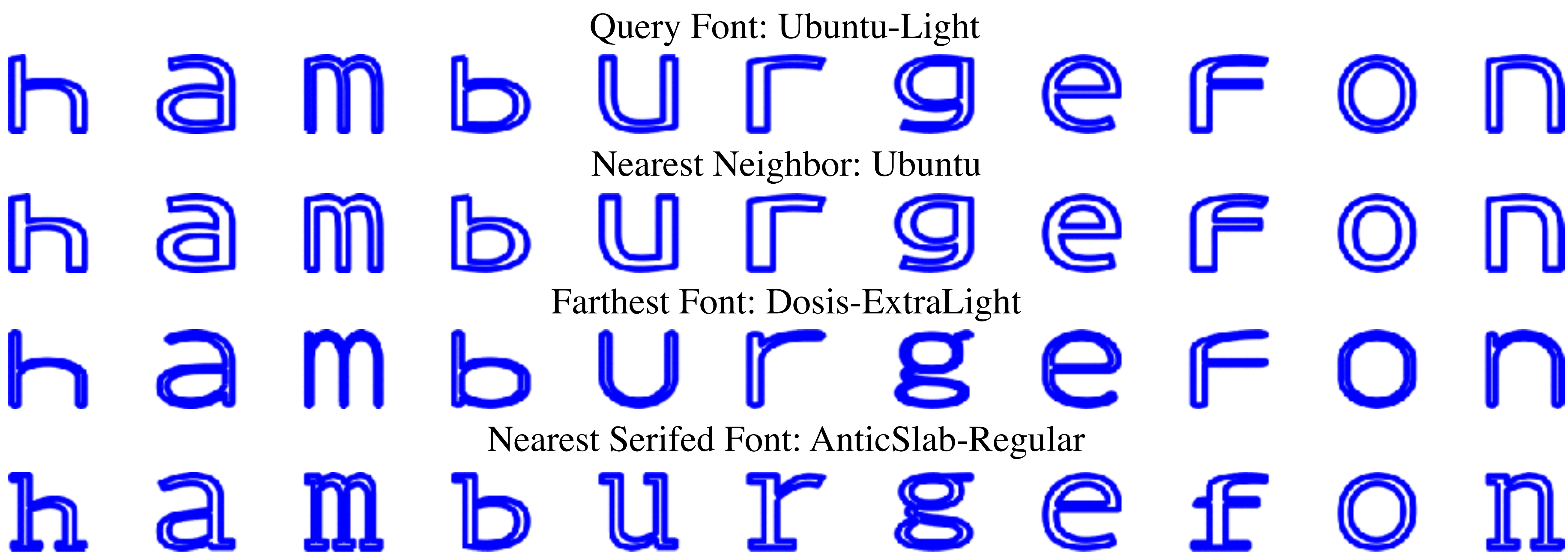
Query Font: Ubuntu-Light

Nearest Neighbor: Ubuntu



Farthest Font: Dosis-ExtraLight

Topology-Aware Retrieval



Topology-Aware Retrieval

Query Font: Ubuntu-Light

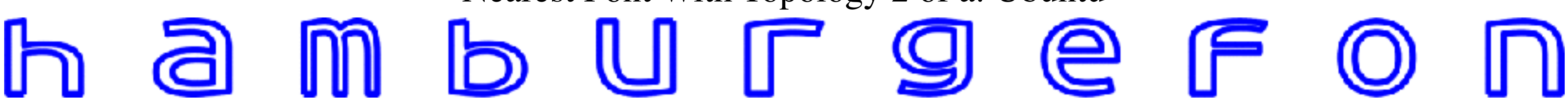
Nearest Neighbor: Ubuntu

Farthest Font: Dosis-ExtraLight

Nearest Serifed Font: AnticSlab-Regular

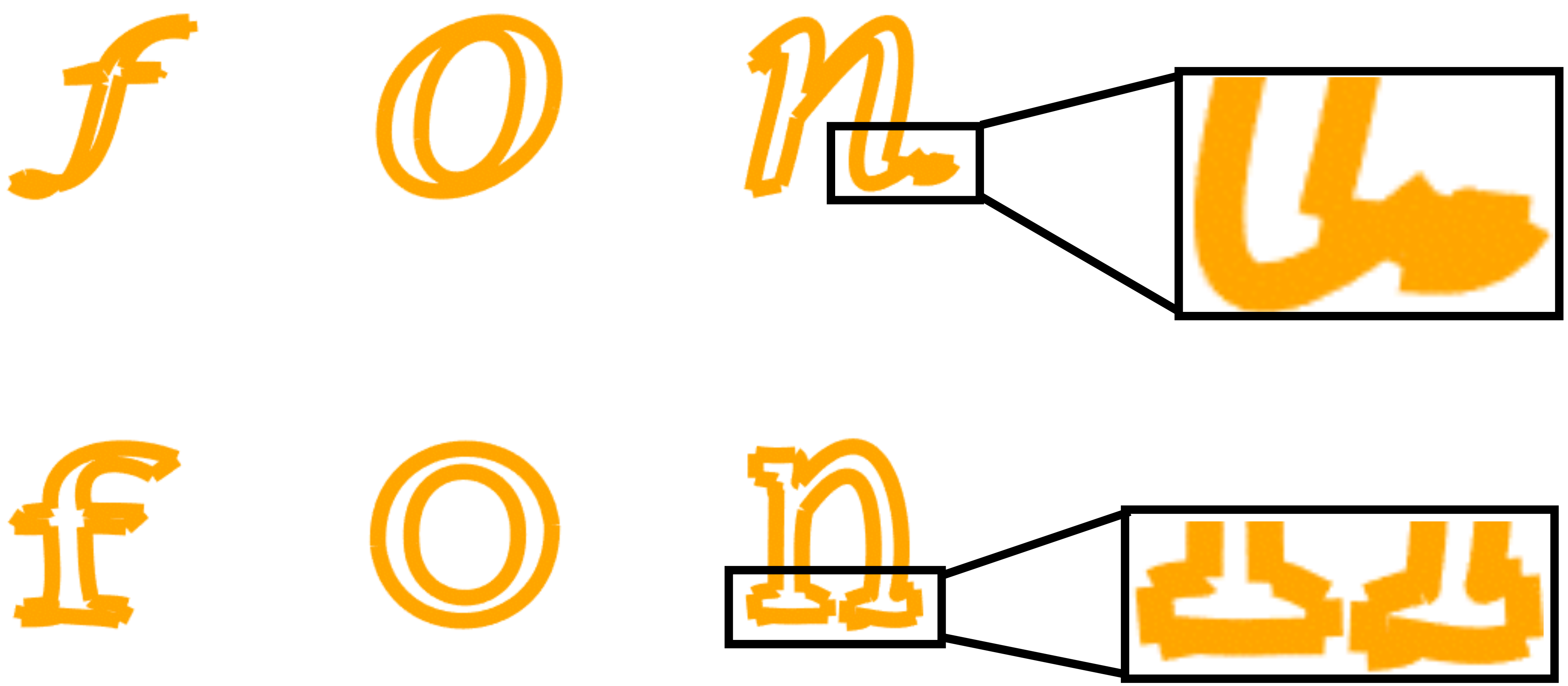
Nearest Font With Topology 1 of a: Inder-Regular

Nearest Font With Topology 2 of a: Ubuntu



Limitations

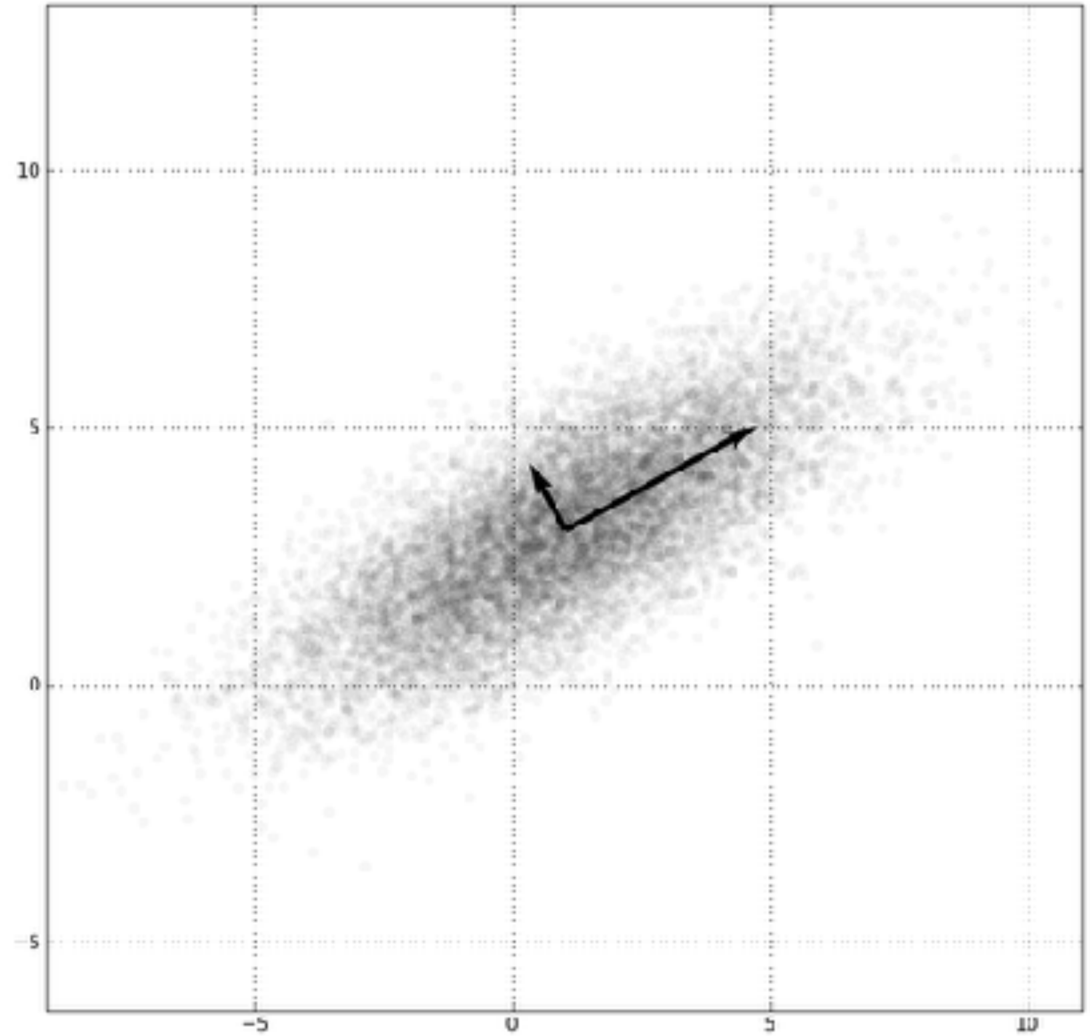
Limitations



Limitations

[https://design.google]

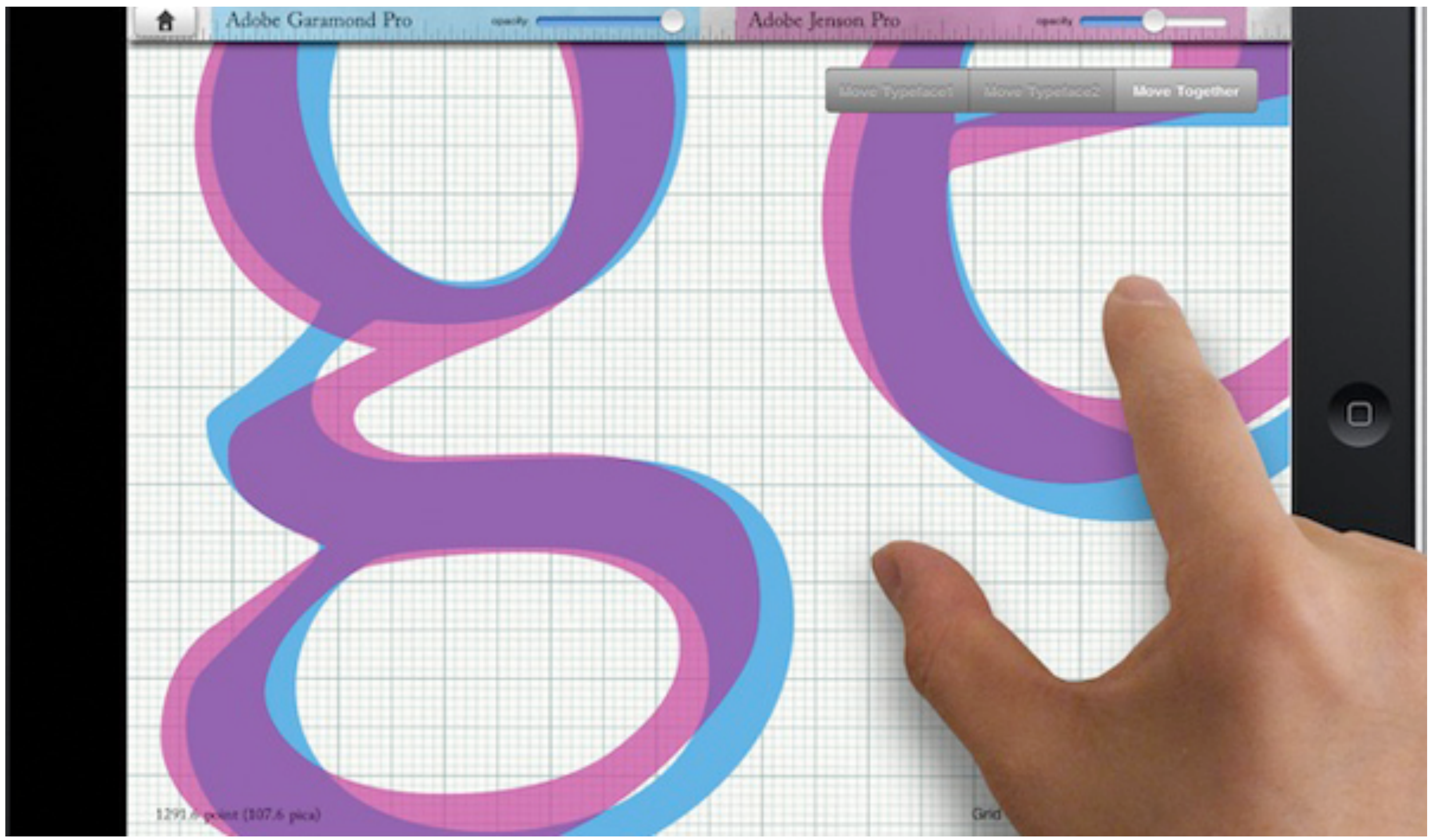
Limitations



[https://inventingsituations.net]

Future Work

Future Work



[https://www.theatlantic.com]

Future Work

[https://www.123rf.com/]

Future Work

[Nejati '16]

Acknowledgments

- Princeton Graphics and Vision groups, Ariel Shamir, anonymous reviewers
- Princeton CS Staff (Asya Dvorkin)
- Support: National Science Foundation Graduate Fellowship (NSF GRFP),NSF CRI 1729971 and Adobe

Thank You!