Multiple Heads are Better than One: Few-shot Font Generation with Multiple Localized Experts

tags: Yonsei University, NAVER AI Lab, NAVER CLOVA

arxive: https://arxiv.org/pdf/2104.00887.pdf
The unified few-shot font generation repository: https://github.com/clovaai/fewshot-font-generation

> clovaai/fewshot-font-generation included:
> **FUNIT** (Liu, Ming-Yu, et al. ICCV 2019) : not originally proposed for FFG tasks, but we modify the unpaired i2i framework to the paired i2i framework for FFG tasks.
> **DM-Font** (Cha, Junbum, et al. ECCV 2020) : proposed for complete compositional scripts (e.g., Korean). If you want to test DM-Font in Chinese generation tasks, you have to modify the code (or use other models).
> **LF-Font** (Park, Song, et al. AAAI 2021) : originally proposed to solve the drawback of DM-Font, but it still require component labels for generation. Our implementation allows to generate characters with unseen component.
> **MX-Font** (Park, Song, et al. ICCV 2021) : generating fonts by employing **multiple experts** where each expert focuses on different local concepts.

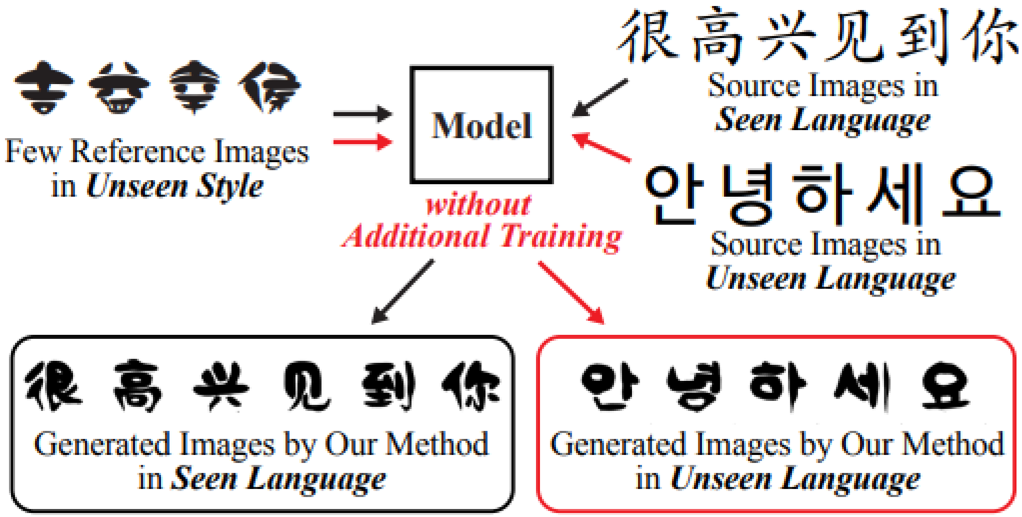


Figure 1. **Cross-lingual few-shot font generation results by MX-Font.** With only four references, the proposed method, MX-Font, can generate a high quality font library. Furthermore, we first show the effectiveness of the proposed method on the *zero-shot cross-lingual* few-shot generation task, *i.e.*, generating unseen Korean glyphs using the Chinese font generation model.

Abstract

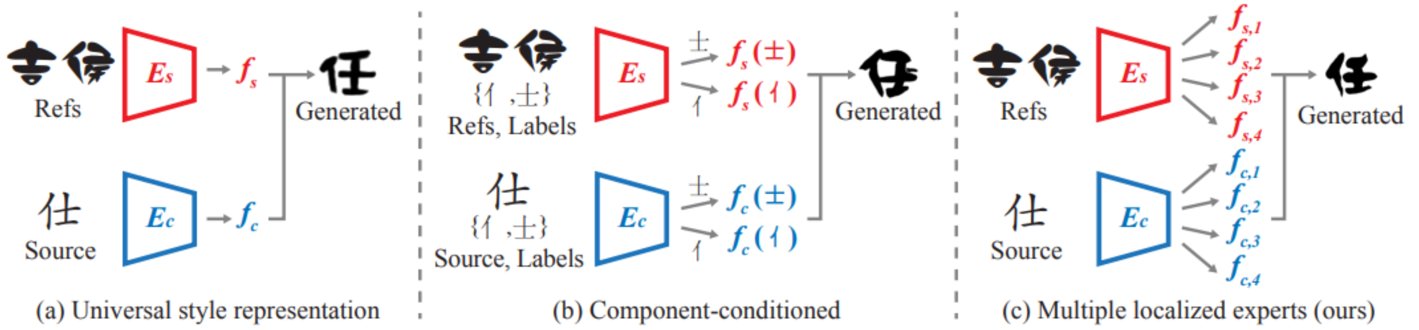
Existing FFG methods aim to disentangle content and style either by extracting **a universal representation style** or extracting **multiple component-wise style** representations.

However, previous methods either fail to capture **diverse local styles** or cannot be **generalized to a character with unseen components**, e.g., unseen language systems.

To mitigate the issues, we propose a novel FFG method, named **Multiple Localized Experts Few-shot Font Generation Network (MXFont)**. MX-Font extracts multiple style features not ~~**explicitly conditioned**~~ on component labels, but automatically by **multiple experts** to represent different local concepts, e.g., left-side sub-glyph.

During training, we utilize component labels as **weak supervision** to guide each expert to be specialized for different local concepts.

We also employ the **independence loss** and the **content-style adversarial** loss to impose the contentstyle disentanglement.



(a) Universal style representation

(b) Component-conditioned

(c) Multiple localized experts (ours)

Figure 2. **Comparison of FFG methods.** Three different groups of FFG are shown. All methods combine style representation f_s from a few reference glyphs (Refs) by a style encoder (E_s) and content representation f_c from a source glyph (Source) by a content encoder (E_c). (a) Universal style representation methods extract only a single style feature for each font. (b) Component-conditioned methods extract component conditioned style features to capture diverse local styles (c) Multiple localized experts method (ours) generates multiple local features without an explicit condition, but attends different local information of the complex input glyph. The generated images in (a), (b) and (c) are synthesized by AGIS-Net [12], LF-Font [37] and MX-Font, respectively.

Related Works

The universal style representation shows limited performances in capturing localized styles and content structures. To address the issue, **component-conditioned methods** such as DM-Font [6], LFFont [37], remarkably improve the stylization performance by employing localized style representation, where the font style is described multiple localized styles instead of a single universal style.

However, these methods require explicit component labels (observed during training) for the target character even at the test time. This property limits practical usages such as cross-lingual font generation. Our method inherits the advantages from component-guided multiple style representations, but **does not require the explicit labels** at the test time.

Method

Model architecture

- Our method consists of three modules:
    1. k-headed encoder, or localized experts E_i
    2. a generator G
    3. style and component feature classifiers Cls_s and Cls_u.
- The localized expert E_i encodes a glyph image x into a local feature f_i.

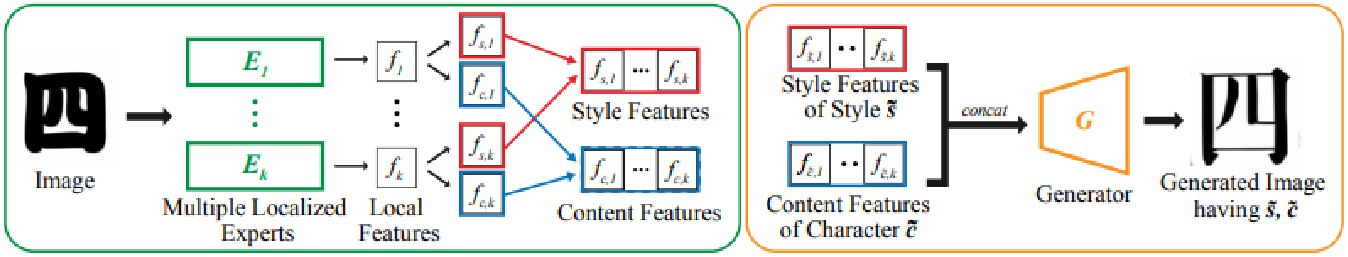


Image
E_1
E_k
Multiple Localized Experts
f_1
f_k
Local Features
$f_{s,1}$
$f_{c,1}$
$f_{s,k}$
$f_{c,k}$
$f_{s,1}$ ··· $f_{s,k}$
Style Features
$f_{c,1}$ ··· $f_{c,k}$
Content Features
$f_{\tilde{s},1}$ ·· $f_{\tilde{s},k}$
Style Features of Style $\tilde{s}$
$f_{\tilde{c},1}$ ·· $f_{\tilde{c},k}$
Content Features of Character $\tilde{c}$
concat
G
Generator
Generated Image having $\tilde{s}$, $\tilde{c}$

Figure 3. **Overview of MX-Font.** Two modules of MX-Font used for the generation are described. The *multiple localized experts* (green box) consist of k experts. E_i (*i.e.* i-th expert) encodes the input image to the i-th local feature f_i and the i-th style and content feature $f_{s,i}$, $f_{c,i}$ are computed from f_i. The right yellow box shows how the generator G generates the target image. When k style features representing the target style $\tilde{s}$ and k content features representing the target style $\tilde{c}$ are given, the target glyph having style $\tilde{s}$ and character $\tilde{c}$ is generated by passing the element-wisely concatenated style and content features to the G.

Here, our localized experts are not supervised by component labels to obtain k local features $f_1, \ldots, f_k$; our local features are not component-specific features. We set the number of the **localized experts, k, to 6** in our experiments if not specified.

- We employ two feature classifiers, Cls_s and Cls_u to supervise $f_{s,i}$ and $f_{c,i}$, which serve as **weak supervision** for f_i.

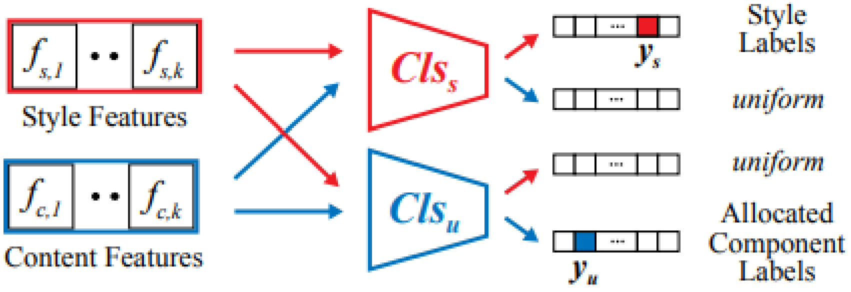


Figure 5. **Feature classifiers.** Two feature classifiers, Cls_s and Cls_u are used during the training. Cls_s classifies the style features to their style label y_s while Cls_u predicts the uniform probability from them. Similarly, Cls_u classifies the content features to their allocated component labels y_u while Cls_s is fooled by them. The details are described in § 3.2 and § 3.3.

- These classifiers are only used during training but independent to the model inference itself. Following the previous methods [6, 37], we use font library labels for style labels y_s, and the component labels U_c for content labels y_c.

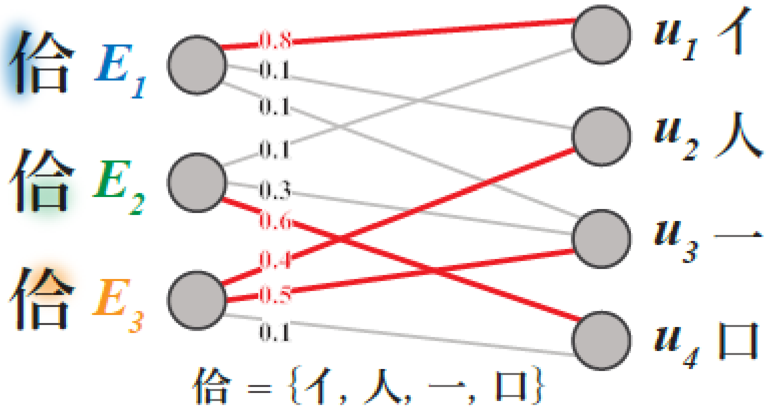


Figure 4. **An example of localized experts.** The number of experts k is three (E_1, E_2, E_3), and the number of target component labels m is four $(u_1, \ldots, u_4)$. An edge between an expert E_i and a component u_j means the prediction probability of u_j by E_i using the component classifier Cls_u. Our goal is to find a set of edges that maximizes the sum of predictions, where the number of the selected edges are upper bounded by $\max(k, m) = 4$ in this example. The red edges illustrate the optimal solution.

> The same **decomposition rule** used by **LF-Font** is adopted. While previous methods only use the style (or content) classifier to train style (or content), we additionally utilize them for the content and style disentanglement by introducing the content-style adversarial loss.

Learning multiple localized experts with weak local component supervision

- Because we do not want that an expert is explicitly assigned to a component label, e.g., strictly mapping “人” component to E1, we solve an automatic allocation algorithm, finding the optimal expert-component matching as shown in Figure 4.
- Specifically, we formulate the component allocation problem as the Weighted Bipartite B-Matching problem, which can be optimally solved by the **Hungarian algorithm**.

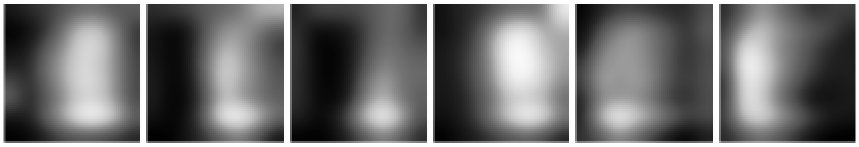
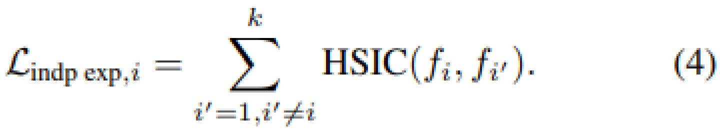
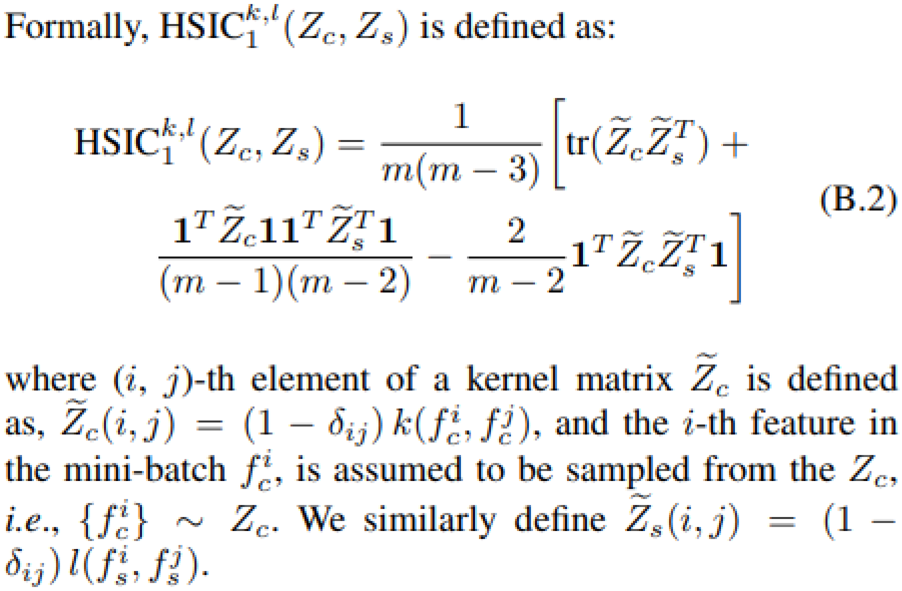


Figure 7. **Each localized expert attends different local areas.** We show the variance of Class Activation Maps (CAMs) on training images for each expert. The brighter intensity indicates that the variance of CAMs is higher in that region.

- We additionally formulate the independence between each expert by the **Hilbert-Schmidt Independence Criterion (HSIC)** [16] which has been used in practice for statistical testing [16, 17], feature similarity measurement [28], and model regularization [38, 51, 3].

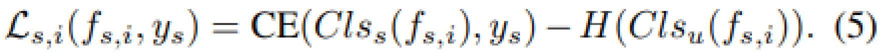


$$\mathcal{L}_{\text{indp exp},i} = \sum_{i'=1, i' \neq i}^{k} \text{HSIC}(f_i, f_{i'}). \qquad (4)$$

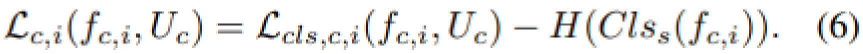
> Hilbert-Schmidt Independence Criterion (HSIC):
>
> 
>
> Formally, $\text{HSIC}_1^{k,l}(Z_c, Z_s)$ is defined as:
>
> $$\text{HSIC}_1^{k,l}(Z_c, Z_s) = \frac{1}{m(m-3)}\Bigg[\text{tr}(\widetilde{Z}_c \widetilde{Z}_s^T) + \frac{\mathbf{1}^T \widetilde{Z}_c \mathbf{1}\mathbf{1}^T \widetilde{Z}_s^T \mathbf{1}}{(m-1)(m-2)} - \frac{2}{m-2}\mathbf{1}^T \widetilde{Z}_c \widetilde{Z}_s^T \mathbf{1}\Bigg] \qquad (\text{B.2})$$
>
> where (i, j)-th element of a kernel matrix $\widetilde{Z}_c$ is defined as, $\widetilde{Z}_c(i,j) = (1 - \delta_{ij})\, k(f_c^i, f_c^j)$, and the i-th feature in the mini-batch f_c^i, is assumed to be sampled from the Z_c, *i.e.*, $\{f_c^i\} \sim Z_c$. We similarly define $\widetilde{Z}_s(i,j) = (1 - \delta_{ij})\, l(f_s^i, f_s^j)$.

Content and style disentanglement

- To achieve perfect content and style disentanglement, the style (or content) features should **include the style** (or content) domain information but **exclude the content** (or style) domain information. We employ two objective functions for this: **content-style adversarial loss** and **independent loss**.
- The content-style adversarial loss, motivated by the domain adversarial network [11], enforces the extracted features for style (or content) is useless to classify content.

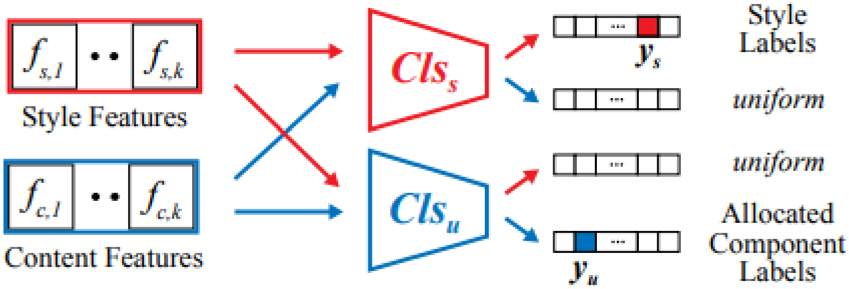


$$\mathcal{L}_{s,i}(f_{s,i}, y_s) = \text{CE}(Cls_s(f_{s,i}), y_s) - H(Cls_u(f_{s,i})). \quad (5)$$



$$\mathcal{L}_{c,i}(f_{c,i}, U_c) = \mathcal{L}_{cls,c,i}(f_{c,i}, U_c) - H(Cls_s(f_{c,i})). \quad (6)$$

> disentanglement framework:



> Figure 5. **Feature classifiers.** Two feature classifiers, Cls_s and Cls_u are used during the training. Cls_s classifies the style features to their style label y_s while Cls_u predicts the uniform probability from them. Similarly, Cls_u classifies the content features to their allocated component labels y_u while Cls_s is fooled by them. The details are described in § 3.2 and § 3.3.

> auxiliary component classification loss is defined as follow:
>
> $$\mathcal{L}_{cls,c,i}(f_{c,i}, U_c) = \sum_{j \in U_c} w_{ij} \text{CE}(Cls_u(f_{c,i}), j). \quad (3)$$
>
> , where variables w_{ij} is getting from Hungarian algorithm.

- We also employ the independence loss between content and style local features, $f_{c,i}$ and $f_{s,i}$ for the disentanglement of content and style representations.

$$\mathcal{L}_{\text{indp},i} = \text{HSIC}(f_{s,i}, f_{c,i}). \quad (7)$$

Training

We use the **hinge generative adversarial loss** L_{adv} [52], **feature matching loss** L_{fm}, and **pixel-level reconstruction loss** L_{recon} by following the previous high fidelity GANs, e.g., BigGAN [4], and state-of-the-art font generation methods, e.g., DM-Font [6] or LF-Font [37].

Now we describe our full objective function. The entire model is trained in an end-to-end manner with the weighted sum of all losses, including (4), (5), (6), and (7).

$$\begin{aligned} \mathcal{L}_D &= \mathcal{L}_{adv}^D, \\ \mathcal{L}_G &= \mathcal{L}_{adv}^G + \lambda_{recon}\mathcal{L}_{recon} + \mathcal{L}_{fm} \\ \mathcal{L}_{exp} &= \sum_{i=1}^{k} [\mathcal{L}_{s,i} + \mathcal{L}_{c,i} + \mathcal{L}_{\text{indp},i} + \mathcal{L}_{\text{indp exp},i}] \end{aligned} \quad (8)$$

As conventional GAN training, we alternatively update L_D, L_G, and L_{exp}.

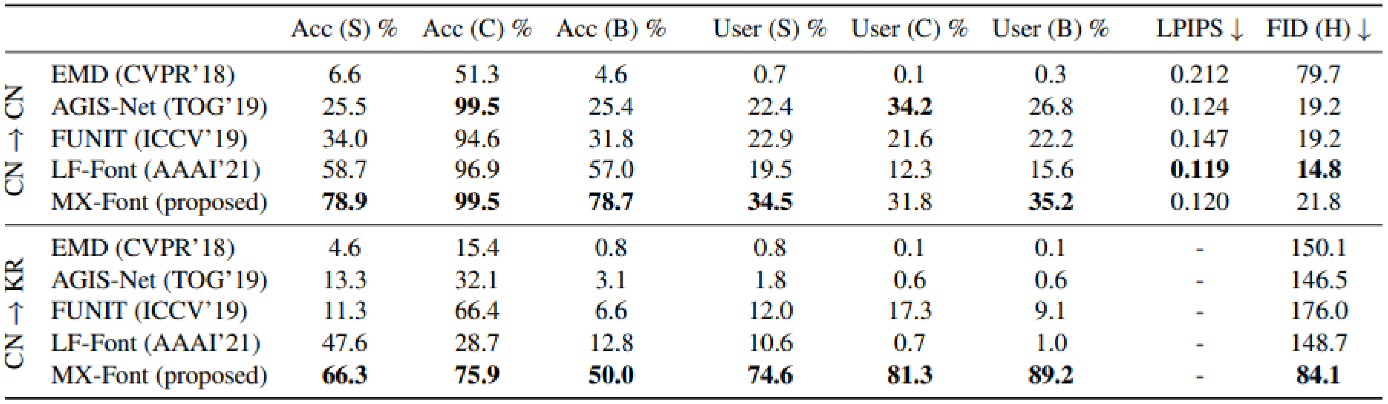
Experiments

Quantitative evaluation.

Following previous works [6, 37], we train evaluation classifiers that **classifies character labels** (content-aware) and **font labels** (style-aware). Note that these classifiers are only used for evaluation, and trained separately to the FFG models.

We conduct a **user study** for quantifying the subjective quality. The participants are asked to pick the three best results, considering the style, the content, and the most preferred considering both the style and the content.

We also report **Learned Perceptual Image Patch Similarity (LPIPS)** [53] scores to measure the dissimilarity between the generated images and their corresponding ground truth images, thus it is only reported for Chinese FFG task.



		Acc (S) %	Acc (C) %	Acc (B) %	User (S) %	User (C) %	User (B) %	LPIPS ↓	FID (H) ↓
CN → CN	EMD (CVPR'18)	6.6	51.3	4.6	0.7	0.1	0.3	0.212	79.7
	AGIS-Net (TOG'19)	25.5	**99.5**	25.4	22.4	**34.2**	26.8	0.124	19.2
	FUNIT (ICCV'19)	34.0	94.6	31.8	22.9	21.6	22.2	0.147	19.2
	LF-Font (AAAI'21)	58.7	96.9	57.0	19.5	12.3	15.6	**0.119**	**14.8**
	MX-Font (proposed)	**78.9**	**99.5**	**78.7**	**34.5**	31.8	**35.2**	0.120	21.8
CN → KR	EMD (CVPR'18)	4.6	15.4	0.8	0.8	0.1	0.1	-	150.1
	AGIS-Net (TOG'19)	13.3	32.1	3.1	1.8	0.6	0.6	-	146.5
	FUNIT (ICCV'19)	11.3	66.4	6.6	12.0	17.3	9.1	-	176.0
	LF-Font (AAAI'21)	47.6	28.7	12.8	10.6	0.7	1.0	-	148.7
	MX-Font (proposed)	**66.3**	**75.9**	**50.0**	**74.6**	**81.3**	**89.2**	-	**84.1**

Table 1. **Performance comparison on few-shot font generation scenario.** The performances of five few-shot font generation methods with four reference images are compared. We report accuracy measured by style-aware (Acc (S)) and content-aware (Acc (C)) classifiers and accuracy considering both the style and content labels (Acc (B)). The summarized results of the user study are also reported. The User preference on considering style (User (S)), content (User (C)), both of them (User (B)) are shown. LPIPS shows a perceptual dissimilarity between the ground truth and the generated glyphs. The harmonic mean (H) of style-aware and content-aware FID is also reported. Note that the FIDs are computed differently in two FFG scenarios. All numbers are average of 50 runs with different reference glyphs.

Qualitative evaluation

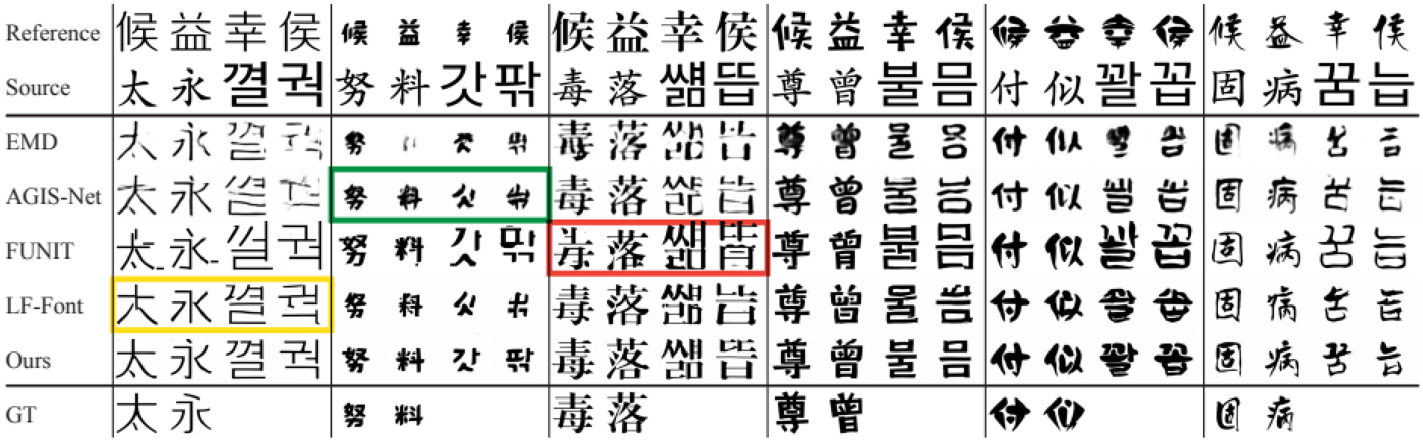


Figure 6. **Generated Samples.** The generated images by five different models are shown. We also provide the reference and the source images used for the generation in the top two rows. The available ground truth images (GT) are shown in the bottom row. We highlight the samples that reveal the drawback of each model with colored boxes; green for AGIS-Net, red for FUNIT, and yellow for LF-Font.

Analyses

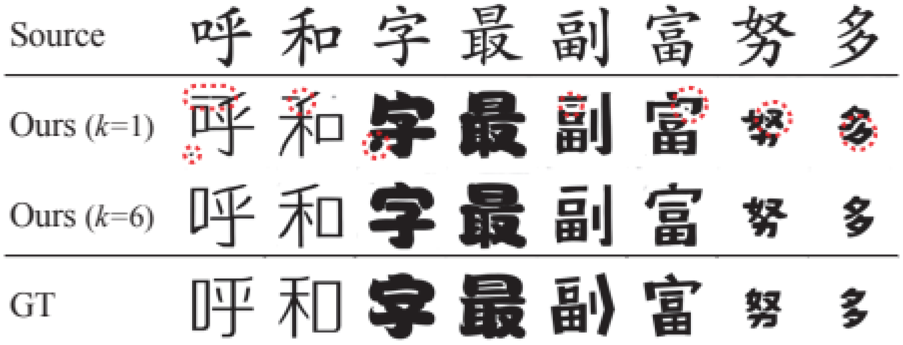
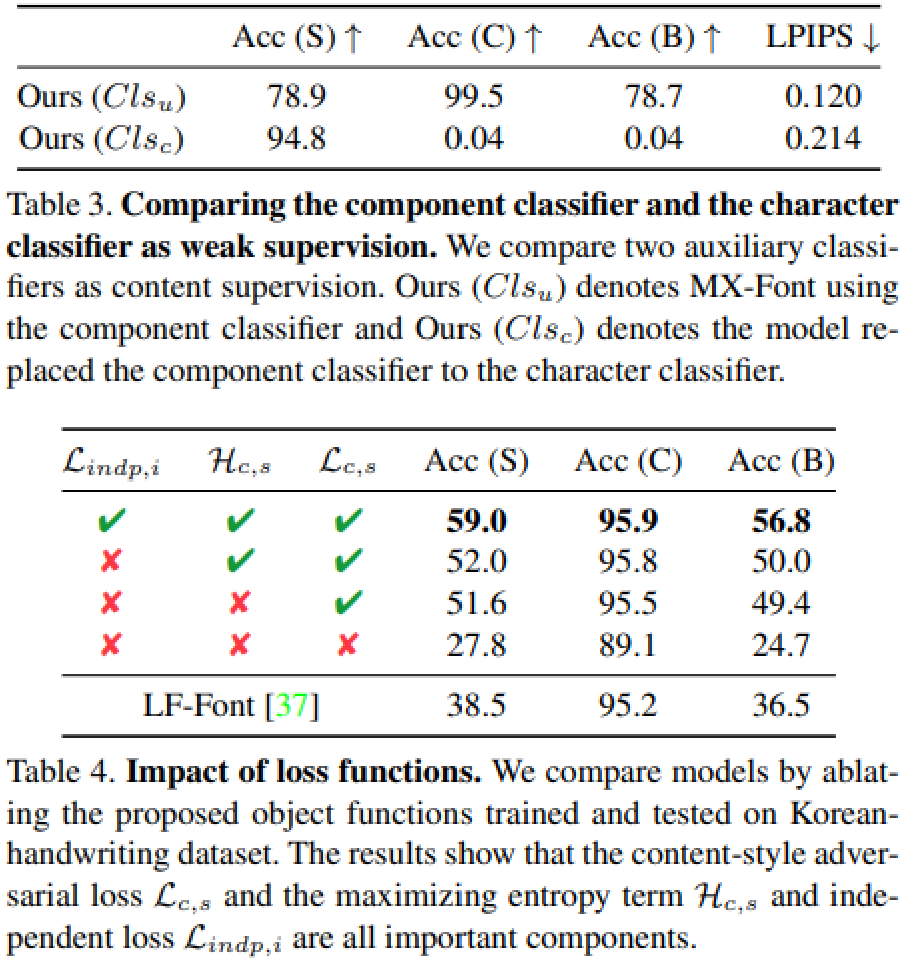


Figure 8. **Generated samples of the models having different number of heads.** The samples generated with four reference glyphs by the single-headed model and multi-headed model are shown. We highlight the defects in red dotted circles that appeared in the images generated by the single-expert model. k denotes the number of experts.

	Acc (S) ↑	Acc (C) ↑	Acc (B) ↑	LPIPS ↓
Ours ($k = 1$)	72.2	98.7	71.4	0.133
Ours ($k = 6$)	78.9	99.5	78.7	0.120

Table 2. **Impact of the number of experts k.** Single-expert model ($k = 1$) and multiple-experts model ($k = 6$, proposed) are compared on in-domain Chinese transfer benchmark.



	Acc (S) ↑	Acc (C) ↑	Acc (B) ↑	LPIPS ↓
Ours (Cls_u)	78.9	99.5	78.7	0.120
Ours (Cls_c)	94.8	0.04	0.04	0.214

Table 3. **Comparing the component classifier and the character classifier as weak supervision.** We compare two auxiliary classifiers as content supervision. Ours (Cls_u) denotes MX-Font using the component classifier and Ours (Cls_c) denotes the model replaced the component classifier to the character classifier.

$\mathcal{L}_{indp,i}$	$\mathcal{H}_{c,s}$	$\mathcal{L}_{c,s}$	Acc (S)	Acc (C)	Acc (B)
✓	✓	✓	**59.0**	**95.9**	**56.8**
✗	✓	✓	52.0	95.8	50.0
✗	✗	✓	51.6	95.5	49.4
✗	✗	✗	27.8	89.1	24.7
LF-Font [37]			38.5	95.2	36.5

Table 4. **Impact of loss functions.** We compare models by ablating the proposed object functions trained and tested on Korean-handwriting dataset. The results show that the content-style adversarial loss $\mathcal{L}_{c,s}$ and the maximizing entropy term $\mathcal{H}_{c,s}$ and independent loss $\mathcal{L}_{indp,i}$ are all important components.

Appendix

In the **cross-lingual** FFG, MX-Font can produce promising results in that they are all readable. Meanwhile, all other competitors provide inconsistent results, which are often impossible to understand. These results show a similar conclusion as our main paper.

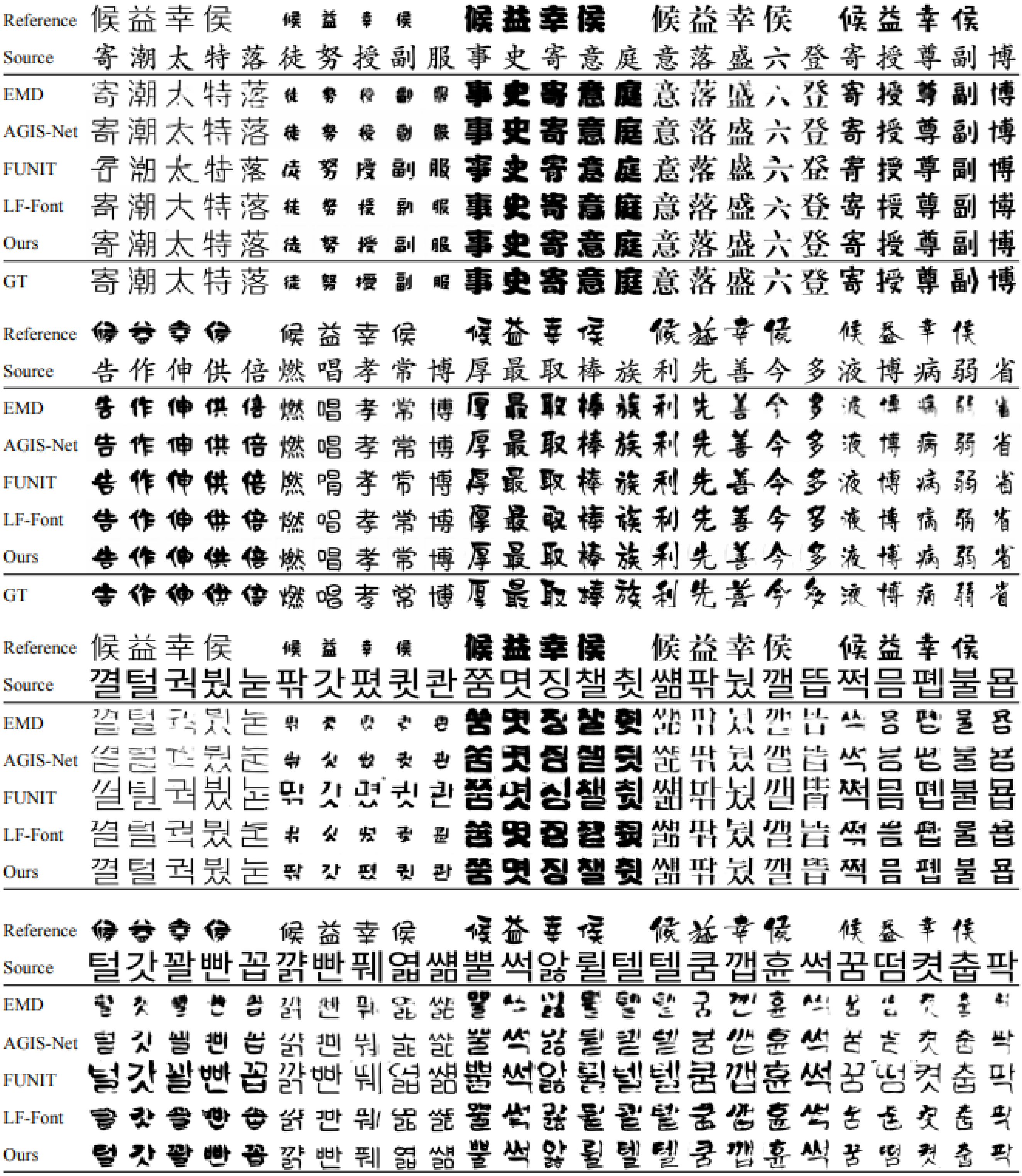


Figure A.2. **Generation samples.** We provide more generated glyphs with four reference glyphs.