Beyond Patches: Global-aware Autoregressive Model for Multimodal Few-Shot Font Generation

Supplementary Material

A. Overview

This supplementary material provides additional details on dataset construction, implementation, and extended experiments supporting the main paper. The sections are organized as follows:

- **Sec. B**: Model configurations of G-Tok, the autoregressive generator, and the multimodal style encoder.
- **Sec. C**: Data curation and partitioning, covering both training and evaluation protocols.
- **Sec. D**: Extended quantitative results, including scaling analyses of our autoregressive generator.
- **Sec. E**: Detailed and visualized ablations of key design choices in GAR-Font.
- **Sec. F**: Qualitative results for visual-only and multimodal FFG, cross-language, and higher-resolution generation.
- **Sec. G**: Analysis of GAR-Font failure cases in dense-stroke and complex font styles.

B. Model Configuration

Tab. S1 details the architectural specifications of GAR-Font. The framework relies on three core components: (1) The **Global-aware Tokenizer (G-Tok)** , which employs a hybrid CNN-ViT to discretize glyphs into a compact codebook; (2) The **Autoregressive Generator**, which serves as the synthesis backbone, using a Transformer decoder to predict tokens conditioned on aggregated content and style features; and (3) The **Multimodal Style Encoder**, which utilizes a lightweight adapter to align textual embeddings with visual style features for text-driven control.

Key Components	Params (M)
G-Tok	79.59
CNN Encoder	28.56
ViT Encoder (layers = 6)	4.73
Codebook (size = 2048, dim = 8)	0.02
ViT Decoder (layers = 6)	4.73
CNN Decoder	41.42
AR-Generator	346.23
Content Encoder	28.56
Visual Style Encoder	2.78
Content-style Aggregator (layers = 3)	0.79
Transformer Decoder (layers = 24)	314.10
Multimodal Style Encoder	8.04
Projection	0.52
Visual Style Encoder	2.78
Language-Style Adapter (layers = 6)	4.74

Table S1. Key GAR-Font components and parameter counts.

C. Data Curation

C.1. Data Collection and Statistics

We construct a comprehensive font dataset derived from the official GB2312 character set. As illustrated in Fig. S1, the whole training and test dataset is structured as a matrix spanned by two orthogonal axes: *Font Style* (vertical axis) and *Character Category* (horizontal axis). The collected data comprises 3,040 fonts and 6,763 characters.

For training data, along the Character Axis, we split 6,251 training characters (left column) and left 512 characters unseen (right column). The unseen characters are reserved strictly for testing to evaluate the model's capability to generate novel glyph structures. Similarly, the font library is divided into 3,000 training fonts (top rows) and 40 unseen test fonts (bottom rows).

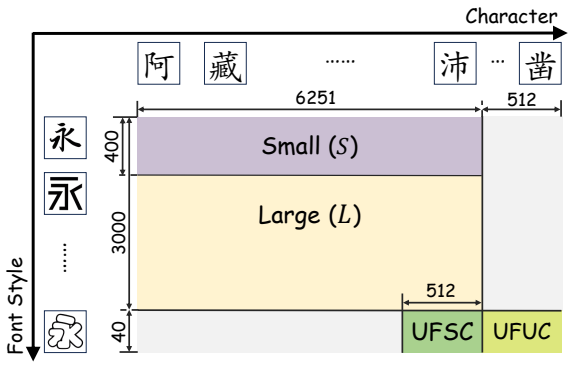


Figure S1. **Visual illustration of the dataset partition.** The data is organized along font and character axes. Pre-training utilizes the purple and yellow regions (S and L). Evaluation is conducted on the bottom green regions (*UFSC* and *UFUC*), strictly isolating unseen styles and characters.

C.2. Pretraining Data

The pretraining phase utilizes the data located in the upper-left quadrant of Fig. S1, defined by the intersection of training fonts and training characters. Within this quadrant, we define two configurations to investigate scaling behaviors:

- Large (L): The full training block consisting of all 3,000 training fonts paired with the 6,251 training characters (represented by the blue region).
- Small (S): A subset consisting of the first 400 training fonts paired with the same 6,251 characters (represented by the reddish overlay).

Training on S versus L allows us to assess the model's data efficiency and performance scaling with respect to the diversity of source styles.

C.3. Textual Prompt Collection

To support multimodal few-shot font generation (FFG), we construct a consistent textual prompt set that captures font-level stylistic attributes. Since human-authored font design descriptions are not available in existing datasets, we automatically generate textual prompts to approximate human design intent. For each font, we randomly sample 40 glyph images and jointly input them into Qwen2.5-VL. The model is instructed to produce a single, unified description summarizing only the visual properties that remain consistent across the full glyph set—such as stroke weight, curvature, structural proportions, spatial rhythm, edge texture, and overall tonal characteristics. This process yields a controlled and stylistically coherent textual representation for each font.

The exact prompt used for textual description extraction is provided below:

Textual Style Description Collection Prompt

You are an experienced typographic style analyst. You are given a set of glyph images belonging to the same font. Your task is to synthesize a unified stylistic description that captures only the consistent, font-level visual attributes shared across the full glyph set.
Your output must adhere to the following specifications:
1. Required Format
Provide a single paragraph that:
- begins with the phrase "A font that …",
- contains approximately 45–50 words,
- includes only stylistic properties observable across all glyphs,
- avoids speculative or uncertain expressions.
2. Allowed Stylistic Dimensions
Constrain your analysis to the following attributes:
- stroke weight (light, medium, bold, uniform, contrasting),
- curvature (straight, angular, rounded, flowing, sharp),
- structural proportions (compact, tall, wide, balanced),
- spacing and rhythm (tight, loose, even, irregular),
- edge rendering (smooth, sharp, rough, brush-like),
- overall tone or mood (elegant, modern, classical, playful, gentle, formal).
3. Constraints
All statements must be visually grounded in the provided glyph set. Do not reference features specific to individual characters. The description must reflect global stylistic coherence and maintain typographic precision.

C.4. Post-Refinement Data

To further adapt the model to novel styles and enhance structural consistency, we employ specific data subsets:
Novel Font Adaptation (NFA). NFA adapts the pre-trained model to the style of the 40 unseen test fonts. For each test font, we sample 8 characters (NFA-8) from the 6,251 training character set to serve as style references. This process operates within the vertical column of the training characters but focuses on the unseen font rows.
Structural Enhancement (SE). SE aims to consolidate global glyph consistency. It utilizes the entire 6,763 characters (spanning both training and unseen characters) but restricts the style to a manageable subset of 400 fonts (sampled from S). This ensures the model sees a complete range of structural geometries during the refinement phase without the computational cost of the full font library.

Table S2. Quantitative evaluation on VQ-Font vs. VQ-Font (G). Replacing the original VQ-VAE with G-Tok halves the FID score and boosts content accuracy by nearly 9%.

Unseen Fonts Seen Characters (UFSC)						
Method	RMSE↓	SSIM↑	LPIPS↓	FID↓	Acc(C)↑	Acc(S)↑
VQ_Font	0.2727	0.5642	0.1830	35.2472	0.8763	0.0016
VQ_Font (G)	**0.2725**	**0.5644**	**0.1731**	**17.3296**	**0.9646**	**0.0022**
Unseen Fonts Unseen Characters (UFUC)						
Method	RMSE↓	SSIM↑	LPIPS↓	FID↓	Acc(C)↑	Acc(S)↑
VQ_Font	0.2744	0.5616	0.1822	36.7914	0.8882	0.0016
VQ_Font (G)	**0.2732**	**0.5637**	**0.1731**	**17.9152**	**0.9653**	**0.0023**

Figure S2. Qualitative comparison on VQ-Font vs. VQ-Font (G). marks structural errors in the generated glyph. G-Tok improves structure preservation and produces more faithful font styles.

C.5. Evaluation Data

Evaluation is strictly conducted on the held-out bottom rows of the matrix (Fig. S1), ensuring no overlap with the pre-training data. We define two rigorous settings:

- *UFSC* (Unseen Fonts, Seen Characters): Represented by the light green region. This setting evaluates the model's ability to stylize known characters into novel font styles.
- *UFUC* (Unseen Fonts, Unseen Characters): Represented by the dark green region. This is the most challenging zero-shot setting, where the model must generate glyphs that are novel in both style and structure.

D. More Quantitative Experiments

D.1. Adaptation on G-Tok to Other FFG Methods

To verify the versatility of our G-Tok, we integrated it into VQ-Font by replacing its native VQ-VAE with our G-Tok while maintaining the original model architecture and configuration. The modified model, **VQ-Font (G)**, was trained on the Small dataset with G-Tok. As shown in Tab. S2, this simple replacement yields significant improvements across all metrics. Most notably, FID↓ decreases by nearly 50% (e.g., 35.25 → 17.33 on *UFSC*) and Content Accuracy↑ improves by approximately 9% ($\sim$ 87% →$\sim$ 96%). These substantial gains demonstrate that G-Tok's

Table S3. Quantitative results of NFA glyph number ablation for few-shot font adaptation. Increasing from NFA-8 to NFA-128 consistently improves style faithfulness and perceptual quality. Applying SE further enhances structural fidelity.

Method	Train	Unseen Fonts Seen Characters (UFSC)						Unseen Fonts Unseen Characters (UFUC)					
	Set	RMSE↓	SSIM↑	LPIPS↓	FID↓	Acc(C)↑	Acc(S)↑	RMSE↓	SSIM↑	LPIPS↓	FID↓	Acc(C)↑	Acc(S)↑
NFA-8	*S*	0.2979	0.5418	0.1177	6.6909	0.9195	0.1128	0.3002	0.5354	0.1195	6.4693	0.9191	0.1112
	L	0.2600	0.6158	0.0979	6.5634	0.9210	0.3313	0.2603	0.6160	0.0983	6.5842	0.8921	0.3518
NFA-8+SE	*S*	0.2909	0.5619	0.1111	8.4951	0.9817	0.1101	0.2935	0.5553	0.1129	8.3504	0.9804	0.1025
	L	0.2503	0.6411	0.0885	8.9851	0.9795	0.3518	0.2540	0.6356	0.0903	8.6602	0.9670	0.3735
NFA-32	*S*	0.2880	0.5598	0.1095	5.7662	0.8993	0.1786	0.2849	0.5679	0.1075	5.8886	0.8962	0.1978
	L	0.2561	0.6238	0.0949	6.1329	0.9093	0.3707	0.2570	0.6234	0.0957	6.1493	0.8758	0.3946
NFA-32+SE	*S*	0.2803	0.5821	0.1028	7.1792	0.9734	0.1754	0.2781	0.5878	0.1012	7.1134	0.9724	0.1970
	L	0.2455	0.6515	0.0854	8.2308	0.9775	0.4048	0.2460	0.6513	0.0862	7.8937	0.9581	0.4342
NFA-128	*S*	0.2712	0.5933	0.0992	5.4254	0.9236	0.3179	0.2836	0.5702	0.1079	5.6667	0.8817	0.3277
	L	0.2435	0.6507	0.0855	5.7570	0.9228	0.4457	0.2496	0.6397	0.0904	5.8078	0.8830	0.4625
NFA-128+SE	*S*	0.2671	0.6089	0.0948	6.9103	0.9718	0.2833	0.2788	0.5884	0.1022	7.1309	0.9242	0.2958
	L	0.2398	0.6627	0.0831	8.3751	0.9776	0.4154	0.2508	0.6377	0.0908	7.2034	0.9068	0.4183

Table S4. Quantitative results of GAR-Font (M2/M4) across different description sources and formats under UFSC. M2/M4 denote inference with 2/4 reference glyphs plus text. All results outperform the corresponding non-text baselines in Table 2.

Method	Variants	RMSE↓	SSIM↑	LPIPS↓	FID↓	Acc(C)↑	Acc(S)↑
SmolVLM2-2.2B (Fixed-Template)	M2	0.2768	0.5819	0.1111	7.9185	0.9143	0.1384
	M4	0.2732	0.5878	0.1088	7.4247	0.9206	0.1747
Qwen2.5-VL (Free-Form)	M2	0.2765	0.5808	0.1110	7.7404	0.9263	0.1329
	M4	0.2730	0.5890	0.1095	6.9813	0.9016	0.1702

hybrid CNN-ViT architecture captures far richer structural and stylistic semantics than standard VQ-VAEs, serving as a robust plug-and-play enhancement for quantization-based FFG methods. Fig. S2 illustrates that integrating G-Tok helps the model preserve coherent global structures for complex fonts and generate styles that better align with the target font, indicating a richer and more semantically stable global representation G-Tok than the original VQ-VAE.

D.2. Effect of the Number of NFA Glyphs

In Section 4.3, we adopt NFA-8 to maintain a strict few-shot adaptation setting. We further investigate the impact of using more adaptation glyphs by extending the setting to NFA-32 and NFA-128.

As visualized in Tab. S3, increasing the number of adaptation glyphs consistently improves the style modeling and perceptual quality. Style Accuracy (Acc(S)↑) rises notably from NFA-8 to NFA-128 on both UFSC and UFUC, indicating that additional glyphs provide richer stylistic cues for capturing font-specific characteristics. Notably, on the *Large* dataset, NFA-128 achieves the highest style accuracy (0.4457 on UFSC and 0.4625 on UFUC), substantially outperforming NFA-8. These results suggest that despite the NFA-8 setting adopted in the main paper already provides a strong and practical few-shot configuration, more NFA glyphs further refine the font transfer quality.

D.3. Robustness Across Textual Description Sources

In Section 4.4, multimodal experiments are conducted using the generated descriptions of Qwen2.5-VL with the fixed-form template in Section C.3. To validate the robustness of our approach, we further evaluate GAR-Font using two types of descriptions: a free-form prompt for Qwen2.5-VL and the fixed-template prompt for SmolVLM2-2.2B-Instruct [1], each generated using only **8** reference glyphs.

Tab. S4 shows the multimodal gains are consistent across description sources and formats. GAR-Font(M_2/M_4) outperforms corresponding non-text baselines, demonstrating that our vision-language adaptation is robust to variations in prompt style and source model.

D.4. Post-Refinement of Multimodal FFG

In Section 4.4, we demonstrated the efficacy of GAR-Font(M_2/M_4) on multimodal FFG, but limited to pretraining stage. To fully assess the potential of our lightweight vision-language adaptation, we extend the evaluation to the complete pipeline. We apply our NFA-128 and SE stages to both the vision-only baselines and our multimodal variants. All models are trained on the Large (*L*) dataset.

As visualized in Tab. S5, the inclusion of textual descriptions significantly enhances the effectiveness of the post-refinement stage. Unlike the pre-training phase where multimodal models showed a slight dip in style accuracy (Acc(S)↑), the fully refined GAR-Font(M_2) and GAR-Font(M_4) exhibit a substantial lead in Acc(S)↑ compared to their vision-only counterparts ($n_{\text{ref}} = 2$ and $n_{\text{ref}} = 4$). Notably, **GAR-Font(M_4) outperforms the 8-reference vision-only baseline ($n_{\text{ref}} = 8$)** across most key metrics, including RMSE↓, SSIM↑, LPIPS↓, and Style Accuracy↑ (0.4566 vs. 0.4154 on *UFSC*).

Table S5. Quantitative evaluation of Multimodal FFG with full post-refinement (NFA and SE) on Unseen Fonts. All models listed are post-trained with NFA-128 and SE stages on the Large dataset.

Method	Unseen Fonts Seen Characters (UFSC)						Unseen Fonts Unseen Characters (UFUC)					
	RMSE↓	SSIM↑	LPIPS↓	FID↓	Acc(C)↑	Acc(S)↑	RMSE↓	SSIM↑	LPIPS↓	FID↓	Acc(C)↑	Acc(S)↑
$n_{\text{ref}} = 2$	0.2501	0.6438	0.0888	10.1464	**0.9851**	0.3314	0.2701	0.5987	0.1069	9.4294	0.8720	0.3433
$n_{\text{ref}} = 4$	0.2437	0.6552	0.0848	9.5960	0.9839	0.3711	0.2692	0.6007	0.1049	9.0877	0.8828	0.3835
$n_{\text{ref}} = 8$	0.2398	0.6627	0.0831	**8.3751**	0.9776	0.4154	**0.2508**	**0.6377**	**0.0908**	**7.2034**	**0.9068**	0.4183
GAR-Font(M_2)	0.2361	0.6707	0.0799	8.8867	0.9817	0.4508	0.2527	0.6352	0.0909	8.1444	0.9029	0.4247
GAR-Font(M_4)	**0.2358**	**0.6712**	**0.0796**	8.8073	0.9800	**0.4566**	0.2524	0.6353	**0.0908**	8.0699	0.9023	**0.4391**

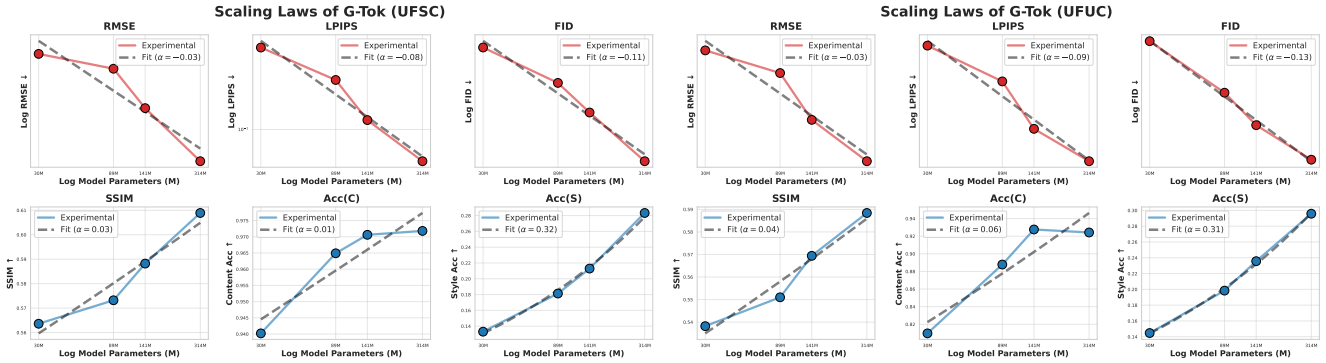


Figure S3. Scaling laws of the GAR-Font (I_8) generator with NFA-128 and SE refinement. The plots show performance metrics across model sizes (30M, 89M, 141M, 314M) on Unseen Fonts Seen Characters (*UFSC*) and Unseen Fonts Unseen Characters (*UFUC*). The dashed lines represent power-law fits, highlighting the predictable improvements in both perceptual quality and style generalization.

Table S6. Quantitative evaluation of G-Tok's architecture on Unseen Fonts. All models listed are pre-trained on the Small dataset.

Method	Unseen Fonts Seen Characters (UFSC)						Unseen Fonts Unseen Characters (UFUC)					
	RMSE↓	SSIM↑	LPIPS↓	FID↓	Acc(C)↑	Acc(S)↑	RMSE↓	SSIM↑	LPIPS↓	FID↓	Acc(C)↑	Acc(S)↑
CNN	0.3212	0.4836	0.1442	9.5071	0.9051	0.0235	0.3447	0.4350	0.1728	10.5239	0.6722	0.0221
CNN+Non-Causal ViT	0.3183	0.4919	0.1458	**7.9101**	0.9268	0.0402	0.3271	0.4745	0.1562	8.7504	0.8019	0.0436
CNN+Causal ViT	**0.3080**	**0.5052**	**0.1313**	7.9484	**0.9408**	**0.0802**	**0.3142**	**0.4932**	**0.1421**	**8.4841**	**0.8993**	**0.0796**

D.5. Scaling Laws of GAR-Font(I_8)

We evaluate the scalability of GAR-Font(I_8) with NFA-128 and SE on the small dataset by training models from 30M to 314M parameters and measuring performance across standard quantitative metrics. Following established scaling-law formulations, we model the relationship between model size N and loss metric L using a power law $L(N) \propto N^{-\alpha}$, and analyze trends in log–log space, where an ideal scaling law appears linear and the slope α reflects scaling efficiency.

As shown in Fig. S3, the enhanced GAR-Font models closely follow these power-law predictions, exhibiting smooth, monotonic improvements across all metrics. Loss-based metrics (FID↓, LPIPS↓, RMSE↓) scale linearly with negative slopes, with FID↓ showing a pronounced gain, indicating that larger models continue to yield substantial perceptual improvements without saturation. Accuracy metrics display complementary behavior: Content Accuracy (Acc(C)↑) saturates early due to task simplicity, whereas Style Accuracy (Acc(S)↑) benefits most from increased capacity. This steep scaling trend highlights that NFA and SE effectively exploit larger parameter budgets to capture and generalize complex stylistic attributes, underscoring the central role of scale in high-fidelity font generation.

E. Additional Ablative Studies

E.1. On G-Tok's hybrid Architecture

To further illustrate the robustness of our hybrid CNN–ViT tokenizer, we provide complete visualizations of the **Reconstruction Robustness** experiment, where glyphs are corrupted with localized Gaussian noise ($\sigma = 0.2$, affecting 20% area). The qualitative results in Fig. S4 demonstrate that G-Tok robustly recovers structural layout and stylistic traits even under severe perturbations, while non-hybrid alternatives fail to reconstruct consistent structure.

Figure S4. Reconstruction Robustness under localized Gaussian noise ($\sigma = 0.2$, 20% area). ☐ marks structural errors. G-Tok (CNN-ViT-6) preserves structure and style despite heavy corruption, while non-hybrid tokenizers exhibit unstable reconstructions.

E.2. On G-Tok's Global and Causal Modeling

We present full ablation results for the global and causal modeling components of G-Tok. Tab. S6 reports the complete quantitative comparison on the *UFSC/UFUC*. As discussed in Section 4.5.2, adding global self-attention (CNN + Non-causal ViT) significantly outperforms the CNN-only baseline, while the causal ViT further improves sequential modeling and yields the best overall performance.

Fig. S5 provides qualitative comparisons on (*UFSC*, Small) and (*UFUC*, Small). The AR Generator implemented with a CNN-only tokenizer often exhibits style mismatches and inconsistent strokes. Introducing ViT modules into the tokenizer enhances its ability to perceive and capture global stylistic context, leading to more coherent font generation. The AR variant with full G-Tok (CNN + Causal ViT) achieves the most robust performance, showing visible improvements in stylistic and structural fidelity.

E.3. On AR Generator's Soft-Decoding

We provide full visualizations to assess the impact of pixel-level supervision and the soft-decoding strategy. As shown in Fig. S6 under both (*UFSC*, Small) and (*UFUC*, Small), pixel-level supervision enhances structural accuracy, while soft decoding yields smoother, more continuous strokes and reduces broken segments and visual artifacts.

E.4. On Multimodal Style Encoder's Adaptation

We compare our decoupled multimodal training paradigm against joint training of the multimodal style encoder. While quantitative results are provided in Section 4.5.4, we present the full set of qualitative comparisons here.

Fig. S7 presents visual comparisons on (*UFSC/UFUC*, Large). The results reveal that GAR-Font(M_2/M_4), trained with the decoupled training scheme, generate glyphs whose font styles more closely align with the target compared to the jointly trained GAR-Font(VL_2/VL_4). They also demonstrate better character-structure accuracy. The decoupled training strategy enables the model to fully leverage the visual encoder's representational capacity, thereby preserving fine-grained style features and structural priors that may be harder to retain under joint optimization.

F. Visualization Results

F.1. Comparison on Few-shot Font Generation

We provide complete visualizations for the experiment in Section 4.3. In Fig. S8, we show the full qualitative comparisons of visual-only FFG models trained on Small and Large datasets, evaluated under both *UFSC* and *UFUC* protocols. These results indicate that methods such as LF–Font, VQ-Font, DG-Font, CF-Font and Diff-Font often fail to preserve structural fidelity in intricate fonts. IF-Font tends to produce incomplete characters, while Font-Diffuser generates with inaccurate stroke widths. In contrast, GAR-Font(I_8,+NFA-8+SE) achieves the best style fidelity while maintaining structural consistency, effectively capturing fine stroke details of the target fonts.

F.2. Efficient Vision-Language Adaptation

F.2.1. Pretrain

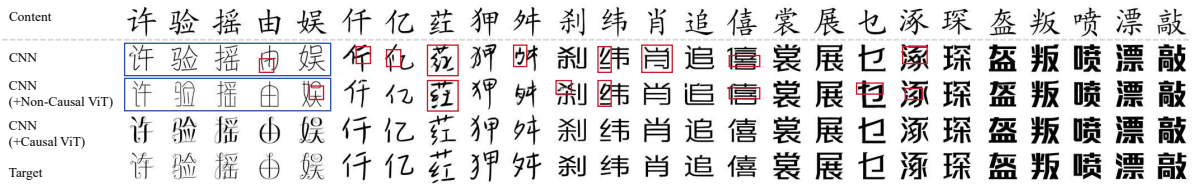
We provide full qualitative results complementing the experiment in Section 4.4. In Fig. S9, we show multimodal FFG comparisons under both *UFSC* and *UFUC* settings on the Large dataset, illustrating the improvements introduced by incorporating textual style descriptions in GAR-Font(M_2) and GAR-Font(M_4) compared with their vision-only counterparts. With textual style guidance, GAR-Font(M_2) and GAR-Font(M_4) better align with the target style, generating glyphs with strokes closely matching the target and improved structural fidelity.

F.2.2. Post-Refinement

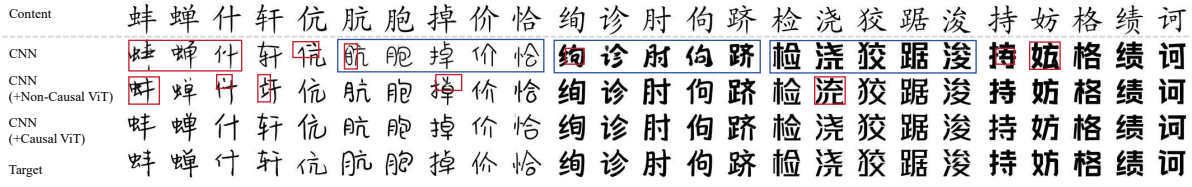
To further assess the potential of our efficient vision–language adaptation, we apply the complete post-refinement pipeline (NFA-128 and SE) to GAR-Font(M_2) and GAR-Font(M_4). Fig. S10 presents qualitative results under *UFSC* and *UFUC* on the Large dataset. Applying NFA and SE post-refinement significantly improves both structural and style fidelity for all models. Textual guidance further enables GAR-Font(M_2) and GAR-Font(M_4) to more accurately capture the target style, yielding glyphs with improved style fidelity.

F.3. Effect of Post-Refinement

To further analyze the effect of the post-refinement, we provide visual comparisons. As shown in Fig. S11, the pretrained GAR-Font(I_8) already produces characters with generally correct structures and styles, though minor font inconsistency exists. Applying NFA significantly improves style fidelity but may introduce slight distortions in fine strokes. The SE stage preserves style fidelity while further enhancing visual clarity and the accuracy of stroke details especially in complex fonts.



(a) *UFSC*, Small dataset



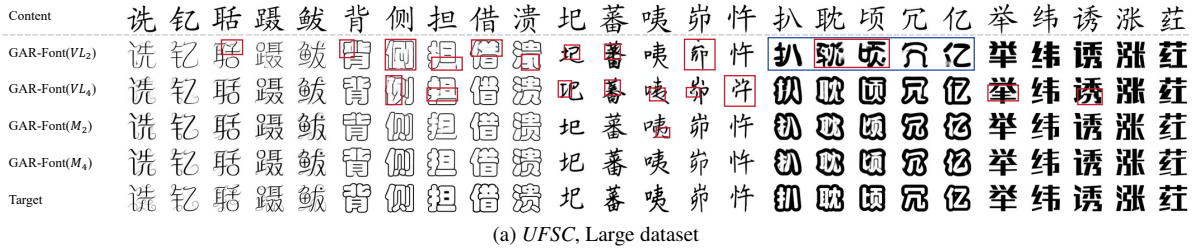
(b) *UFUC*, Small dataset

Figure S5. Qualitative results on G-Tok's Global and Causal Modeling under *UFSC* and *UFUC* protocols (Small dataset). ☐ / ☐ indicate structural errors and style mismatches.

(a) *UFSC*, Small dataset

(b) *UFUC*, Small dataset

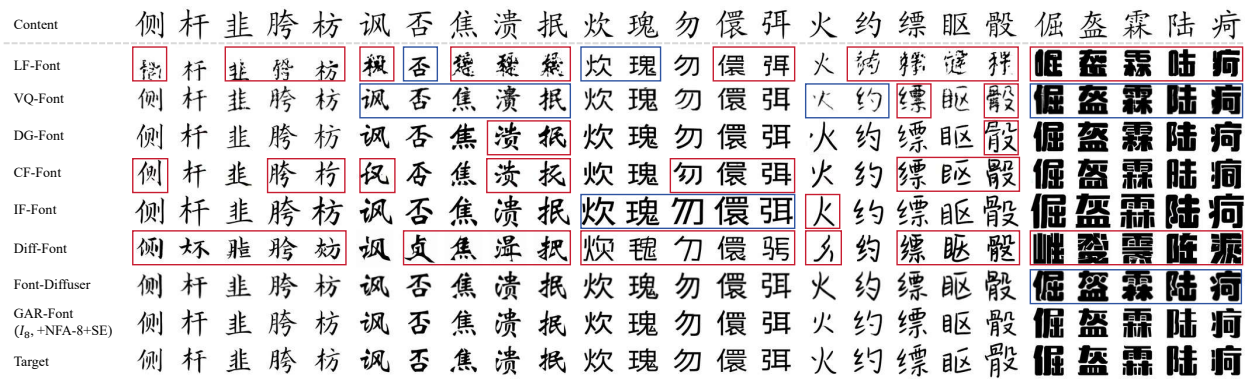
Figure S6. Qualitative results on AR Generator's Soft-decoding under *UFSC* and *UFUC* protocols (Small dataset). ☐ / ☐ indicate structural errors and style mismatches.



(a) *UFSC*, Large dataset

(b) *UFUC*, Large dataset

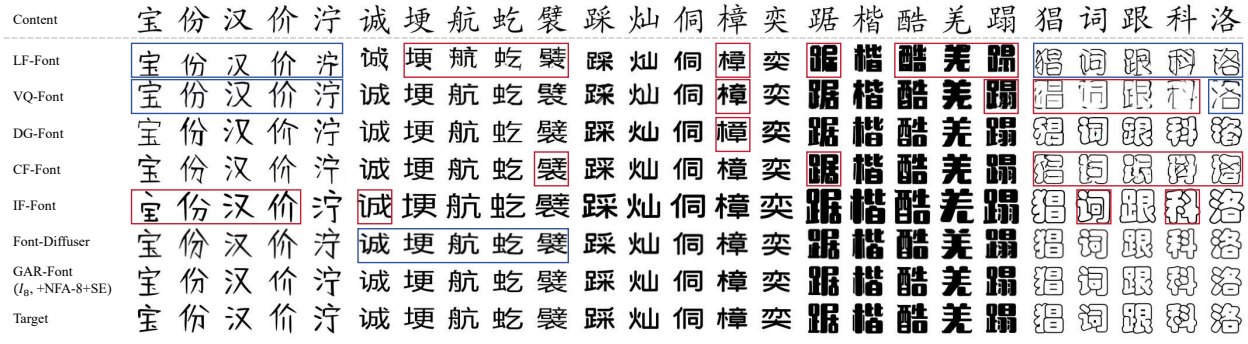
Figure S7. Qualitative results on Multimodal Style Encoder's Adaptation under *UFSC* and *UFUC* protocols (Large dataset). ☐ / ☐ indicate structural errors and style mismatches.



(a) *UFSC*, Small dataset

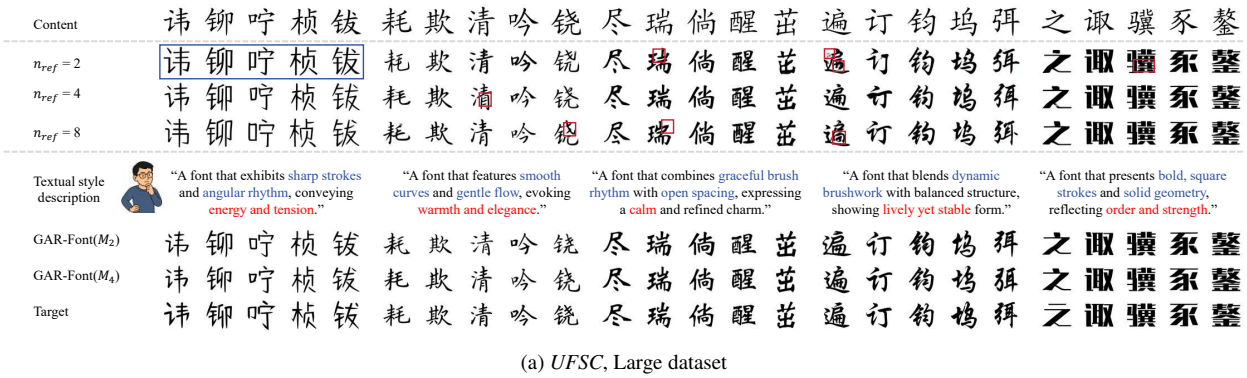
(b) *UFUC*, Small dataset

(c) *UFSC*, Large dataset

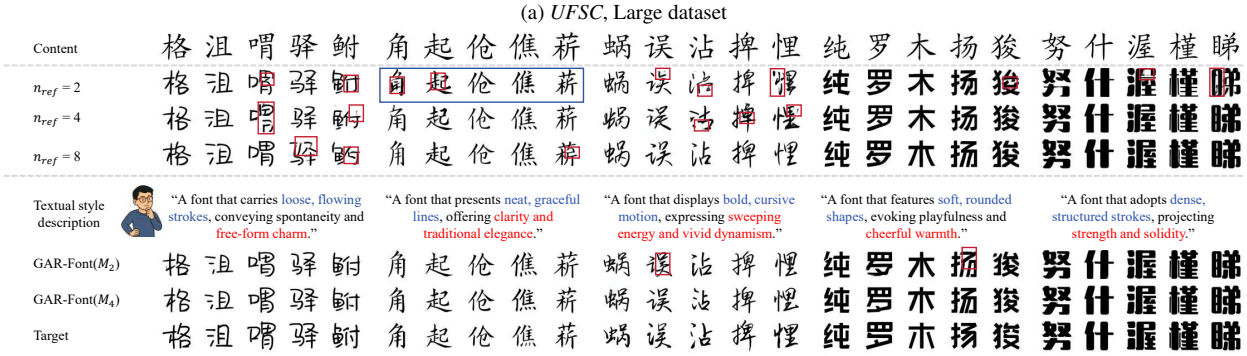


(d) *UFUC*, Large dataset

Figure S8. Qualitative results on vision-only FFG across *UFSC/UFUC* protocols and Small/Large datasets. / indicate structural errors and style mismatches.

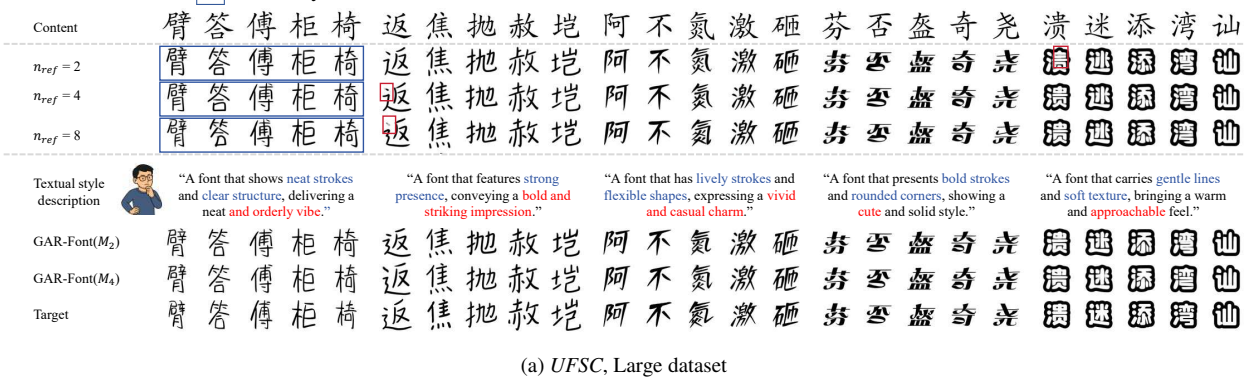


(a) *UFSC*, Large dataset



(b) *UFUC*, Large dataset

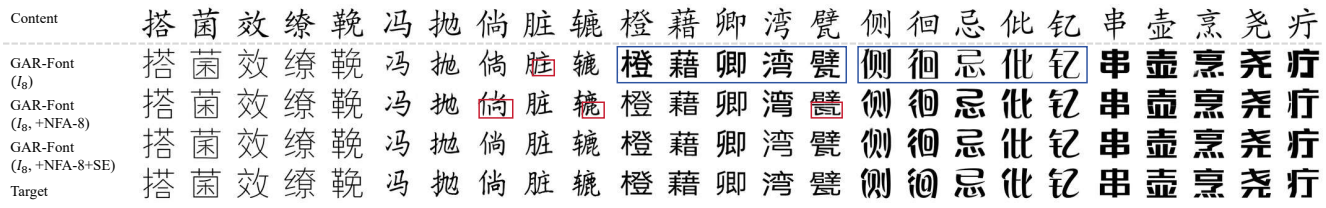
Figure S9. Qualitative results of Pre-Train multimodal FFG under *UFSC* and *UFUC* protocols (Large dataset). ☐ denotes local slight structural mistakes, and ☐ marks stylistic drift.



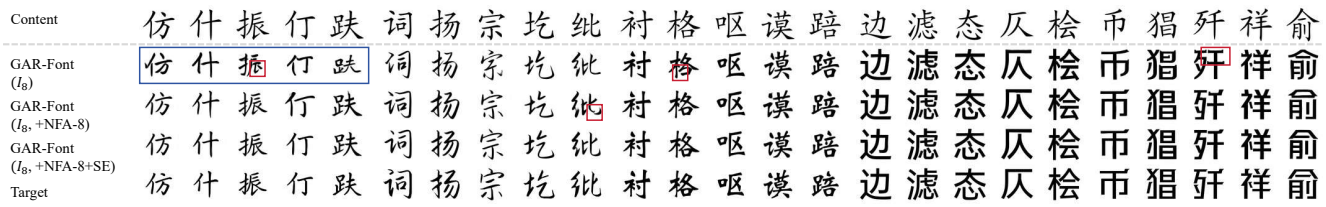
(a) *UFSC*, Large dataset

(b) *UFUC*, Large dataset

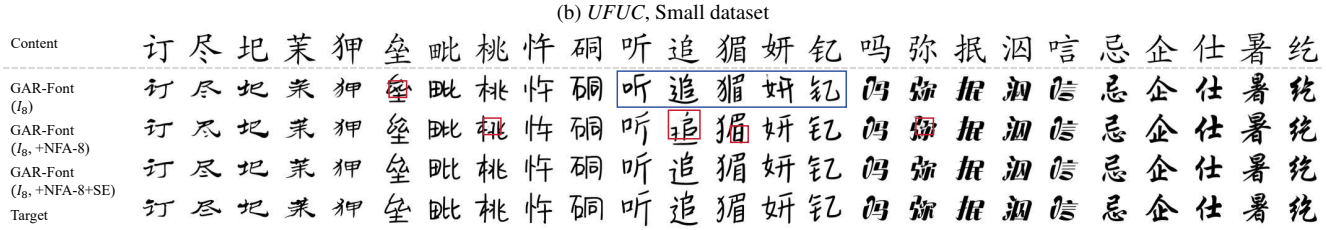
Figure S10. Qualitative results of Post-Refine multimodal FFG under *UFSC* and *UFUC* protocols (Large dataset). ☐ denotes local slight structural mistakes, and ☐ marks stylistic drift.



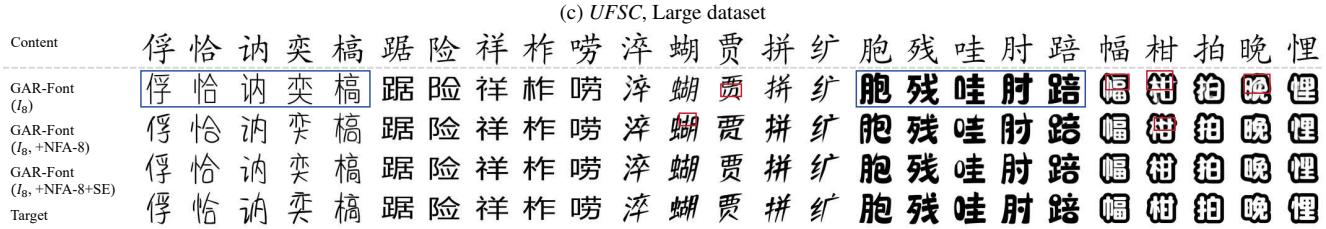
(a) *UFSC*, Small dataset



(b) *UFUC*, Small dataset



(c) *UFSC*, Large dataset



(d) *UFUC*, Large dataset

Figure S11. Qualitative results on Post-refinement across *UFSC/UFUC* protocols and Small/Large datasets. ☐ denotes local slight structural mistakes, and ☐ marks stylistic drift.

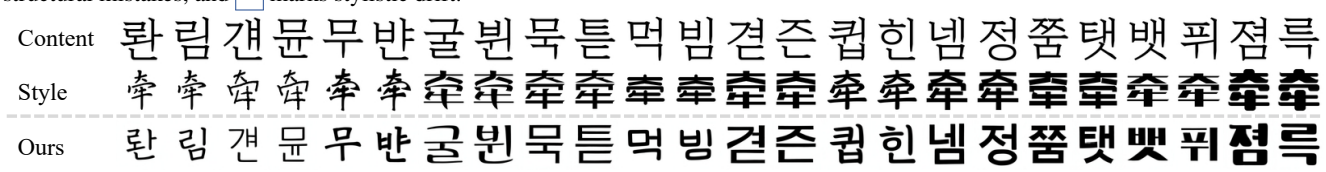


Figure S12. Qualitative results of cross-language font synthesis using GAR-Font(I_8) trained on the Large dataset, demonstrating the strong generalization ability.

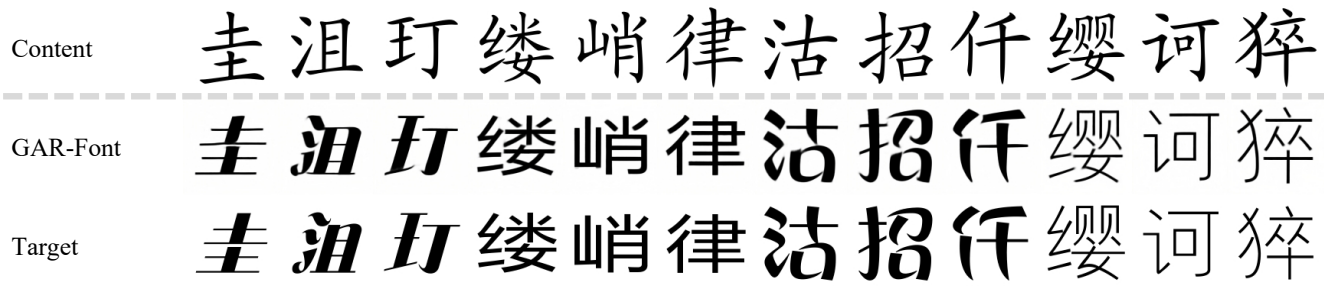


Figure S13. Qualitative results of 128 × 128 font generation using GAR-Font(I_8, +NFA-8+SE) trained on the Large dataset, showing the scalability of GAR-Font to higher resolutions while preserving style faithfulness and structural fidelity.

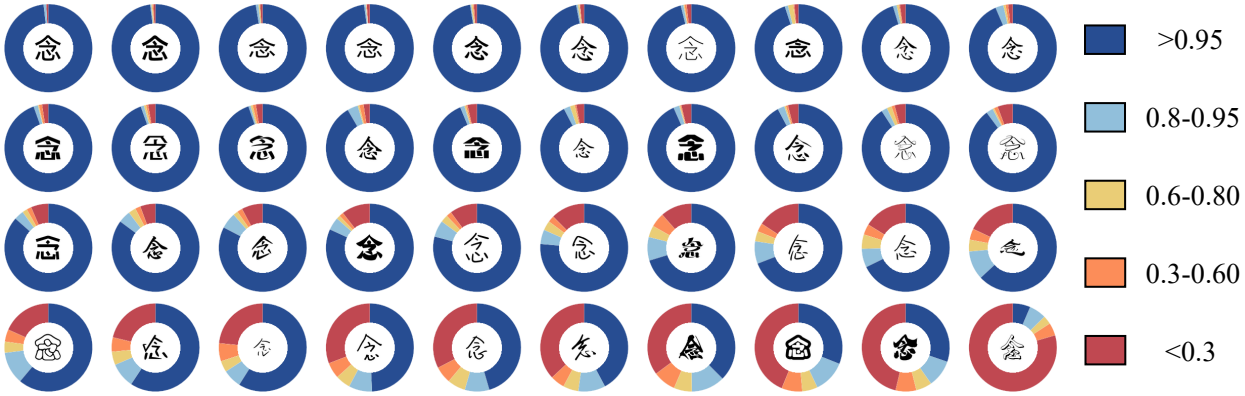


Figure S14. Content confidence distribution of GAR-Font generated characters across different font styles. Each pie chart corresponds to a specific font, indicated by the central character. The color segments represent the proportion of samples falling into different content confidence ranges, highlighting that more complex styles tend to have lower content confidence.

Figure S15. Failure cases. ☐ highlights regions with dense details where GAR-Font tends to produce distorted strokes.

F.4. Cross-Language Font Synthesis

To evaluate the generalizability of GAR-Font, we conduct a cross-language experiment in which the model synthesizes Korean characters using styles learned from Chinese fonts. As shown in Fig. S12, GAR-Font accurately generates Korean characters while preserving the reference font style, demonstrating the effective generalization of our method.

F.5. High-Resolution Font Generation

To demonstrate the scalability of GAR-Font beyond the 64×64 resolution adopted in our main experiments, we modify the CNN encoder within G-Tok to discretize a 128×128 glyph into 64 tokens, corresponding to a downsample ratio of 16. As illustrated in Fig. S13, GAR-Font maintains both style faithfulness and structural fidelity at this increased resolution, demonstrating its potential in high-resolution font generation tasks.

F.6. More GAR-Font Generation Examples

To illustrate the capabilities of GAR-Font, we generate the full GB2312 character set for five test fonts with GAR-Font(I_8, +NFA-8+SE, trained on Large dataset) and randomly select 1,280 samples per font. The generated glyphs are shown in Fig. S16-Fig. S20 , demonstrating the model's ability to produce large-scale character sets while faithfully preserving each font's distinctive stylistic features.

G. Failure Cases and Analysis

While GAR-Font generally performs well, distortions and blurring occasionally appear in dense-stroke regions of highly complex fonts (Fig. S15). To investigate this, we applied a content classifier to all *UFUC* samples generated by GAR-Font(I_8,+NFA-128+SE), using the softmax output as a measure of content confidence. The results reveal a clear trend: content confidence notably decreases as stylistic complexity increases (Fig. S14), suggesting the model sometimes sacrifices structural accuracy to better capture stylistic features.

We hypothesize that this structural degradation results from the error accumulation inherent in autoregressive modeling. Without explicit structural constraints, the model tends to drift when generating intricate stroke patterns. A promising direction for future work is to incorporate explicit structural priors, such as character skeletons or stroke sequences, to guide the generation process. This would help preserve structural fidelity in complex styles.

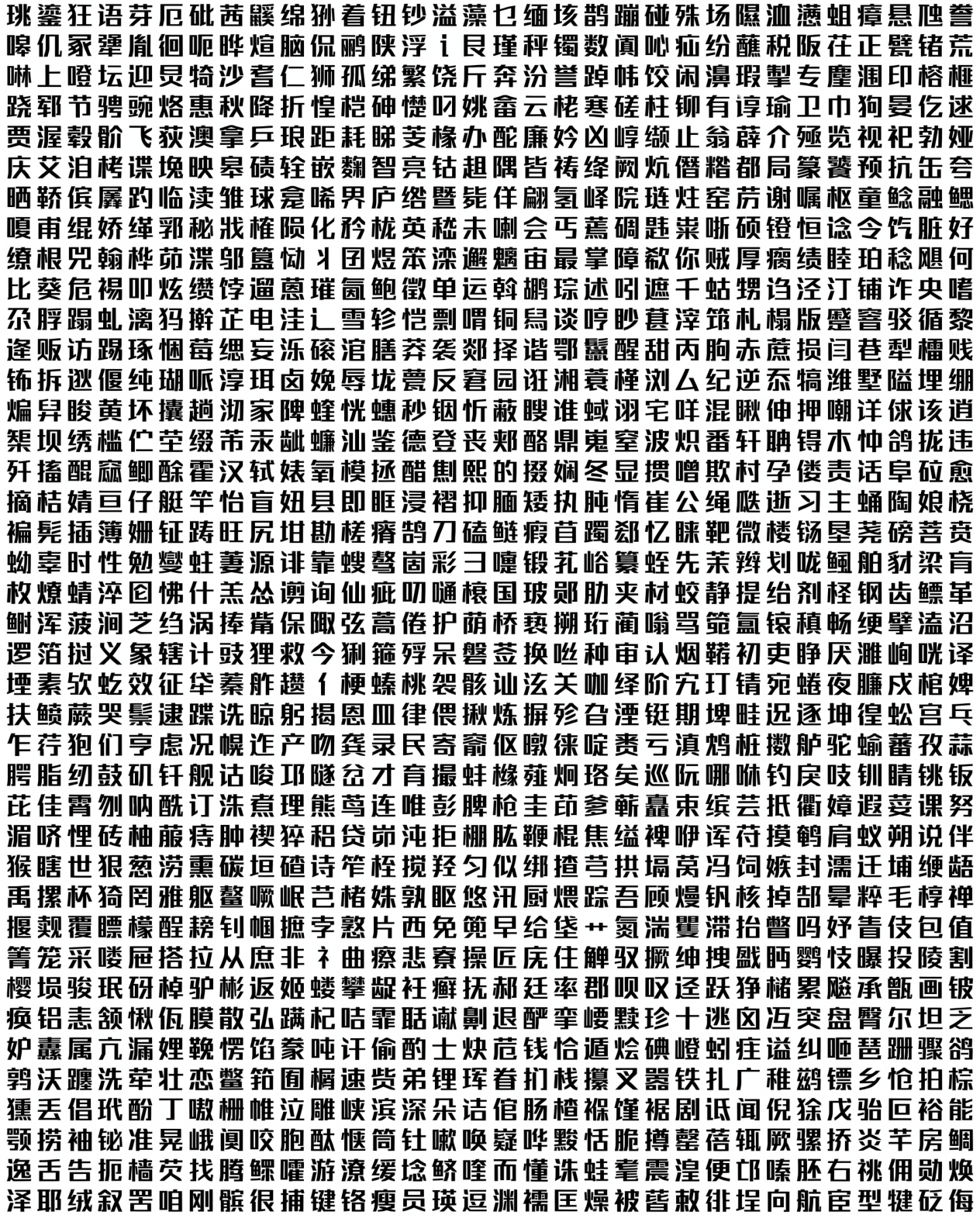


Figure S16. Generated glyphs from test fonts using GAR-Font(I_8, +NFA-8+SE, Large dataset).

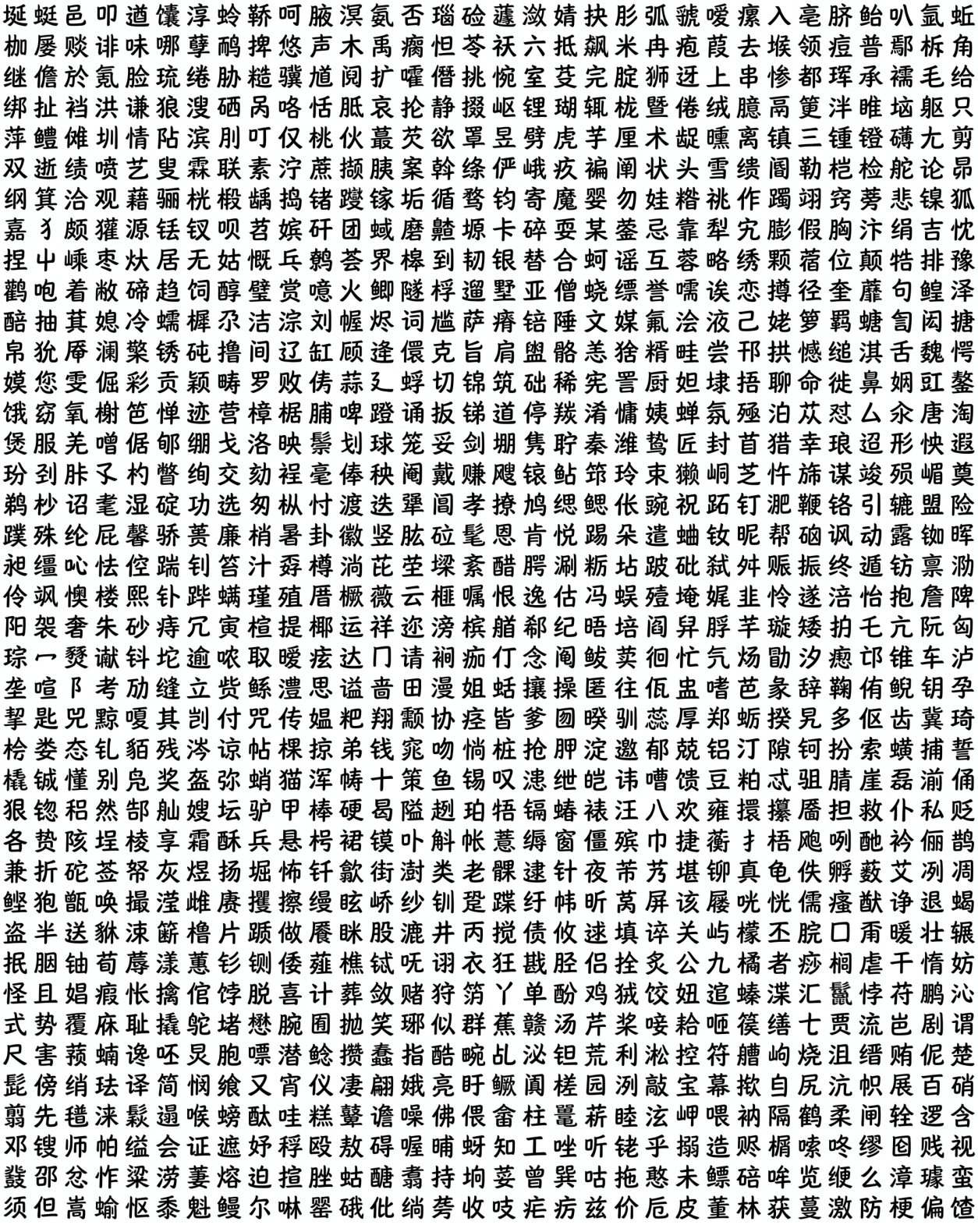


Figure S17. Generated glyphs from test fonts using GAR-Font(I_8, +NFA-8+SE, Large dataset).

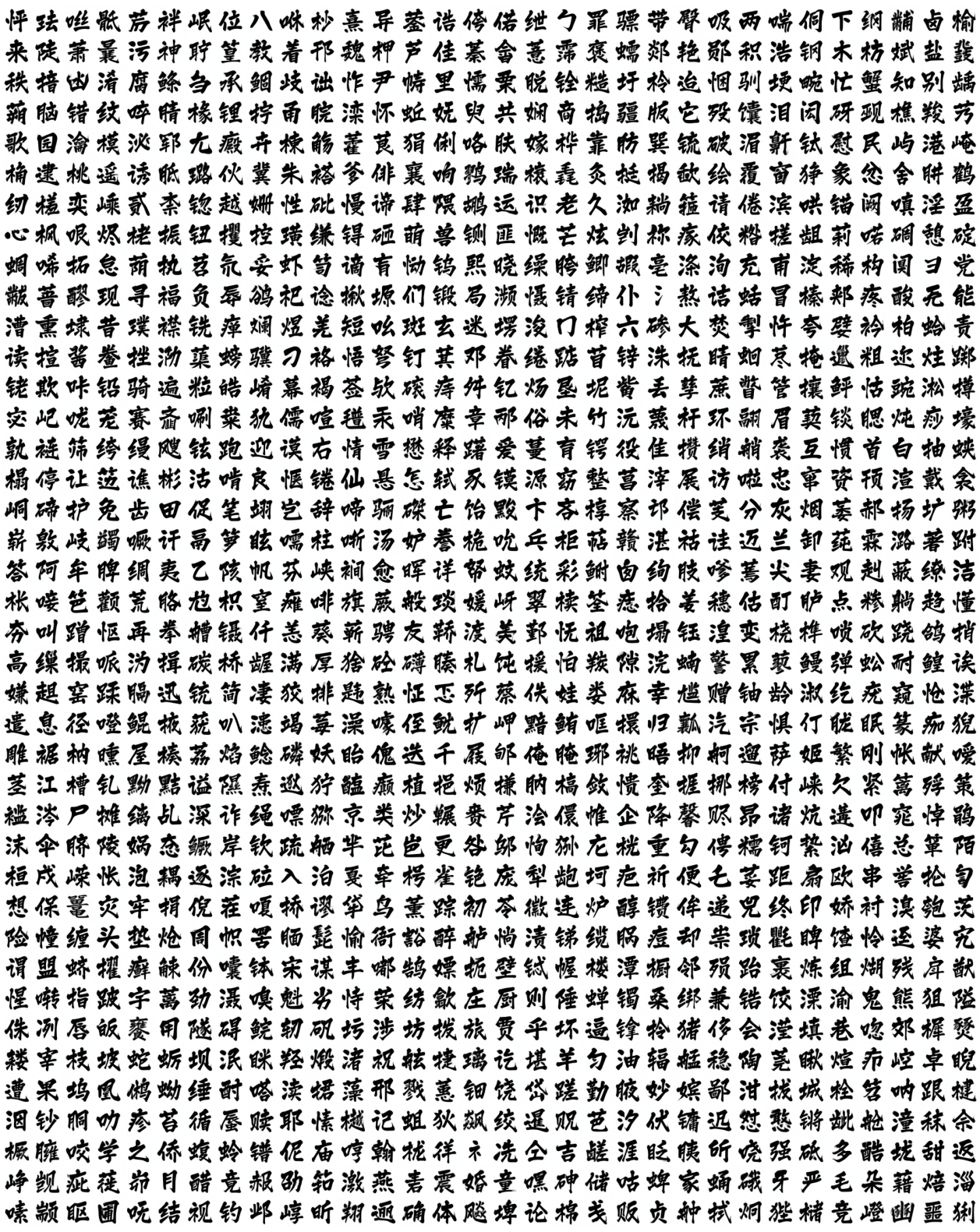


Figure S18. Generated glyphs from test fonts using GAR-Font(I_8, +NFA-8+SE, Large dataset).

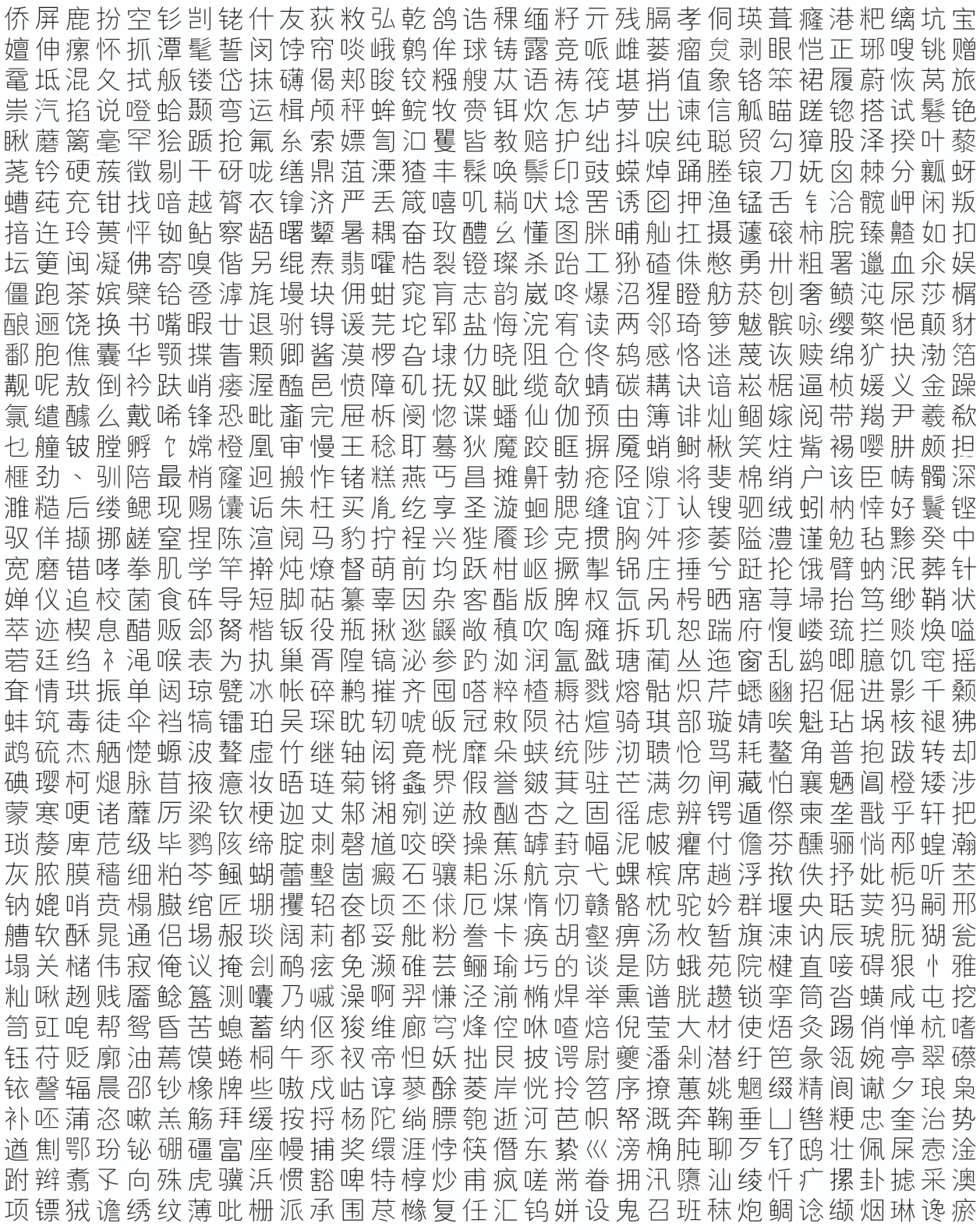


Figure S19. Generated glyphs from test fonts using GAR-Font(I_8, +NFA-8+SE, Large dataset).

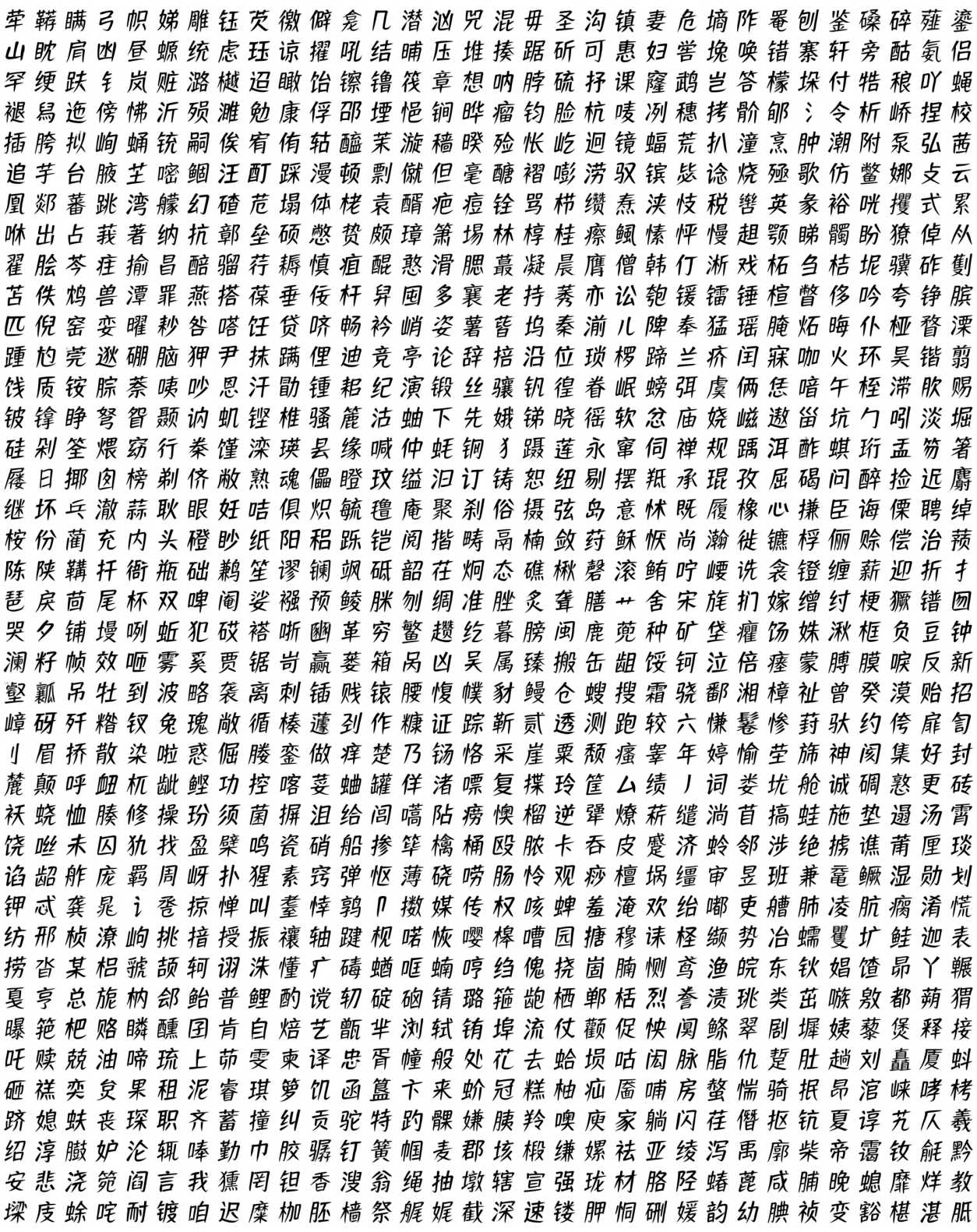


Figure S20. Generated glyphs from test fonts using GAR-Font(I_8, +NFA-8+SE, Large dataset).

References

[1] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, et al. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025. 3