# FROM CHARACTER PRIORS TO CONTROLLABLE CHINESE GLYPH GENERATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Pretrained image editors can render complex Chinese characters, yet this capability alone does not ensure transfer of a typeface from a few examples or control over its weight. We present a unified framework that turns pretrained character priors into few-shot Chinese font style transfer and explicit weight control. We first post-train Qwen-Image-Edit-2511 on font-transfer data to synthesize text-specified characters from two other-character references, without per-font tuning. We then extend the learned generator with signed weight control while keeping its generation parameters fixed. This enables font style transfer and weight control in one model and a single generation run, with the reference glyphs held fixed. On 50 adaptation-held-out targets, adaptation reduces LPIPS from 0.6832 to 0.1925 and raises foreground IoU from 0.2107 to 0.5490 relative to the unadapted base editor, with the largest gain where neither the font nor the character was seen; it also improves both metrics over the evaluated configurations of dedicated font generation models MX-Font++ and FontDiffuser. Fixed-reference control experiments change stroke weight in every requested direction and recover ordinary generation exactly at zero requested weight change, and a subspace-constrained residual improves target-weight matching, perceptual quality, and spatial fidelity together, where an unconstrained residual improves only the first. These results connect pretrained character knowledge with reference-guided, controllable glyph synthesis.

## 1 INTRODUCTION

Font generation aims to synthesize unseen glyphs while preserving the visual identity of a typeface. This capability is particularly valuable for Chinese typography, where a complete font may contain thousands of characters and designing each glyph manually is costly. Few-shot font generation reduces this burden by reconstructing a typeface from a small set of reference glyphs, enabling font completion, personalized type design, and rapid exploration of new styles. Recent dedicated font generation systems have made substantial progress in transferring style across characters from sparse references (Park et al., 2021a;b; Yang et al., 2024; Wang et al., 2025; He et al., 2025). Meanwhile, large pretrained image generators and editors have become increasingly reliable at rendering complex Chinese characters (Rombach et al., 2022; Wu et al., 2025), suggesting that large-scale image-text pretraining provides a strong prior over character structure and appearance.

These two lines of progress provide complementary capabilities. Dedicated font models explicitly learn cross-character style transfer from font-specific data, whereas pretrained image models bring broad visual knowledge and a strong prior for rendering Chinese glyphs. However, existing systems still lack a unified mechanism that exploits pretrained character priors for few-shot font generation while supporting explicit manipulation of font attributes after the reference style is fixed. In particular, a practical system should not only reproduce a target character in the style of a few reference glyphs, but also allow controlled variations, such as making the glyph lighter or heavier, without replacing references or retraining.

However, two key challenges remain. First, few-shot font generation is fundamentally a cross-character transfer problem. The reference glyphs reveal stroke treatment, terminals, proportions, and other typeface-specific cues, but not the geometry of the requested character. The model must therefore combine target-character knowledge with style information from different characters. Knowing



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

what to write does not imply knowing how that character should look in the target style. Second, explicit weight control introduces another tension. If the reference glyphs are fixed, the requested weight cannot be supplied through a new reference set. The controller must instead alter stroke mass while preserving character structure and typeface identity. Too little change fails to realize the requested weight, whereas overly aggressive changes can distort terminals, close counters, or damage the glyph. Weight controllability and glyph fidelity must therefore be considered jointly.

We address both challenges by adapting a pretrained image editor into a controllable few-shot font generator. Starting from Qwen-Image-Edit-2511, we first learn a font-generation LoRA (Hu et al., 2021) on font style-transfer data. This adaptation preserves the pretrained backbone and its native visual-language and VAE reference pathways while teaching the model to transfer a typeface across characters. At inference time, the target text specifies which character to generate, and two other-character bitmaps specify the typeface; no target-character bitmap, font file, source outline, stroke annotation, or per-font optimization is required. We then introduce a second low-rank update for weight control while freezing the generation LoRA. The control update is scaled by a signed displacement, with positive values producing heavier glyphs and negative values producing lighter ones. At zero displacement, the control update vanishes and the model reduces exactly to ordinary few-shot font generation.

Because generation and weight modification may rely on partially shared directions, we do not require the two adapters to be strictly orthogonal. Instead, inspired by subspace regularization and adapter composition (Wang et al., 2023b; Shah et al., 2024; Po et al., 2024), we softly penalize the fraction of the complete control update that lies in the output subspace of the generation update, and compare this design with both hard projection and an unconstrained residual. This encourages the controller to introduce a meaningful weight change without unnecessarily overwriting the representation learned for font generation. On 50 adaptation-held-out targets, font-specific adaptation improves the unadapted pretrained editor from an LPIPS of 0.6832 to 0.1925 and foreground IoU from 0.2107 to 0.5490. It also improves both metrics over the evaluated configurations of the dedicated font generation models MX-Font++ and FontDiffuser, although differences in pretraining, resolution, and training resources make these system-level comparisons. A six-family CJK pilot and input interventions further characterize cross-character style transfer and its dependence on the supplied references. For weight control, the constrained residual improves target-weight agreement, perceptual similarity, and spatial fidelity together, whereas an unconstrained residual changes weight only by giving up the latter two.

Our contributions are:

- We adapt a pretrained image editor for few-shot Chinese font generation, enabling cross-character style transfer from sparse glyph references while leveraging its pretrained character prior.
- We introduce signed weight control with fixed references, allowing the same model to generate lighter or heavier glyphs without changing references or retraining.
- We formulate weight control as an extension of the learned font generator and introduce soft subspace regularization, which improves stroke-weight modification and the preservation of character structure and typeface identity together.

## 2 Related Work

**Few-shot font generation.** AGIS-Net transfers shape and texture, DM-Font uses memory, and DG-Font learns deformations (Gao et al., 2019; Cha et al., 2020; Xie et al., 2021). LF-Font factorizes local styles; MX-Font and MX-Font++ aggregate multiple experts (Park et al., 2021a;b; Wang et al., 2025). Component supervision and local style matching further refine generation (Kong et al., 2022; Tang et al., 2022; Wang et al., 2023a), while FontDiffuser and AGDFont use font-specific denoising (Yang et al., 2024; He et al., 2025). Our evaluated baselines consume content bitmaps, but this is not universal: Diff-Font uses character tokens and reference images (He et al., 2024). We study how much of this task a general Chinese-capable editor can acquire through low-rank adaptation, then ask how to modify the weight inherited from its references.

**Pretrained generation and adaptation.** Latent diffusion, ControlNet, InstructPix2Pix, and IP-Adapter provide complementary textual and visual conditioning mechanisms (Rombach et al., 2022;

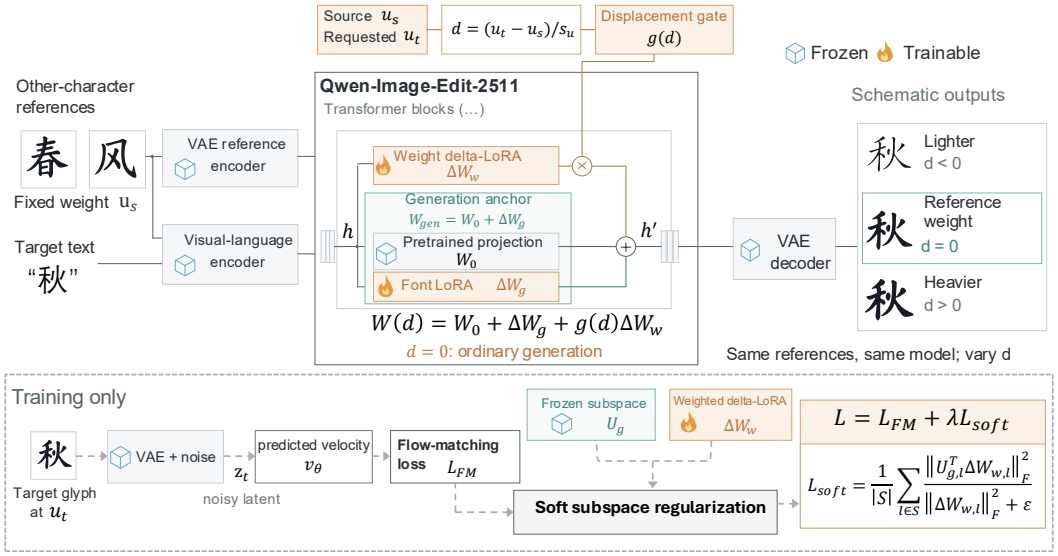



Figure 1: **Unified font generation and weight control.** Target-character text and two other-character references condition Qwen-Image-Edit-2511 through its native visual-language and VAE pathways. The font adaptation $\Delta W_g = B_g A_g$ is learned first and frozen; the weight adaptation $\Delta W_w = B_c A_c$ then applies the requested change through $g(d) = d$ with $d = (u_t - u_s)/s_u$. Both act in one sampling trajectory, both stages use flow matching, and the control term vanishes at $d = 0$. The soft penalty ($L_{\text{soft}}$ here, $\mathcal{L}_{\text{overlap}}$ in Eq. 5) applies only to control training. Outputs are schematic.

Zhang et al., 2023; Brooks et al., 2023; Ye et al., 2023). We retain Qwen-Image's native reference pathways (Wu et al., 2025) and use LoRA (Hu et al., 2021) shared across fonts, without DreamBooth-style test-subject tuning (Ruiz et al., 2023). FLUX-Font already adapts a pretrained generator with content and style LoRA modules (Li et al., 2025). Our contribution is therefore not low-rank adaptation itself, but the evaluated text-and-reference interface and its extension by a residual trained against the same frozen generator.

**Attribute control and adapter composition.** Attribute2Font controls font attributes through explicit labels (Wang et al., 2020); Concept Sliders, SliderEdit, and NumeriKontrol study low-rank or numerical control in image generation (Gandikota et al., 2024; Zarei et al., 2026; Xu et al., 2026). NIV varies a supplied target-character outline (Benedek et al., 2026), whereas our interface synthesizes that character from other-character references. These input contracts support different uses and preclude a same-input ranking.

O-LoRA uses orthogonal subspaces to reduce interference in language-model continual learning (Wang et al., 2023b). ZipLoRA merges independently trained subject and style adapters (Shah et al., 2024), and Orthogonal Adaptation designs composable spaces for separately learned concepts (Po et al., 2024). Here the control branch accesses the frozen generator throughout training, so direct addition is the baseline and a subspace penalty is optional. Our loss measures the complete update $B_c A_c$ rather than arbitrary factor coordinates. Strict inter-LoRA orthogonality did not ensure semantic disentanglement in the settings of Zhang et al. (2025); output fidelity and control response, not parameter overlap alone, must therefore determine whether our constraint is useful.

## 3 METHODS

Our method first adapts a pretrained image editor for cross-character font style transfer, then extends the learned generator with explicit weight control. Training proceeds in two stages: the first learns to use sparse glyph references, and the second learns weight variation relative to the resulting generator. Both capabilities operate within the same editor at inference (Figure 1).

### 3.1 A UNIFIED INTERFACE FOR CHARACTER, TYPEFACE, AND WEIGHT

Let $x_{f,c,u}$ denote character $c$ in typeface family $f$ at weight $u$. The ordered reference set $\mathcal{R}_{f,u_s} = (x_{f,c_1,u_s}, \ldots, x_{f,c_K,u_s})$ contains other characters, $c \notin \{c_1, \ldots, c_K\}$, at source weight $u_s$. We use $K = 2$ references. Given a requested weight $u_t$, we normalize the change by a fixed scale $s_u = 800$ and generate

$$\hat{x}_{f,c,u_t} = G_{\theta(d)}(c, \mathcal{R}_{f,u_s}; \epsilon), \qquad d = \frac{u_t - u_s}{s_u}, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{1}$$

Text specifies character identity, the reference glyphs specify the typeface, and $d$ requests a change from the reference weight. The weights $u_s$ and $u_t$ are nominal labels within each family. At $d = 0$, the model performs ordinary style transfer; positive and negative displacements request heavier and lighter glyphs, respectively. For a given character and reference set, a weight request changes only $d$, leaving the text and reference conditioning fixed.

The model is shared across fonts and requires no per-font adaptation. Its inputs are target-character text, reference bitmaps, and the requested displacement; no target-character image or source outline is needed.

### 3.2 ADAPTING CHARACTER PRIORS FOR FONT STYLE TRANSFER

Qwen-Image-Edit-2511 provides the pretrained character-rendering capability. Our generation stage teaches it to use other-character glyphs as evidence of a typeface. We retain the editor's native visual-language and VAE reference paths: the visual-language encoder processes the instruction and reference images, while the VAE encodes each reference into latent tokens. Together they provide conditioning $C(c, \mathcal{R}_{f,u_s})$ throughout generation.

We adapt the attention query, key, value, and output projections with rank-16 LoRA (Hu et al., 2021), keeping the pretrained backbone, encoders, and decoder frozen. For a projection $W_0 \in \mathbb{R}^{m \times n}$, the learned generation update is $\Delta W_g = B_g A_g$, where $B_g \in \mathbb{R}^{m \times r_g}$ and $A_g \in \mathbb{R}^{r_g \times n}$. These factors are shared across fonts. Each training example pairs a target character with two different-character references from the same font face. The reference glyphs provide conditioning, while the target glyph supplies the noisy training latent and velocity supervision. This cross-character pairing trains the editor to apply the references' style to the requested character.

We train both stages with the same flow-matching objective (Lipman et al., 2022). Each training example provides a target glyph $x = x_{f,c,u_t}$, its character $c$, references $\mathcal{R} = \mathcal{R}_{f,u_s}$, and displacement $d$. The target latent $z_0 = E(x)$ is mixed with noise as $z_t = (1 - t)z_0 + t\epsilon$, and the editor predicts the corresponding velocity:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{x,c,\mathcal{R},d,t,\epsilon}\left[\omega(t) \| v_{\theta(d)}(z_t, t, C(c, \mathcal{R})) - (\epsilon - z_0) \|_2^2\right]. \tag{2}$$

In the generation stage, $u_t = u_s$ and $d = 0$, so the objective learns transfer at the reference font's weight. Only $B_g$ and $A_g$ are optimized. We use unit loss weighting, $\omega(t) = 1$; the timestep distribution, optimizer, and training schedule are reported in Appendix B. At inference, sampling starts from Gaussian noise, and the frozen VAE decoder maps the final latent to a glyph image.

### 3.3 EXTENDING THE GENERATOR WITH SIGNED WEIGHT CONTROL

Once style-transfer training is complete, we freeze the learned generator as the anchor for weight control. This is the same generation model evaluated in Section 4. Freezing it provides a fixed foundation from which a new residual learns the requested weight variation. Absorbing LoRA scaling into the factors, each adapted projection becomes

$$W(d) = \underbrace{W_0 + B_g A_g}_{\text{frozen generation anchor}} + \underbrace{d B_c A_c}_{\text{trainable weight residual}}, \qquad d = \frac{u_t - u_s}{s_u}. \tag{3}$$

The control branch uses rank $r_c = 16$ and is initialized with $B_c A_c = 0$. Control training uses genuine target glyphs at $u_t$ and reference glyphs at $u_s$ from the same family. It reuses Eq. 2, updating only $B_c$ and $A_c$. Positive and negative displacements are trained together with the fixed signed gate $g(d) = d$. Thus lightening and thickening share one residual, applied in opposite parameter-space directions, and target-weight supervision determines the resulting visual changes.

The frozen anchor and signed gate give $W(0) = W_0 + B_g A_g$ at every training step. Ordinary generation is therefore retained as the zero-displacement setting; with identical conditioning, initial noise, and sampling, this setting also recovers the anchor's output. Nonzero requests activate the residual within the same generation process.

For a fixed request, the generation and control updates combine into a single effective adapter:

$$B_{\text{eff}}(d) = [B_g \quad dB_c], \qquad A_{\text{eff}} = \begin{bmatrix} A_g \\ A_c \end{bmatrix}, \qquad B_{\text{eff}}(d)A_{\text{eff}} = B_g A_g + dB_c A_c. \tag{4}$$

Its rank is at most $r_g + r_c$ (32 with our two rank-16 branches). Both updates act throughout one sampling trajectory, so the model synthesizes the requested character in the reference typeface with the weight request applied during generation.

### 3.4 Regularizing overlap with the generation adaptation

A control residual can alter the style-transfer behavior learned in the first stage. At the same time, weight is part of the appearance already modeled by the generator, so generation and control may benefit from shared update directions. We therefore introduce a soft overlap penalty to regulate the control update while allowing target-glyph supervision to retain useful shared directions. For each adapted projection $\ell$, let $U_{g,\ell}$ be a fixed orthonormal basis for the effective column space of $\Delta W_{g,\ell} = B_{g,\ell} A_{g,\ell}$. Over the constrained projections $\mathcal{S}$, the soft-constrained residual is trained with

$$\mathcal{L}_{\text{overlap}} = \frac{1}{|\mathcal{S}|} \sum_{\ell \in \mathcal{S}} \frac{\|U_{g,\ell}^\top B_{c,\ell} A_{c,\ell}\|_F^2}{\|B_{c,\ell} A_{c,\ell}\|_F^2 + \varepsilon}, \qquad \mathcal{L} = \mathcal{L}_{\text{FM}} + \lambda \mathcal{L}_{\text{overlap}}. \tag{5}$$

The numerator measures the control update's squared Frobenius energy within the generation update's output subspace. Normalization expresses it as a stabilized fraction of the complete control update's energy. Because the penalty is defined on $B_c A_c$, it is invariant to equivalent factorizations of that update. The basis is extracted once from the frozen generation adapter, and $\lambda$ is selected on validation data.

This formulation balances the target-weight objective against overlap with the learned generation directions. Setting $\lambda = 0$ gives the unconstrained residual. A hard-projected variant instead uses $(I - U_g U_g^\top)\widetilde{B}_c A_c$, excluding the generation subspace throughout training and inference. All three variants retain the same frozen anchor, signed gate, and one-trajectory inference structure. Section 5.2 evaluates weight response and generation retention; Appendix F.5 gives the full constrained runs and baselines.

## 4 Reference-Conditioned Generation

### 4.1 Protocol and evidence map

We organize the evaluation around four questions. **Q1**: how much does task adaptation improve reference-conditioned generation? (§4.2). **Q2**: does the improvement extend to fonts and characters that adaptation never saw? (§4.3). **Q3**: how does the interface respond when references or the instruction change? (§4.4). **Q4**: can an explicit request change the inherited weight without destroying glyph geometry? (§5). This section evaluates the Seed1 generation LoRA; Section 5 adds control residuals to that frozen anchor, with the full protocol in Appendix F.5.

The evaluation panel contains 50 targets, grouped by exposure (Appendix A): 25 **UFUC** targets, whose font and character are both unseen from adaptation, and 25 **SFUC** targets, whose font appears in the recorded adaptation lists while the character is unseen. The panel therefore covers no UFSC cell (unseen font, seen character). Twelve supervised diagnostics are reported separately.

We report AlexNet LPIPS, grayscale SSIM, L1 and foreground IoU (Zhang et al., 2018; Wang et al., 2004) at generation IoU threshold 0.5, resizing native baseline outputs bicubically to $256^2$. A blank image reaches SSIM 0.7041 at zero IoU on the combined evaluation and diagnostic panels (Table 9), and no single pixel metric separates character correctness from font consistency (Appendix D).

Table 1: Reference-conditioned generation on the 50 evaluation targets, split by exposure: UFUC (unseen font, unseen character, $n = 25$) and SFUC (seen font, unseen character, $n = 25$). Target-equal means, each averaged over five bootstrap resamples (bootstrap-5); two other-character references; no content template or test-font optimization; no weight request ($d = 0$). Exposure follows the generation adaptation's records. Only the editor pair runs without a public content bitmap (Table 4), so the dedicated rows are system-level records; native outputs are scored on the common $256^2$ canvas. Ours is the frozen anchor the weight residual composes with.

| | Evaluation (50) | | UFUC (25) | | SFUC (25) | |
|---|---|---|---|---|---|---|
| Configuration | LPIPS↓ | IoU↑ | LPIPS↓ | IoU↑ | LPIPS↓ | IoU↑ |
| Base editor | 0.6832 | 0.2107 | 0.6860 | 0.2430 | 0.6803 | 0.1784 |
| Ours | **0.1925** | **0.5490** | **0.1522** | **0.6631** | **0.2327** | **0.4350** |
| MX-Font++ | 0.3197 | 0.5197 | 0.3071 | 0.6045 | 0.3323 | 0.4349 |
| FontDiffuser (official) | 0.4216 | 0.3812 | 0.4201 | 0.4492 | 0.4232 | 0.3133 |
| LF-Font (official inference) | 0.3246 | 0.4859 | 0.2978 | 0.6006 | 0.3514 | 0.3712 |
| AGDFont (local substitute, native $K$=4)[‡] | 0.6341 | 0.2203 | 0.6781 | 0.2262 | 0.5901 | 0.2144 |

[‡]AGDFont is a local substitute whose content and style encoders were randomly initialized and trained jointly, because the official pretrained encoder weights are not released (Appendix C.1); it is trained and sampled with four same-font references, so it differs from the others in both reference count and input rendering (two-reference record in Table 7).

### 4.2 Main comparison

Task adaptation substantially improves target similarity (Table 1): LPIPS falls from 0.6832 to 0.1925 on the 50-target panel and foreground IoU rises from 0.2107 to 0.5490, so pretraining alone does not supply this behavior under the task interface. Both exposure cells improve, including the UFUC targets, where neither the font nor the character was seen (LPIPS 0.6860 to 0.1522 and IoU 0.2430 to 0.6631, against 0.6803 to 0.2327 and 0.1784 to 0.4350 on SFUC). This measures task weights under one fixed configuration rather than the editor's tuned best case, since no prompt or guidance search is included (Appendix B); Figure 2 shows the change qualitatively.

The dedicated systems additionally receive a public target-character bitmap and differ in resolution, exposure, and pretraining, so this is a system-level comparison rather than an architecture ranking. LF-Font uses its completed 400k/200k local retraining with official inference; AGDFont is a local substitute at its native $K = 4$ with training-style renders, differing in both reference count and input rendering (Appendix B.2). Full tables and old-data diagnostics are in Appendices C and B.1.

### 4.3 Transfer to unseen fonts and characters

The 50 evaluation identities are excluded from both the adaptation target glyphs and reference images, so the improvement is measured on characters adaptation never targeted. The two cells differ in font exposure: in UFUC neither the target font nor the character appears in the adaptation records, while in SFUC the font does. Twenty-five of the 50 target fonts are absent from our adaptation records and twenty-one from the dedicated baselines' pool (Appendix A).

The six-family pilot examines transfer across fonts: six families, six common CJK characters spanning three stroke tiers, fixed $K = 2$ references, two seeds, 72 outputs per model, with the editor pair sharing initial-noise bytes and MX-Font++ producing 36 duplicate seed pairs. Family-equal LPIPS falls from 0.6521 to 0.2588 against the base editor and from 0.3747 to 0.2588 against MX-Font++, with paired family-cluster bootstrap intervals $[-0.4295, -0.3555]$ and $[-0.1510, -0.0840]$. Figure 3 resolves those means by family.

### 4.4 Input interventions

The remaining experiments change the interface rather than the weights (Appendix E). Two references work best for the adaptation trained with two: across 50 targets, two nested reference sets and two seeds (800 outputs), $K = 4$ worsens LPIPS on all 50 targets after averaging and $K = 8$ worsens all four metrics on every target.




Figure 2: **Generation against the evaluated baselines.** Twelve targets span six styles (F1–F6). Editors use text and both references; dedicated systems also use public content. All checkpoints match Table 1. The AGDFont row is the local substitute in its native four-reference configuration (Appendix B.2); its two starred cells come from the earlier two-reference round. Starred columns are prespecified qualitative requests excluded from that table, and the final row repeats each target's ground truth, which is evaluation-only.

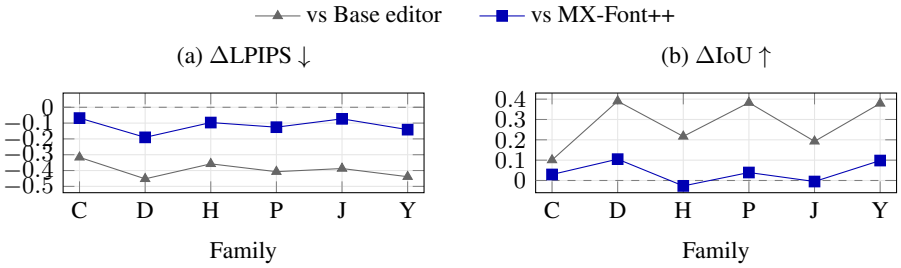
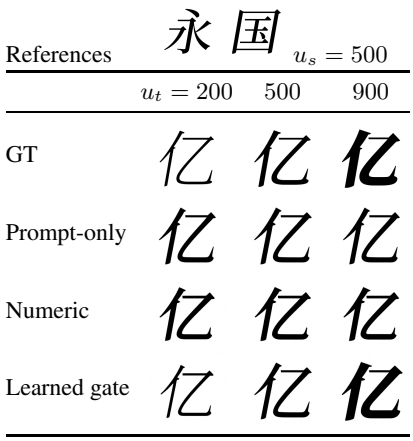



Figure 3: **Family-level paired effects in the six-family pilot.** Differences are Ours minus comparator; lower LPIPS and higher IoU favour Ours. Each point averages six characters and two seeds within a family. C: ChenYuluoyan, D: DelaGothicOne, H: HachiMaruPop, P: PottaOne, J: YujiSyuku, Y: YuseiMagic. LPIPS improves in all six families against both comparators, and IoU in all six against the unadapted editor; the two below-zero IoU points are H and J against MX-Font++ (all 216 predictions: Appendix C.3).

Adding a public rendering of the target character is likewise harmful: across 46 targets from 27 font keys and two seeds (368 outputs, four pixel-identity cases excluded), template addition worsens LPIPS for Ours on all 46 targets and IoU on 44, most sharply for the adapted arm, which is tuned to a 2-reference, template-free contract. Changing only the reference font, with the target and reference characters, prompt, and seed fixed, shifts our output's LPIPS margin toward the new-font GT on 50/50 targets, against 42/50 for the base editor and 48/50 for MX-Font++ (Appendix C.4).

Native baseline settings and the AGDFont reference-count diagnosis appear in Appendix B.2.

**(a) Fixed-reference weight sequence**




**(b) Ink response and fidelity**

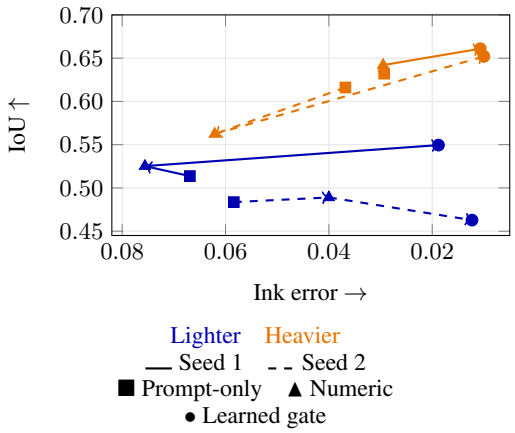



Figure 4: **Weight response and its fidelity cost.** (a) Sanmu Xiesong, U+4EBF, Seed 1 (frozen row-order selection). References stay at 500; GT is evaluation-only. The unconstrained learned gate uses displacement, other interfaces numeric text; Appendix Figure 15 retains both seeds and the base editor. (b) Held-out-family means, 90 light/50 heavy instances per seed; arrows connect interfaces. Ink error is absolute foreground-fraction error, and ink and IoU use gray $< 200/255$.

## 5 Weight Control Under Fixed References

With two fixed references at $u_s = 500$, signed displacement $d$ requests a lighter ($d < 0$) or heavier ($d > 0$) glyph; at $d = 0$, the residual vanishes and the interface recovers ordinary font transfer. We ask whether a signed request changes stroke ink, and whether a subspace constraint can do so while retaining the learned generator. Both hold: the constrained endpoints give the strongest target-weight match of the compared interfaces, and they leave ordinary generation untouched at zero displacement.

### 5.1 Signed Requests Change Ink

We compare prompt-only, numeric-conditioned, and learned-gate interfaces on 170 held-out-family and 1,003 seen-family requests per seed (weight-stream splits; Appendix F). Figure 4 shows the fixed-reference sequence and the resulting ink error: the learned gate is lowest in all four held-out-family direction–seed cells, with the seen-family panel repeating this aggregate response. A signed request therefore changes stroke ink consistently across directions, seeds, and both panels.

### 5.2 Retaining Generation While Controlling Weight

We fix $g(d) = d$, since paired gates show no consistent learned-gate advantage (Appendix F). Final 256-step endpoints use 768 test requests, two seeds, and validation-selected soft $\lambda = 0.01$. Figure 5 shows examples; Appendix F.5, Table 21, gives full scores.

**Active control and glyph structure.** Soft and hard reduce the generator's ink error from 0.0451 to 0.0102 and 0.0118, improve LPIPS from 0.1820 to 0.1537 and 0.1565, and improve IoU from 0.5472 to 0.5832 and 0.5710: both improve target-weight agreement, perceptual quality, and spatial fidelity together, and soft leads hard on all three. A, our unconstrained baseline on the same anchor, data, and budget, lowers ink error to 0.0135 but leaves LPIPS at 0.1876 and IoU at 0.5208, below the generation anchor, so without the constraint part of the ink response is bought by degrading the generator. The selected examples retain open contours and interior layout.

**Exact recovery at zero displacement.** Both constrained endpoints reproduce Seed1 on all 48 ordinary-generation and 240 zero-displacement test outputs; PNG, pixel, latent, noise, and prompt hashes agree. The gate study also recovers its anchor on all 248 retention outputs (Appendix F.4). Recovery follows from the frozen anchor and zero gate.

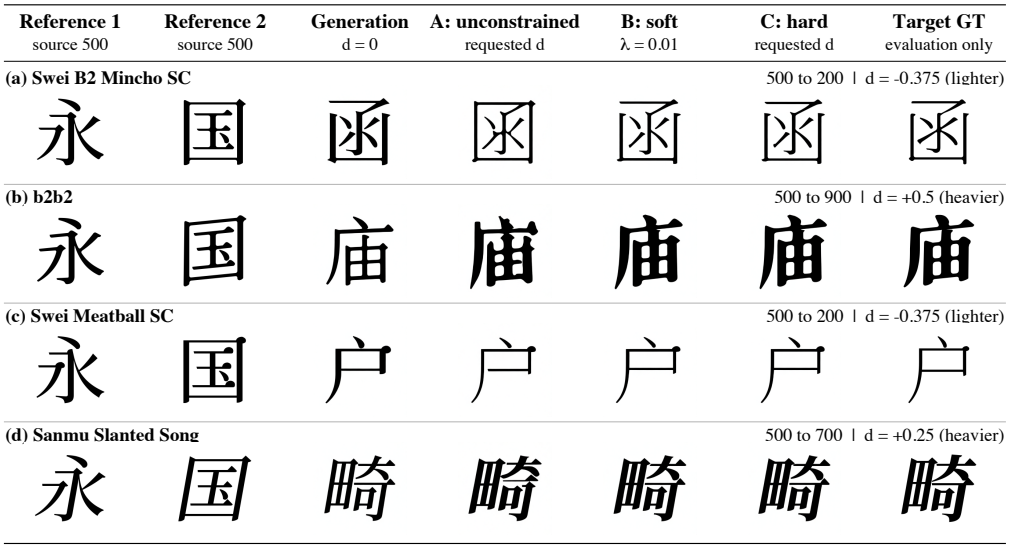



Figure 5: **Weight control with retained glyph structure.** Four selected test cases (seed 20260726), shown as complete original images at a common scale. Two source-500 references condition generation; GT is for evaluation only. B and C retain the open contour in (a), interior layout in (b), and slanted design in (d) more closely than A and are visually similar to each other; in (c) they close a visible gap left by A without improving the GT metrics. Full-test scores appear in Appendix F.5, Table 21.

**Complex fonts.** On 18 handwritten, calligraphic, and display faces, soft and hard ink responses are close ($-16.74\%/+19.57\%$ versus $-17.01\%/+19.78\%$). At similar achieved response, soft has lower LPIPS to zero (by 0.0008–0.0028) and higher IoU (by 0.0004–0.0031) across three matching tolerances, so its advantage carries over to faces outside the regular test set. These diagnostics include the intended weight change and lack nonzero target-weight GT. Stronger thinning on six faces gives nearly identical ink responses. Full results: Appendix F.5.

**Summary.** The subspace-constrained residual is the best weight-control interface we evaluate. Soft cuts target-weight ink error by a factor of 4.4 relative to the generation anchor and also improves LPIPS and IoU, ahead of the unconstrained residual and hard projection on all three metrics and on the response-matched complex-font subsets, whereas the unconstrained residual exhibits the very tradeoff this design targets. We therefore take soft as the default and keep hard projection as the strict-separation alternative. The endpoints are single training runs; the ordering is not significance-tested.

## 6 Conclusion

We presented a unified framework for few-shot Chinese font style transfer and explicit weight control, building on the character priors of a pretrained image editor. Task-specific LoRA adaptation generates a text-specified character in the typeface of two other-character references without per-font tuning, improving perceptual similarity and foreground overlap over the unadapted editor and the evaluated dedicated font generation models on adaptation-held-out targets.

Building on the frozen generator, a signed, displacement-scaled residual enables lighter and heavier outputs in a single generation run with the references fixed, and exactly recovers ordinary generation at zero displacement. Soft subspace regularization of that update improves target-weight agreement, perceptual similarity, and spatial fidelity together, the best scores among the tested control variants, connecting pretrained character knowledge with reference-guided, controllable glyph synthesis.

## AI USE STATEMENT

AI models are used as part of the systems studied in this work. In particular, we adapt a pretrained image editor for few-shot Chinese font style transfer and explicit weight control, and evaluate its outputs alongside those of other font generation models. Details of the models, training procedures, data, and experimental protocols are provided in the main paper and appendix. Large language models were additionally used for language editing and polishing of the manuscript. All research methodology, experimental analysis, and scientific conclusions were reviewed and verified by the authors, who take full responsibility for the final manuscript.

## REFERENCES

Nadav Benedek, Ariel Shamir, and Ohad Fried. Niv: Neural axis variations for variable font generation. *arXiv preprint arXiv:2606.05261*, 2026.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402. IEEE, 2023.

Junbum Cha, Sanghyuk Chun, Gayoung Lee, Bado Lee, Seonghyeon Kim, and Hwalsuk Lee. Few-shot compositional font generation with dual memory. In *European conference on computer vision*, pp. 735–751. Springer, 2020.

Rohit Gandikota, Joanna Materzyńska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. In *European Conference on Computer Vision*, pp. 172–188. Springer, 2024.

Yue Gao, Yuan Guo, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. Artistic glyph image synthesis via one-stage few-shot learning. *ACM Transactions on Graphics (ToG)*, 38(6):1–12, 2019.

Haibin He, Xinyuan Chen, Chaoyue Wang, Juhua Liu, Bo Du, Dacheng Tao, and Qiao Yu. Diff-font: Diffusion model for robust one-shot font generation. *International Journal of Computer Vision*, 132(11):5372–5386, 2024.

Yan He, Xiang Xiang, and Xiaofei Liao. Few-shot font generation via attribute-guided diffusion with style contrastive learning. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pp. 387–402. Springer, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Yuxin Kong, Canjie Luo, Weihong Ma, Qiyuan Zhu, Shenggao Zhu, Nicholas Yuan, and Lianwen Jin. Look closer to supervise better: One-shot font generation via component-based discriminator. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 13472–13481. IEEE, 2022.

Zheyu Li, Honglie Wang, Yan-Ming Zhang, and Cheng-Lin Liu. Flux-font: stylized font generation based on fine-tuning pre-trained diffusion transformer. In *Asian Conference on Pattern Recognition*, pp. 324–338. Springer, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Song Park, Sanghyuk Chun, Junbum Cha, Bado Lee, and Hyunjung Shim. Few-shot font generation with localized style representations and factorization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 2393–2402, 2021a.

Song Park, Sanghyuk Chun, Junbum Cha, Bado Lee, and Hyunjung Shim. Multiple heads are better than one: Few-shot font generation with multiple localized experts. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13880–13889. IEEE, 2021b.

Ryan Po, Guandao Yang, Kfir Aberman, and Gordon Wetzstein. Orthogonal adaptation for modular customization of diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7964–7973. IEEE, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22500–22510. IEEE, 2023.

Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora: Any subject in any style by effectively merging loras. In *European Conference on Computer Vision*, pp. 422–438. Springer, 2024.

Licheng Tang, Yiyang Cai, Jiaming Liu, Zhibin Hong, Mingming Gong, Minhu Fan, Junyu Han, Jingtuo Liu, Errui Ding, and Jingdong Wang. Few-shot font generation by learning fine-grained local styles. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7885–7894. IEEE, 2022.

Chi Wang, Min Zhou, Tiezheng Ge, Yuning Jiang, Hujun Bao, and Weiwei Xu. Cf-font: Content fusion for few-shot font generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1858–1867. IEEE, 2023a.

Weihang Wang, Duolin Sun, Jielei Zhang, and Longwen Gao. Mx-font++: Mixture of heterogeneous aggregation experts for few-shot font generation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, 2023b.

Yizhi Wang, Yue Gao, and Zhouhui Lian. Attribute2font: Creating fonts you want from attributes. *ACM Transactions on Graphics (TOG)*, 39(4):69–1, 2020.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Yangchen Xie, Xinyuan Chen, Li Sun, and Yue Lu. Dg-font: Deformable generative networks for unsupervised font generation. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 5126–5136. IEEE, 2021.

Zhenyu Xu, Xiaoqi Shen, Haotian Nan, and Xinyu Zhang. Numerikontrol: Adding numeric control to diffusion transformers for instruction-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4389–4399, 2026.

Zhenhua Yang, Dezhi Peng, Yuxin Kong, Yuyi Zhang, Cong Yao, and Lianwen Jin. Fontdiffuser: One-shot font generation via denoising diffusion with multi-scale content aggregation and style contrastive learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 6603–6611, 2024.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Arman Zarei, Samyadeep Basu, Mobina Pournemat, Sayan Nag, Ryan A Rossi, and Soheil Feizi. Slideredit: Continuous image editing with fine-grained instruction control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14430–14439, 2026.

Andi Zhang, Xuan Ding, Haofan Wang, Steven McDonagh, and Samuel Kaski. Rethinking inter-lora orthogonality in adapter merging: Insights from orthogonal monte carlo dropout. *arXiv preprint arXiv:2510.03262*, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824. IEEE, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

## GUIDE TO THE EVIDENCE APPENDICES

The appendices provide the evidence behind the main results: the data and exposure records, the baseline reproduction audit, the complete generation and control panels, and the retained failure cases.

| Appendix | Reader question |
|---|---|
| A: Task and data | What information and training exposure does each panel have? |
| B: Reproduction | Which settings and identities produced the evidence? |
| C: Generation | What do the baseline and multi-character panels show? |
| D: Character identity | Is the output the requested character, and does it follow a substituted reference face? |
| E: Interface | How do reference count, template addition, and statistical weighting change results? |
| F: Control and protocol | What do the existing control panels establish, and what do the constrained runs and their baselines show? |
| G: Failure index | Where can readers inspect the retained failure cases? |

## A TASK, DATA, AND EXPOSURE

The input contract in Section 3 applies to actual model inputs rather than to display columns. A public content image specifies the requested character in another font; a style reference supplies a different character from the desired font; target GT is a training label or evaluation image; a displayed source500 image is not a conditional-editor input.

The generation character split excludes the 50 evaluation identities from both target supervision and the reference images used during task adaptation. We label each target by exposure: *UFUC* has an unseen font and an unseen character, *SFUC* a seen font and an unseen character, *SFSC* a seen font and a seen character. A font is seen when it appears in the recorded adaptation font list, and a character is seen when it appears as an adaptation target. The evaluation panel is 25 UFUC and 25 SFUC targets; the 12 supervised diagnostics are SFSC. Every evaluation character is unseen, so the panel contains no UFSC target and the two evaluation cells differ only in font exposure.

The recorded baseline adaptation lists contain 181 fonts for our model and 389 for the common baseline pool, the same list the LF run trains on (Appendix B); both are drawn from 1,100-character lists. These are manifest-level exposure summaries. Their exposure labels differ on 21 of the 62 targets. Twenty-five of the 50 evaluation fonts are absent from our recorded list, twenty-one from the baseline pool, and ten targets are absent from both recorded font and character lists, spanning only OPlusSans and SourceHanSans family groups. On those ten targets, which lie inside the evaluation pool, our model obtains LPIPS 0.1243 and IoU 0.6840 in the frozen per-request records. Table 2 reports the per-cell metrics.

Name aliases, family grouping, shared ancestry and model-selection exposure are distinct from simple face membership; independent-family claims require their joint audit. Foundation pretraining exposure is unknown, and name and ancestry checks cannot rule out undocumented lineage.

For the weight request, the held-out-family panel's three families are held out from the weight-conditioning stream and the seen-family panel's nine are seen by that stream. Holdout is per stream, and the exposure of the composition is the union of the two: a family or character withheld from the residual's stream may still appear in the anchor's transfer stream, so these panels do not establish whole-run holdout. Generation IoU thresholds foreground at 0.5, whereas the control panels use $200/255$ for binary IoU and ink coverage; their IoUs are not combined.

Statistical summaries preserve their original units: target-equal and font-equal generation means are distinguished, and current control first aggregates within character/weight sequences before giving each family equal weight. Repeated seeds measure sampling variation, not training replication. The three-family gate intervals are exploratory.

Table 2: Exposure cells of the frozen panel. UFUC and SFUC are the 25 evaluation targets whose font and character are both unseen, or whose font is seen while the character is unseen; the 12 supervised diagnostics are SFSC, seen in both. Cell membership follows the recorded exposure of the generation adaptation, and the dedicated systems' own exposure labels differ on 21 of the 62 targets. LPIPS and IoU follow Table 1.

| | UFUC (25) | | SFUC (25) | | SFSC diagnostic (12) | |
|---|---|---|---|---|---|---|
| Method | LPIPS↓ | IoU↑ | LPIPS↓ | IoU↑ | LPIPS↓ | IoU↑ |
| Ours | **0.1522** | **0.6631** | **0.2327** | **0.4350** | **0.2421** | **0.4408** |
| Base editor | 0.6860 | 0.2430 | 0.6803 | 0.1784 | 0.6821 | 0.1920 |
| MX-Font (official) | 0.3005 | 0.5994 | 0.3520 | 0.3999 | 0.3278 | 0.4255 |
| MX-Font++ (retrained) | 0.3071 | 0.6045 | 0.3323 | 0.4349 | 0.3374 | 0.3839 |
| MX-Font (retrained) | 0.3151 | 0.5857 | 0.3653 | 0.3630 | 0.3554 | 0.3780 |
| FontDiffuser (official) | 0.4201 | 0.4492 | 0.4232 | 0.3133 | 0.4285 | 0.3258 |
| FontDiffuser (retrained) | 0.4776 | 0.3928 | 0.4288 | 0.3087 | 0.4576 | 0.2648 |
| LF-Font (official inference) | 0.2978 | 0.6006 | 0.3514 | 0.3712 | 0.3420 | 0.4134 |
| AGDFont (local substitute, native $K{=}4$)[‡] | 0.6781 | 0.2262 | 0.5901 | 0.2144 | 0.5680 | 0.2429 |
| Content copy | 0.2706 | 0.4926 | 0.3738 | 0.3575 | 0.3593 | 0.3086 |
| Blank image | 0.7232 | 0.0000 | 0.6282 | 0.0000 | 0.6513 | 0.0000 |

The three cells partition the 62 frozen correct-reference requests and combine to the pooled values used elsewhere. [‡]AGDFont is shown in its native inference configuration, four references drawn with replacement plus training-style content and reference renders; its two-reference record is in Table 7. LF-Font uses the retrained checkpoint with official inference.

Table 3: Model identities and evidence roles. All control branches use the frozen Seed1 generation LoRA. Shared anchor, architecture, or initialization does not imply identical trained residuals.

| Configuration | Weight identity | Evidence role |
|---|---|---|
| Base editor | Qwen-Image-Edit-2511 without task weights | Generation-panel comparator |
| Ours, generation | Seed1 generation LoRA, rank 16, step 3,000 | Generation panels and frozen control anchor |
| Ours, prompt-only | Frozen generation anchor; requested weight appears only in the instruction | Prompt-based control reference |
| Ours, numeric | Same anchor with the recorded numeric-conditioned interface | Request-channel comparison |
| Ours, learned gate | Frozen anchor plus a trained rank-16 residual and learned gate | Existing unconstrained control evidence |
| Ours, linear gate | Same anchor plus a separately trained rank-16 residual, $g(d) = d$ | Existing unconstrained gate comparison |
| A: unconstrained | Same anchor with a linear gate | Unconstrained control baseline |
| Soft / hard constraint | Same anchor; linear signed gate, trainable rank-16 residual; constraint varies | Completed comparison, Section 5.2 |

Table 5 gives a compact summary of the training corpus and its use terms. The training corpus comprises 1,000 open-source fonts drawn from Google Fonts, the Chinese Web Font Project (`KonghaYao/chinese-free-web-font-storage`), the Free Font collections `jaywcjlove/free-font` and `wordshub/free-font`, and further open-source font databases; their glyph sets cover the Chinese characters of the GB2312 standard, and the two-reference rows used for adaptation are sampled from this corpus (Appendix B). The study is for scientific use only, and glyphs produced by this model may not be used commercially.

Table 4: Input and output contracts of the evaluated configurations. $K$ counts style references only. Raster sizes are native outputs, not the common scoring canvas. No listed inference configuration updates parameters for the test font.

| Method | Content image | $K$ | Other input | Output |
|---|---|---|---|---|
| MX-Font / MX-Font++ | Required | 2 | Content image specifies character | $128^2$ raster |
| LF-Font | Required | 2 | Character decompositions | $128^2$ raster |
| FontDiffuser* | Required | 2 | Content image specifies character | $96^2$ raster |
| AGDFont (local substitute) | Required | 4 | Content image specifies character | $80^2$ raster |
| Base editor / Ours | **None** | 2 | Target-character text; optional weight displacement | $256^2$ raster |

*Multi-reference configurations include local adaptations; $K = 2$ is the comparison setting, not each original method's unique supported count. The editor configurations use target text and reference images without a target-character bitmap or outline; our adapted version additionally supports displacement control. The dedicated baselines above require a content image. AGDFont is listed at $K = 4$, the reference count used both in its training and by its released sampler, and it is the one row whose $K$ is not the comparison setting; its native row in Table 1 also uses training-style content and reference renders.

Table 5: Compact summary of the training corpus and its use terms. All training fonts are open-source. The study is for scientific use only, and glyphs produced by this model may not be used commercially.

| Item | Summary |
|---|---|
| Training fonts | 1,000 open-source fonts |
| Font sources | the Chinese Web Font Project (`KonghaYao/chinese-free-web-font-storage`); the Free Font collections `jaywcjlove/free-font` and `wordshub/free-font`; Google Fonts; further open-source font databases |
| Licensing | Every training font and source database is released under an open-source license |
| Character coverage | Chinese characters in the GB2312 standard |
| Adaptation rows | 49,006 two-reference font-transfer rows sampled from this corpus (Appendix B) |
| Use restriction | Scientific use only; glyphs generated by this model may not be used commercially |

## B Method, Reproducibility, and Baseline Diagnostics

The source package provides the figure assets, per-request metric records, calculation scripts, model identities, and input configurations. The transfer panels evaluate the generation adaptation itself, and the control panels evaluate that same adaptation with the weight residual composed on it. The qualitative additions are excluded from the quantitative panels.

**Training.** Our model uses rank/alpha 16, 3,000 steps, 49,006 two-reference training rows sampled from the 1,000-font corpus of Appendix A, and $256^2$ targets. Its recorded optimizer is AdamW with learning rate $10^{-4}$, $\beta = (0.9, 0.999)$, weight decay $10^{-4}$, epsilon $10^{-8}$, gradient clipping 1.0, and a constant learning-rate scheduler. LoRA dropout and effective reference-set dropout are zero: the no-content dataset path overrides the saved reference-dropout default of 0.1. Timesteps are logit-normal with mean 0 and standard deviation 1 before the sigmoid, the training noise-time transform $t \mapsto st/[1 + (s - 1)t]$ has $s = 1$ (the identity), and the velocity MSE has $\omega(t) = 1$. The recorded control stage freezes that Seed1 generation LoRA and trains 480 rank-16 factor tensors with 16,384 draws (256 steps, global batch 64). The anchor is shared, while the learned- and linear-gate residuals are trained separately. Only the learned gate parameterization adds the 82,689 gate parameters. Paired inference uses the same inputs and noise. The four request channels compared in Section 5 — prompt-only, numeric, and the two displacement parameterizations — are whole-interface comparisons with different prompts.

**Anchor identity and zero displacement.** The control panels use the same trained Seed1 generation LoRA as the generation experiments. For the displacement branches, recovery at $d = 0$ follows from freezing the anchor and zero gating, not from zero initialization or subspace regularization. With prompts, references, noise, and sampling held fixed, the retained regression panel verifies

PNG-byte and final-latent equality (Appendix F.4). The soft and hard variants pass their own output check with separate counts (Appendix F.5).

**Effective generation inference.** Both editor arms use the same manual joint VL/VAE runner. The call sets `true_cfg_scale=1.0`, `guidance_scale=None`, `negative_prompt=""`, and `max_sequence_length=1024`. The transformer has no guidance embedding; the CFG negative pass is inactive at scale 1.0. The loaded flow-matching Euler scheduler receives 40 linearly spaced input sigmas from 1 to $1/40$, followed by its configured timestep shift. The task prompt translates as "Generate the Chinese character [c], preserving the font style of the reference glyphs." The exact Chinese string is constructed below using Python Unicode escapes, where `c` is the requested character:

```
prefix = "\u751f\u6210\u6c49\u5b57\u201c"
suffix = ("\u201d\uff0c\u4fdd\u6301\u53c2\u8003"
          "\u5b57\u5f62\u7684\u5b57\u4f53\u98ce"
          "\u683c\u3002")
prompt = prefix + c + suffix
```

No independent base-editor prompt or guidance search is included in the comparison, so the comparison measures task weights under one fixed inference configuration. The released dual-image pipeline documents other settings; a tuned base-editor arm selected on a held-out split and a check that this runner reproduces the released pipeline are not part of this record. The panel does not separate emitting an isolated glyph from using the references, because the base editor additionally returns scene-like composites.

**Evidence checks and statistical units.** Generation evaluation and diagnostic means use the frozen per-request scores. SSIM uses a uniform $7 \times 7$ window, and generation IoU thresholds foreground at 0.5. K800 and template368 preserve target-equal estimands; family-equal curves and bootstrap intervals remain separate. Current control averages axes within family–character before giving families equal weight. Repeated seeds are sampling replicates. The original 1,020 adapted outputs from the held-out-family panel and 340 base-editor outputs were retained, with different upstream coverage. Completed paired-gate panels have 680 outputs from the held-out-family panel and 4,012 from the seen-family panel. Ordinary retention has 496 outputs. Reused scoring rows are not additional predictions.

### B.1 Bounded old-data baseline diagnostics

The LF comparison uses the completed local retraining with official inference described below. AGDFont retains its local substitute checkpoint and native four-reference configuration. The AGD continuation experiments are separate diagnostics on old training data.

**LF retraining and official inference.** The LF run uses the original 389-font training pool (the common baseline list of Appendix A), five validation fonts and 1,100 training characters, with bearing-corrected rendering and the distributed-training wrapper correction. Phase one reaches the saved iteration label 400,000 and phase two reaches 200,000, at global batch eight. The vendor's inclusive iteration convention gives 400,001 and 200,001 phase updates; phase two initializes from the completed phase-one checkpoint. All reported LF outputs use this final checkpoint and the vendor's original component selection at inference, without the target-component intervention. The recorded phase-two checkpoint SHA-256 begins `ce7a419115c69286`; the full hash and execution receipts accompany the source package.

The frozen two-reference panel has 124 outputs: 62 targets under correct and substituted references. The 50 evaluation targets and 12 supervised diagnostics remain separate. Native $128^2$ predictions are scored on the common $256^2$ canvas. This completed retraining replaces the earlier checkpoint and bounded renderer-continuation comparison throughout the paper; it does not isolate the causal effect of either implementation correction. No larger evaluation set or native 44/45-reference LF result is claimed for this checkpoint. The two fixed starred qualitative requests are additionally inferred with the same weights and official equations on CPU, and are excluded from the quantitative panel. A

repeated panel request differs from GPU inference by 0.0173 gray levels on average (maximum 12 on the 0–255 scale).

**AGD encoders.** Both arms initialize from the original step-800,000 EMA on the old 391-font, C800 data. They use fresh AdamW at $10^{-5}$, global batch 32, four ranks per arm, and 1,000 total updates. The frozen arm keeps encoder parameters and batch-normalization state fixed; the joint arm updates both. All 4,000 paired rank-steps match data draws, diffusion timesteps and noise. Frozen encoder state is unchanged, joint gradients are nonzero, and terminal parameters agree across ranks; parameters and normalization-state effects are not separated by this treatment.

The historical run behind this checkpoint trained both encoders jointly from random initialization (Appendix C.1); the official entry point instead marks the two encoder weights named on its command line as not requiring gradients, and neither file is publicly released, so its training command cannot be run as published. The historical training contract draws four same-font references with replacement from the training renderer, which is the count and the rule the released sampler uses.

The diagnostic rerun uses the verified training-style public content image and original K2 references. Its initial-old-EMA scores belong to this protocol, not to the two-reference record. All four pre-specified raw/EMA endpoints are reported in Table 6, with 248 new outputs plus 62 initial-control outputs. The 0.9999 EMA retains approximately 90.48% of its initialization after 1,000 updates; joint normalization buffers use current rank-zero state. The metric changes are small and mixed.

Two contract mismatches between this checkpoint's training data and the main table's ground truth are measured rather than assumed: the historical data was rendered by a training-style renderer at $128^2$ while the frozen ground truth is rendered under the evaluation contract at $256^2$, and switching one render domain at a time moves the comparison in opposite directions that a single metric does not order. Both directions, and the nested reference-count intervention, are reported with their paired table in Appendix B.2.

Table 6: AGD old-C800 diagnostic: 1,000 updates, frozen versus jointly trained encoders, both raw and EMA weights. Evaluation rows use the 50 evaluation targets and diagnostic rows the 12 supervised diagnostics. Initial old EMA is rerun under this diagnostic content-image protocol. None of these rows is used in Table 1, and none replaces the two-reference record.

| Pool | Configuration | SSIM | L1 | LPIPS | IoU |
|---|---|---|---|---|---|
| Evaluation (50) | initial old ema | 0.5729 | 0.2074 | 0.5364 | 0.2846 |
| Evaluation (50) | frozen raw | 0.5765 | 0.2058 | 0.5352 | 0.2860 |
| Evaluation (50) | frozen ema | 0.5729 | 0.2072 | 0.5370 | 0.2846 |
| Evaluation (50) | joint raw | 0.5764 | 0.2061 | 0.5382 | 0.2862 |
| Evaluation (50) | joint ema | 0.5785 | 0.2042 | 0.5340 | 0.2906 |
| Diagnostics (12) | initial old ema | 0.6162 | 0.1810 | 0.4791 | 0.3283 |
| Diagnostics (12) | frozen raw | 0.6114 | 0.1843 | 0.4816 | 0.3243 |
| Diagnostics (12) | frozen ema | 0.6158 | 0.1814 | 0.4795 | 0.3276 |
| Diagnostics (12) | joint raw | 0.6110 | 0.1850 | 0.4823 | 0.3212 |
| Diagnostics (12) | joint ema | 0.6141 | 0.1826 | 0.4775 | 0.3252 |

### B.2 Inference with native baseline settings

Our two-reference protocol changes some baseline inference interfaces. We therefore add a separate inference-setting supplement on all 62 frozen correct-reference requests, retaining the 50 evaluation targets and 12 supervised diagnostics as separate panels. The non-LF checkpoint identities and their K2 records remain unchanged. LF uses the retrained checkpoint in the main two-reference panel, has no native-setting arm here, and its two-reference row is shown for reference. We use the same original $256^2$ GT mapping and evaluator as the main results.

**What is official, and what remains adapted.** FontDiffuser's released sampler accepts one style image, so we run each original reference separately and average their per-target scores rather than selecting the better output. The released weights, 20-step DPM-Solver++ and guidance 7.5 remain fixed. The author-maintained FFG implementation for MX-Font supplies a 45-character Chinese

reference list; we remove frozen same-font query characters, giving 44 or 45 references. MX-Font++ consumes all images in its supplied reference folder; we provide four (its validation default), use its native source-font renderer, and apply its released `refine` and save operations. Raw pre-refinement outputs are retained. We also repeat the native-setting tests for both released and locally retrained MX-Font weights and for locally retrained FontDiffuser weights. MX native rendering uses the released `getsize` semantics, including the vertical offset, through a Pillow compatibility wrapper.

AGDFont's released sampler averages four independently encoded references using DDPM with CFG 2, the same four-reference count as its training. The native-setting arm draws four times with replacement from the old 800-character reference pool, excluding frozen same-font query characters, and adds deterministic per-request seeds. Public content and style glyphs use the verified local training renderer. This arm supplies AGDFont's rows in Tables 1 and 9; the two-reference record for the same checkpoint is kept below for comparison. The old EMA checkpoint is still the jointly trained, randomly initialized-encoder adaptation described in the baseline audit.

Table 7: Inference-setting supplement. LF uses the retrained checkpoint with official inference; other checkpoint identities are unchanged. Evaluation (50 targets) and supervised diagnostics (12 targets) are separate. Unequal reference budgets and native-rendering changes make these system-level comparisons, not a same-$K$ ranking.

| | | Evaluation (50) | | Diagnostic (12) | |
|---|---|---|---|---|---|
| Configuration | $K$ | LPIPS↓ | IoU↑ | LPIPS↓ | IoU↑ |
| FontDiffuser, two-reference protocol | 2 | 0.4216 | 0.3812 | 0.4285 | 0.3258 |
| FontDiffuser, native single-reference | 1 | 0.4253 | 0.3771 | 0.4333 | 0.3238 |
| MX-Font++, two-reference protocol | 2 | 0.3197 | 0.5197 | 0.3374 | 0.3839 |
| MX-Font++, native settings | 4 | 0.4538 | 0.2868 | 0.4455 | 0.2784 |
| LF-Font, retrained / official inference | 2 | 0.3246 | 0.4859 | 0.3420 | 0.4134 |
| AGDFont, two-reference protocol | 2 | 0.5592 | 0.2472 | 0.5379 | 0.2469 |
| AGDFont, native settings | 4 | 0.6341 | 0.2203 | 0.5680 | 0.2429 |
| MX-Font (released), two-reference | 2 | 0.3263 | 0.4997 | 0.3278 | 0.4255 |
| MX-Font (released), native | 44/45 | 0.4836 | 0.2619 | 0.4660 | 0.2789 |
| MX-Font (retrained), two-reference | 2 | 0.3402 | 0.4743 | 0.3554 | 0.3780 |
| MX-Font (retrained), native | 44/45 | 0.4499 | 0.3092 | 0.4378 | 0.2912 |
| FontDiffuser (retrained), two-reference | 2 | 0.4532 | 0.3507 | 0.4576 | 0.2648 |
| FontDiffuser (retrained), native | 1 | 0.4522 | 0.3402 | 0.4593 | 0.2681 |

**AGDFont reference-count diagnosis.** A nested intervention preserves both original references and adds two fixed same-font support images, excluding all frozen queries for that font. At fixed evaluation content, increasing $K$ from two to four changes evaluation IoU from 0.2472 to 0.2509 while LPIPS changes from 0.5592 to 0.5597. Changing only the public content renderer at $K = 2$ produces IoU 0.2846 and LPIPS 0.5363. This fixed-panel experiment therefore does not identify reference count as the main explanation of the observed failure. Rerendering the style references while holding content, reference characters, $K$ and noise fixed lowers evaluation-pool IoU from 0.2889 to 0.2187, and the official draw rule gives 0.2203, because those arms reproduce the training renderer's glyph proportion rather than the evaluation contract's: their generated ink covers 1.66 times the ground-truth ink. Rescaling each output to its own ink box before comparison, again holding everything else fixed, reverses that ordering and raises shape agreement from 0.3610 to 0.4220. We therefore report both readings.

**Numerical and visual verification.** The retained non-LF supplement contains 868 final predictions: 372 AGD, 248 FontDiffuser across its two checkpoints, and 124 each for MX-Font++ and MX-Font, including compatibility-renderer diagnostics for MX-Font++. The retrained LF checkpoint has only the two-reference evaluation in this record; native 44/45-reference LF results from the replaced checkpoint are excluded. Every new arm covers all 62 requests. The AGD arms have matching initial noise and checkpoint identity. A full 1,000-step single-request regression against the released per-reference encoding and sampler equations is bitwise exact. The historical adapter

Table 8: AGDFont reference-count and renderer diagnosis. All reruns share the old EMA checkpoint, 1,000-step DDPM, CFG 2, and paired per-request noise. E denotes frozen evaluation content; T denotes the verified training-style public content renderer. "Native refs" additionally rerenders the reference glyphs. AGDFont is trained with four same-font references, so the $K = 2$ and nested-reference rows above the last two are both off its training contract; the last row is the configuration used for AGDFont's rows in Table 1.

| | | Evaluation (50) | | Diagnostic (12) | |
|---|---|---|---|---|---|
| Content / reference setting | $K$ | LPIPS↓ | IoU↑ | LPIPS↓ | IoU↑ |
| E / original references | 2 | 0.5592 | 0.2472 | 0.5379 | 0.2469 |
| E / nested references | 4 | 0.5597 | 0.2509 | 0.5376 | 0.2463 |
| T / original references | 2 | 0.5363 | 0.2846 | 0.4791 | 0.3283 |
| T / nested references | 4 | 0.5330 | 0.2889 | 0.4798 | 0.3295 |
| T / native refs, same characters | 4 | 0.6315 | 0.2187 | 0.5703 | 0.2431 |
| T / native refs, official draw rule | 4 | 0.6341 | 0.2203 | 0.5680 | 0.2429 |

encoded references together. Native-size PNGs, all actual references, public content inputs, and failed attempts are retained with the experiment artifacts.

## C Generation Evidence

The following table reports the 62-request panel, with evaluation and supervised diagnostic targets reported separately. Table 2 reports the same configurations split into the UFUC, SFUC and SFSC cells. Pooled 62-request values are the weighted combination of those two columns; ten targets absent from both recorded adaptation lists (Appendix A) are inside the evaluation pool. Reference substitution is evaluated separately in Appendix C.4.

Table 9: Font generation with two same-font references. Evaluation-pool and supervised diagnostic targets are reported separately and are not pooled into a single column; membership in the evaluation pool does not by itself establish unseen-font exposure, and Table 2 gives the UFUC, SFUC and SFSC breakdown. All metrics use a common $256^2$ canvas and grayscale SSIM/L1. Bold marks the best point estimate among the generative configurations in LPIPS and IoU; SSIM carries no best marker because a blank image attains a high value at zero IoU.

| | Evaluation pool ($n = 50$) | | | | Diagnostic ($n = 12$) | | | |
|---|---|---|---|---|---|---|---|---|
| Method | LPIPS↓ | IoU↑ | SSIM↑ | L1↓ | LPIPS↓ | IoU↑ | SSIM↑ | L1↓ |
| Ours | **0.1925** | **0.5490** | **0.7368** | **0.1009** | **0.2421** | **0.4408** | **0.7224** | **0.1231** |
| Base editor | 0.6832 | 0.2107 | 0.2144 | 0.5201 | 0.6821 | 0.1920 | 0.2359 | 0.5213 |
| MX-Font (official) | 0.3263 | 0.4997 | 0.6984 | 0.1178 | 0.3278 | 0.4255 | 0.6964 | 0.1315 |
| MX-Font++ (retrained) | 0.3197 | 0.5197 | 0.7132 | 0.1081 | 0.3374 | 0.3839 | 0.6863 | 0.1415 |
| MX-Font (retrained) | 0.3402 | 0.4743 | 0.6879 | 0.1258 | 0.3554 | 0.3780 | 0.6789 | 0.1467 |
| FontDiffuser (official) | 0.4216 | 0.3812 | 0.6586 | 0.1473 | 0.4285 | 0.3258 | 0.6712 | 0.1516 |
| FontDiffuser (retrained) | 0.4532 | 0.3507 | 0.6533 | 0.1529 | 0.4576 | 0.2648 | 0.6582 | 0.1650 |
| LF-Font (official inference) | 0.3246 | 0.4859 | 0.7053 | 0.1171 | 0.3420 | 0.4134 | 0.6944 | 0.1389 |
| AGDFont (local subst., native $K$=4)[‡] | 0.6341 | 0.2203 | 0.4400 | 0.2942 | 0.5680 | 0.2429 | 0.4904 | 0.2669 |
| Content copy | 0.3222 | 0.4250 | 0.6972 | 0.1365 | 0.3593 | 0.3086 | 0.6589 | 0.1690 |
| Blank image | 0.6757 | 0.0000 | 0.6989 | 0.1774 | 0.6513 | 0.0000 | 0.7256 | 0.1618 |

Official and retrained configurations differ in exposure and training resources. SLD is excluded from the current comparison. The two panel columns combine to pooled 62-request values; pooling them over the blank-image row gives SSIM 0.7041 at zero IoU. LF-Font uses our retrained checkpoint with the upstream inference rule; AGDFont uses the local substitute checkpoint described in Appendix C.1, where reproduction limitations are discussed. [‡]AGDFont is shown in its native inference configuration, four references drawn with replacement plus training-style content and reference renders, because both its training and its released sampler use four references; the two-reference record for the same checkpoint, and with the evaluation-pool content and references, is retained in Table 7.

### C.1 RETRAINED LF-FONT WITH OFFICIAL INFERENCE AND LOCAL AGDFONT SUBSTITUTE

The main table and all qualitative comparisons use the completed LF 400k/200k phase-two checkpoint with official inference and the locally trained AGD EMA step-800,000 substitute described below. LF uses the original training pool and no expanded training data (Appendix B). Native outputs are $128^2$ and $80^2$, respectively, evaluated on the common $256^2$ canvas with the frozen GT mapping, which leaves the native-resolution difference in place. The AGDFont row of the main table is that substitute checkpoint in its native inference configuration: four references drawn with replacement, as in the released sampler and in its own training contract, with training-style content and reference renders.

These are the evaluated local implementations, with the inputs in Table 4; every row except AGDFont is evaluated with two references, as in the main generation panel. LF training and inference follow the corrections and vendor component selection recorded in Appendix B. AGD training-data pairing, encoder identity, and training/inference consistency remain reproduction-audit questions. The AGD content and style encoders were randomly initialized and trained jointly, whereas the official training entry point loads two encoder weight files named on its command line and freezes their parameters; those files are not publicly released. Comparing the step-0 and step-800,000 EMA weights of this run shows that all 20 content-encoder and all 34 style-encoder parameter tensors changed, so this checkpoint cannot be a frozen-encoder product of the official recipe. The editor comparison supports the adaptation claim independently of their relative performance.

### C.2 ADDITIONAL QUALITATIVE COMPARISONS

Figure 6 expands the dedicated-baseline comparison to eight targets and includes additional checkpoint configurations; the final two columns are supervised diagnostics. Figure 7 shows the base-editor comparison. All panels preserve original canvases and grayscale values. The comparison shows the content template and both actual style references.

### C.3 COMPLETE MULTI-CHARACTER PILOT

The following grids retain every target and both seeds for all six families.

Table 10: Six-family, six-character pilot. Means weight families equally; each family has six targets and two seeds (72 requests per method). MX-Font++ is deterministic across the two seeds; the 36 duplicate pairs do not add independent evidence. The generation anchor's ordinary reference-conditioned path is used, not the residual's weight request.

| Method | LPIPS ↓ | IoU ↑ | SSIM ↑ | L1 ↓ |
|---|---|---|---|---|
| Base editor | 0.6521 | 0.2637 | 0.3308 | 0.4371 |
| MX-Font++ | 0.3747 | 0.5006 | 0.6632 | 0.1447 |
| Ours | 0.2588 | 0.5406 | 0.7165 | 0.1239 |




Figure 6: **Additional font-generation configurations.** Columns 1–6 retain the earlier six-target evaluation-pool selection; columns 7–8 (†) are supervised diagnostics in the IPix and Bailu fonts. Rows marked off. use official checkpoints and ret. use retrained checkpoints; the AGDFont row is a local substitute checkpoint in its native four-reference configuration, not a retrained-for-this-comparison checkpoint, and its native content and four references appear per request in Appendix B.2 rather than in the rows above. Parentheses give native raster resolution. Content templates are inputs to dedicated baselines only. References, predictions, and ground truth retain their full original canvases and grayscale values. Selection is illustrative and includes failed predictions.

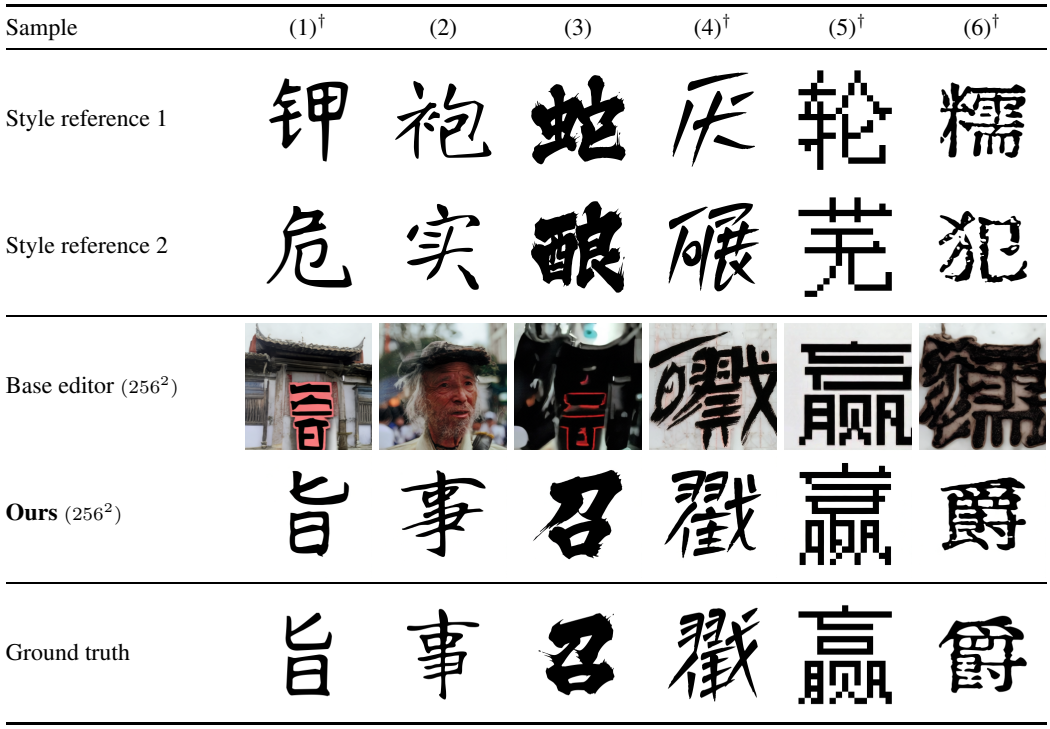



Figure 7: **Effect of font-task adaptation.** The base editor uses the same prompt, ordered references, visual-language/VAE paths, initial noise, sampler, and resolution as our model, without task-specific weights. Columns 2–3 originate from the evaluation pool; columns 1 and 4–6 (†) are supervised diagnostics. All examples retain their original $256^2$ canvases and grayscale values. Failures remain visible. This illustrative panel is not a random sample.

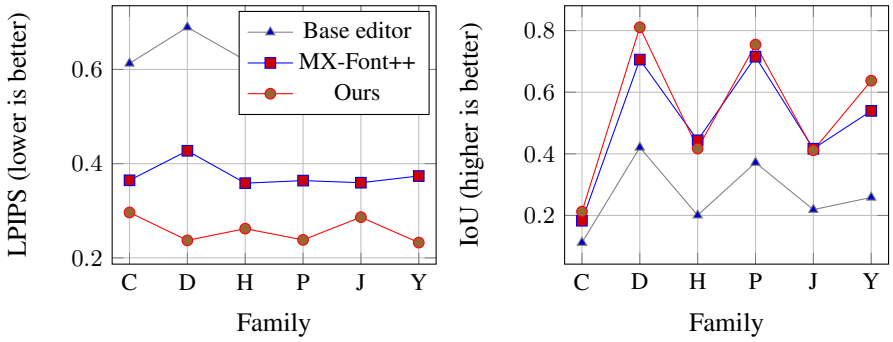



Figure 8: Every family in the multi-character pilot; points average six characters and two seeds. C: ChenYuluoyan, D: DelaGothicOne, H: HachiMaruPop, P: PottaOne, J: YujiSyuku, Y: YuseiMagic. Ours improves LPIPS over both comparators for all six families, but its IoU is below MX-Font++ for H and J. Lines connect categories only.

The six targets are U+81E3, U+53D6, U+6EA2, U+5A7F, U+7CE0 and U+58A8, with 6, 8, 13, 12, 17 and 15 strokes respectively, and the fixed reference characters are U+57A2 and U+8306. The font/target set was frozen before outputs, and no family was removed. The image-like base-editor output for ChenYuluoyan/U+53D6 under the first sampling seed is retained after verifying its prompt, reference binding and initial noise.

All 216 outputs and 282 delivery files were verified. The original editor pair shares 72 initial-noise identities. MX-Font++ has native $128^2$ output and is scored after the declared common-canvas resize; the two editor arms have native $256^2$ output. Available training-list audits cover 181/397-font

original generation records and the MX 389+5 records; coverage of historical binaries is partial. DelaGothicOne's later merged-model lineage association is with BiaoXiaoZhiWuJieHei (Unbounded Sans); YuseiMagic's is through Tanugo and XWMH. These flags concern later adaptation records and do not identify the present Seed1 generation anchor as a merged model. Undocumented lineage and foundation exposure remain unresolved.

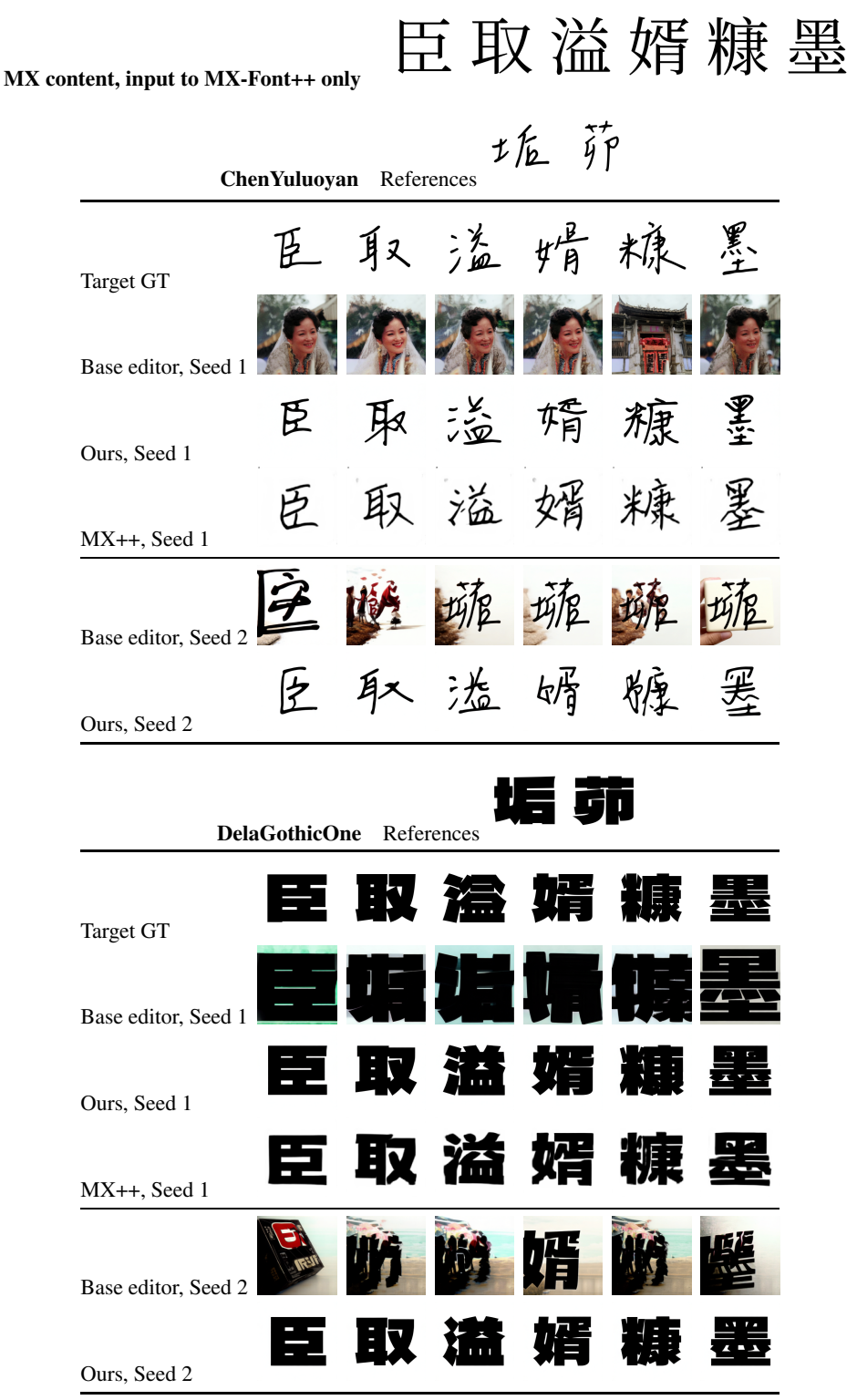



Figure 9: ChenYuluoyan and DelaGothicOne: complete six-character panels at both sampling seeds. Actual reference bitmaps appear beside each family name; target GT is evaluation-only. MX-Font++ is deterministic across sampling seeds, so only its first-seed row is shown; the MX content row is identical for all six families and appears at the top of each panel. Original full canvases and failures are retained.

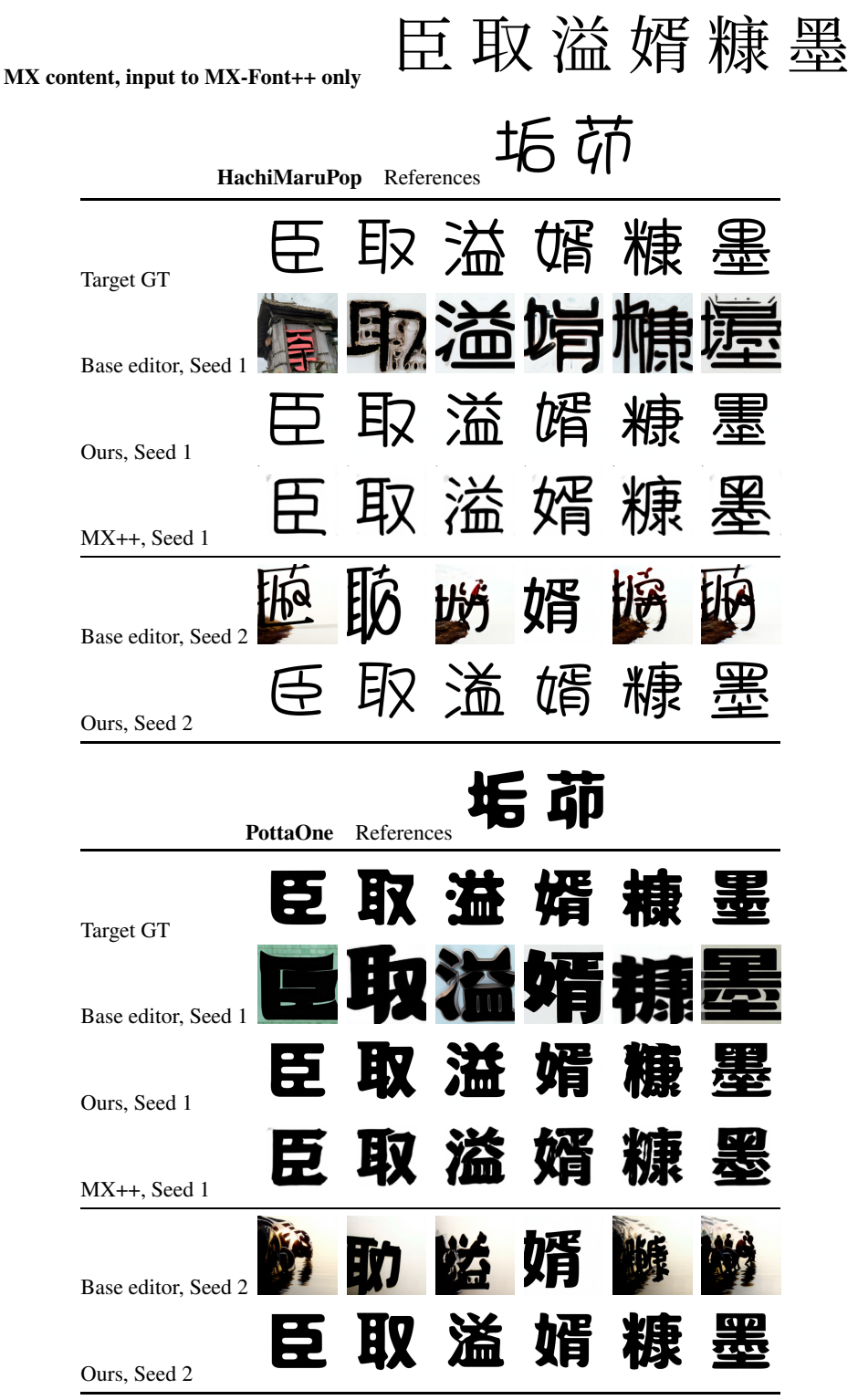



Figure 10: HachiMaruPop and PottaOne: complete six-character panels at both sampling seeds. Actual reference bitmaps appear beside each family name; target GT is evaluation-only. MX-Font++ is deterministic across sampling seeds, so only its first-seed row is shown; the MX content row is identical for all six families and appears at the top of each panel. Original full canvases and failures are retained.




Figure 11: YujiSyuku and YuseiMagic: complete six-character panels at both sampling seeds. Actual reference bitmaps appear beside each family name; target GT is evaluation-only. MX-Font++ is deterministic across sampling seeds, so only its first-seed row is shown; the MX content row is identical for all six families and appears at the top of each panel. Original full canvases and failures are retained.

### C.4 PAIRED REFERENCE-FONT SUBSTITUTION

This diagnostic reuses the same generation anchor under the original generation requests, together with the 50 evaluation targets from the main comparison; the 12 supervised diagnostics are excluded. The historical launch record binds the export to the same task LoRA, and the frozen input manifest verifies both ordered reference characters and image hashes. Each original-reference output and replacement-reference output is scored against both renderings of the requested character. Let $D$ be LPIPS and define $m(y) = D(y, x_{\text{original}}) - D(y, x_{\text{substituted}})$. The paired response is $m(y_{\text{substituted}}) - m(y_{\text{original}})$, so positive values indicate movement toward the substituted face relative to the original output.

Table 11: Paired reference substitution for the generation anchor, the same weights as every other generation panel. Each of 50 targets keeps the requested character, two reference characters, prompt and inference seed fixed while the reference font changes. The four LPIPS columns score both outputs against both font GTs. A positive paired response means that substitution increases relative proximity to the substituted face; it does not measure blinded style fidelity. Native images are retained.

| | Original references | | Substituted references | | Positive |
|---|---|---|---|---|---|
| Method | Original GT | New GT | Original GT | New GT | response |
| Base editor | 0.6832 | 0.6938 | 0.6992 | 0.6631 | 42/50 |
| Ours | 0.1925 | 0.3900 | 0.3911 | 0.1584 | 50/50 |
| MX-Font++ | 0.3197 | 0.4518 | 0.4508 | 0.3082 | 48/50 |

Our model has positive paired response on 50/50 targets, compared with 42/50 for the base editor and 48/50 for MX-Font++. Its mean response is 0.4302, versus 0.0468 and 0.2748. After substitution, 48/50 of our outputs are closer to the substituted GT than the original GT; LPIPS to the substituted GT is 0.1584 and IoU is 0.6081. These are paired metric observations on a fixed panel. In particular, comparing 0.1584 with the correct-reference score of 0.1925 would mix different target fonts. The two diagonal score sets reproduce the existing per-request records within $5 \times 10^{-7}$.

## D CHARACTER IDENTITY

Character identity is checked against the full original output, target GT, and actual reference glyphs, since pixel similarity alone does not determine whether a prediction is the requested character. A frozen PP-OCRv4 recognizer was used for exploratory screening, but it also misreads ground-truth glyphs from pinyin-annotated and running-script faces, so we do not report its accuracy as a comparative result. The screening recipe and raw outputs remain in the evidence archive.

The failure record retains visually confusable characters (including U+7CE0/U+7CD6), a blank pilot output, and omitted pinyin annotation, shown alongside successful outputs in Appendix G. Character identity and following the reference font are separate properties, and final judgments rest on inspection of the original images.

## E INTERFACE AND REFERENCE SENSITIVITY

The current studies change inputs to the same generation anchor. Section 4.4 gives the compact target-equal comparison; this appendix adds the metrics, raw examples, font-equal estimates, and conditioning costs behind it.

### E.1 Reference count, uncertainty, and cost

Table 12: Current native-joint reference-count ablation. Each row averages 50 targets after averaging two reference sets and two sampling seeds per target (200 predictions per $K$; 800 total). All use the same $K = 2$-trained checkpoint. FG-SSIM measures SSIM within the GT foreground bounding box (threshold 0.5).

| $K$ | SSIM↑ | FG-SSIM↑ | IoU↑ | LPIPS↓ |
|---|---|---|---|---|
| 1 | 0.7332 | 0.4702 | 0.5275 | 0.2081 |
| 2 | **0.7418** | **0.4869** | **0.5477** | **0.1933** |
| 4 | 0.7104 | 0.4260 | 0.4867 | 0.2353 |
| 8 | 0.3140 | 0.1859 | 0.1799 | 0.6885 |




Figure 12: **Unmodified current K outputs.** The first two targets in frozen request order, using reference set r0 and the first sampling seed. Each $K$ consumes the first $K$ displayed references. GT is evaluation-only. Complete canvases and grayscale values are retained, including degraded outputs. These two examples illustrate behavior.

Figure 13 gives equal weight to 29 font keys. Repeated references and seeds are repeated measurements, not independent fonts; the targets reuse the controlled pool.

For $K = 1, 2, 4, 8$, measured positive embeddings contain 221/424/830/1,642 tokens and VAE references contain 4,096/8,192/16,384/32,768 tokens. Each reference uses $384^2$ visual-language and $1024^2$ VAE preprocessing. Mean request times are 28.4/66.7/181.8/567.6 seconds under eight concurrent workers. These are wall-clock times under that concurrency rather than single-request latency; peak memory and per-baseline inference cost are not reported. Despite the invoked maximum sequence length of 1,024, the measured K8 embedding has 1,642 tokens.

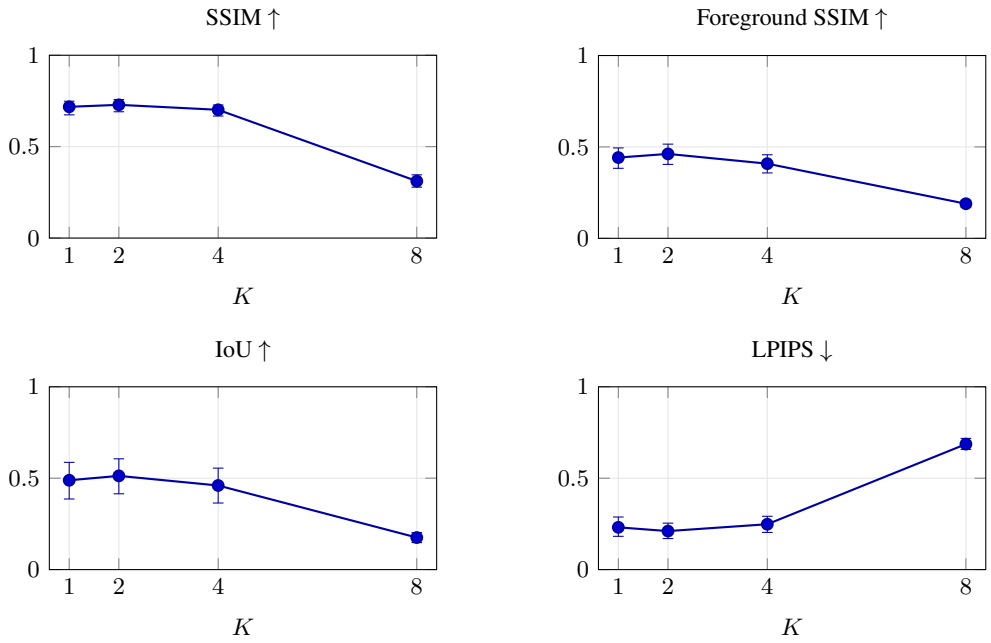



Figure 13: **Reference count under current native-joint inference.** Means give equal weight to each of the 29 font keys, unlike the target-equal means in Table 12. Error bars are exploratory 95% font-cluster percentile bootstrap intervals (10,000 resamples), averaging repeats within targets and targets within each font. The controlled panel is not a new independent test set. K8 is worse in every metric.

### E.2 Template effects and interaction uncertainty

Table 13: Completed content-template comparison. Each row averages 46 targets after averaging two sampling seeds per target (92 predictions per row; 368 total). The two style references and role-aware instruction are fixed; the public Song template is appended only in the template-on condition. These are inference-time interventions on the existing checkpoints.

| Model | Template | SSIM↑ | FG-SSIM↑ | IoU↑ | LPIPS↓ |
|---|---|---|---|---|---|
| Base editor | Off | 0.3280 | 0.1518 | 0.1958 | 0.6394 |
| Base editor | On | 0.2453 | 0.1289 | 0.1792 | 0.7228 |
| Ours | Off | **0.7446** | **0.4931** | **0.5552** | **0.1983** |
| Ours | On | 0.6852 | 0.3825 | 0.4144 | 0.3065 |

Across 27 font clusters, exploratory 95% intervals for our model's font-equal template effect are $[-0.0778, -0.0337]$ (SSIM), $[-0.1997, -0.0840]$ (IoU), and $[0.0952, 0.1315]$ (LPIPS); these differ from the table's target-equal estimand. Interaction intervals cross zero for SSIM/LPIPS but are negative for IoU/FG-SSIM, indicating a larger template penalty after adaptation only on those metrics. Raw failures remain in Figure 14.




Figure 14: **Content-template intervention on the font editor.** The first two eligible targets in frozen request order, each with the first sampling seed. GT is evaluation-only; the same two other-character references are used in all four cells, and only template-on receives the public target-character image. Original full canvases and grayscale are retained, including non-glyph outputs. These examples illustrate the frozen protocol.

# F WEIGHT-CONTROL EVIDENCE AND CONSTRAINT PROTOCOL

This appendix supplies the evidence for the response and generation-retention questions in Section 5. The unconstrained panels compare request channels. Section F.5 gives the full constrained results, zero-displacement checks, complex-font diagnostics, and baseline comparisons.

## F.1 FAMILIES HELD OUT FROM THE WEIGHT-CONDITIONING STREAM AND PAIRED GATES

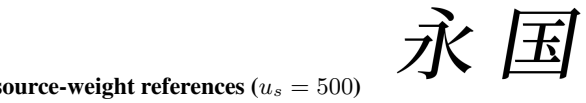
The full held-out-family-panel direction/seed summaries are in Tables 14 and 15. Figure 15 shows the fixed-reference weight sequence over both sampling seeds with the unadapted editor, and Figure 17 the frozen raw example for both displacement parameterizations.

Table 14: Completed held-out-family comparison with the base editor. Each method has 170 target-instance rows and two seeds (340 predictions): three families held out from the weight-conditioning stream and ten held-out characters. Target axes are averaged within family–character cells, then characters within each family, then the three families equally. Ink error is absolute foreground-area-fraction error, not weight-axis error; ink and IoU use gray $< 200/255$.

| | | SSIM↑ | | IoU↑ | | Ink error↓ | |
|---|---|---|---|---|---|---|---|
| Direction | Method | Seed 1 | Seed 2 | Seed 1 | Seed 2 | Seed 1 | Seed 2 |
| Lighter (90) | Base editor | 0.2672 | 0.4672 | 0.2091 | 0.1856 | 0.4527 | 0.1523 |
| | Ours, prompt-only | 0.6903 | 0.6830 | 0.5137 | 0.4836 | 0.0669 | 0.0584 |
| | Ours, numeric | 0.6942 | 0.7054 | 0.5252 | 0.4890 | 0.0756 | 0.0400 |
| | Ours, learned gate | 0.7344 | 0.7063 | 0.5495 | 0.4628 | 0.0188 | 0.0123 |
| Source (30) | Base editor | 0.2579 | 0.4467 | 0.2623 | 0.2283 | 0.4167 | 0.1100 |
| | Ours, prompt-only | 0.7193 | 0.7162 | 0.6189 | 0.5946 | 0.0145 | 0.0110 |
| | Ours, numeric | 0.7252 | 0.7220 | 0.6325 | 0.5735 | 0.0309 | 0.0282 |
| | Ours, learned gate | 0.7225 | 0.7160 | 0.6206 | 0.5891 | 0.0135 | 0.0090 |
| Heavier (50) | Base editor | 0.2546 | 0.4332 | 0.3005 | 0.2549 | 0.3764 | 0.0831 |
| | Ours, prompt-only | 0.7113 | 0.7114 | 0.6321 | 0.6161 | 0.0293 | 0.0368 |
| | Ours, numeric | 0.7148 | 0.6957 | 0.6420 | 0.5623 | 0.0295 | 0.0621 |
| | Ours, learned gate | 0.7163 | 0.7230 | 0.6609 | 0.6520 | 0.0107 | 0.0100 |

All Ours rows share the frozen generation anchor and compare request channels: *Ours, prompt-only* names the weight in the instruction with no explicit displacement, *Ours, numeric* is the numeric-conditioned request, and *Ours, learned gate* is the displacement-conditioned request, which receives nonnumeric character text plus an explicit displacement.

**Fixed source-weight references** ($u_s = 500$)



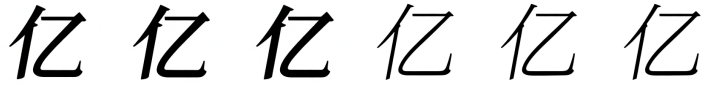
| | Seed 1 | | | Seed 2 | | |
|---|---|---|---|---|---|---|
| | $u_t = 200$ | $u_t = 500$ | $u_t = 900$ | $u_t = 200$ | $u_t = 500$ | $u_t = 900$ |
| GT |  | | | | | |
| Base editor: pretrained |  | | | | | |
| Ours, prompt-only |  | | | | | |
| Ours, numeric |  | | | | | |
| Ours, learned gate |  | | | | | |

Figure 15: **Fixed-reference weight sequences with the unadapted editor.** We retain exactly the earlier held-out-family selection: the first family and character in frozen row order (Sanmu Xiesong, U+4EBF), its minimum/source/maximum available target weights, and both seeds. The two other-character references are actual inputs; GT is evaluation-only. Figure 4a shows the Seed 1 GT and three adapted-interface rows from this complete panel. All 30 glyph cells are unedited original PNGs, including base-editor scene backgrounds and off-target outputs. The base editor, prompt-only and numeric requests use numeric text; the learned gate uses nonnumeric text plus an explicit displacement. These are complete configurations, not a pure adapter ablation.

Table 15: Paired gate comparison on the held-out-family panel (680 outputs). Both arms use the same frozen generation anchor and separately trained residuals, with the same nonnumeric prompt; *Linear* is the fixed linear gate and *Gated* the learned one. Means first average target weights within each family–character (and seeds for pooled rows), then characters within each family, then the three families equally. Ink error is absolute foreground-fraction error, and these families are held out from the weight-conditioning stream. The *Gated* column re-samples the same learned-gate checkpoint in the paired launch, so its values differ slightly from the *Ours, learned gate* rows of Table 14; only the paired differences are read from it.

| | | SSIM↑ | | IoU↑ | | Ink error↓ | |
|---|---|---|---|---|---|---|---|
| Direction | Seed | Linear | Gated | Linear | Gated | Linear | Gated |
| Light | Seed 1 | 0.7296 | 0.7339 | 0.5424 | 0.5480 | 0.0226 | 0.0195 |
| Light | Seed 2 | 0.7074 | 0.7089 | 0.4682 | 0.4681 | 0.0146 | 0.0130 |
| Light | Pooled | 0.7185 | 0.7214 | 0.5053 | 0.5081 | 0.0186 | 0.0163 |
| Identity | Seed 1 | 0.7226 | 0.7226 | 0.6217 | 0.6217 | 0.0136 | 0.0136 |
| Identity | Seed 2 | 0.7147 | 0.7147 | 0.5882 | 0.5882 | 0.0098 | 0.0098 |
| Identity | Pooled | 0.7187 | 0.7187 | 0.6050 | 0.6050 | 0.0117 | 0.0117 |
| Heavy | Seed 1 | 0.7206 | 0.7179 | 0.6658 | 0.6635 | 0.0099 | 0.0112 |
| Heavy | Seed 2 | 0.7209 | 0.7226 | 0.6481 | 0.6506 | 0.0098 | 0.0099 |
| Heavy | Pooled | 0.7207 | 0.7202 | 0.6569 | 0.6571 | 0.0099 | 0.0105 |

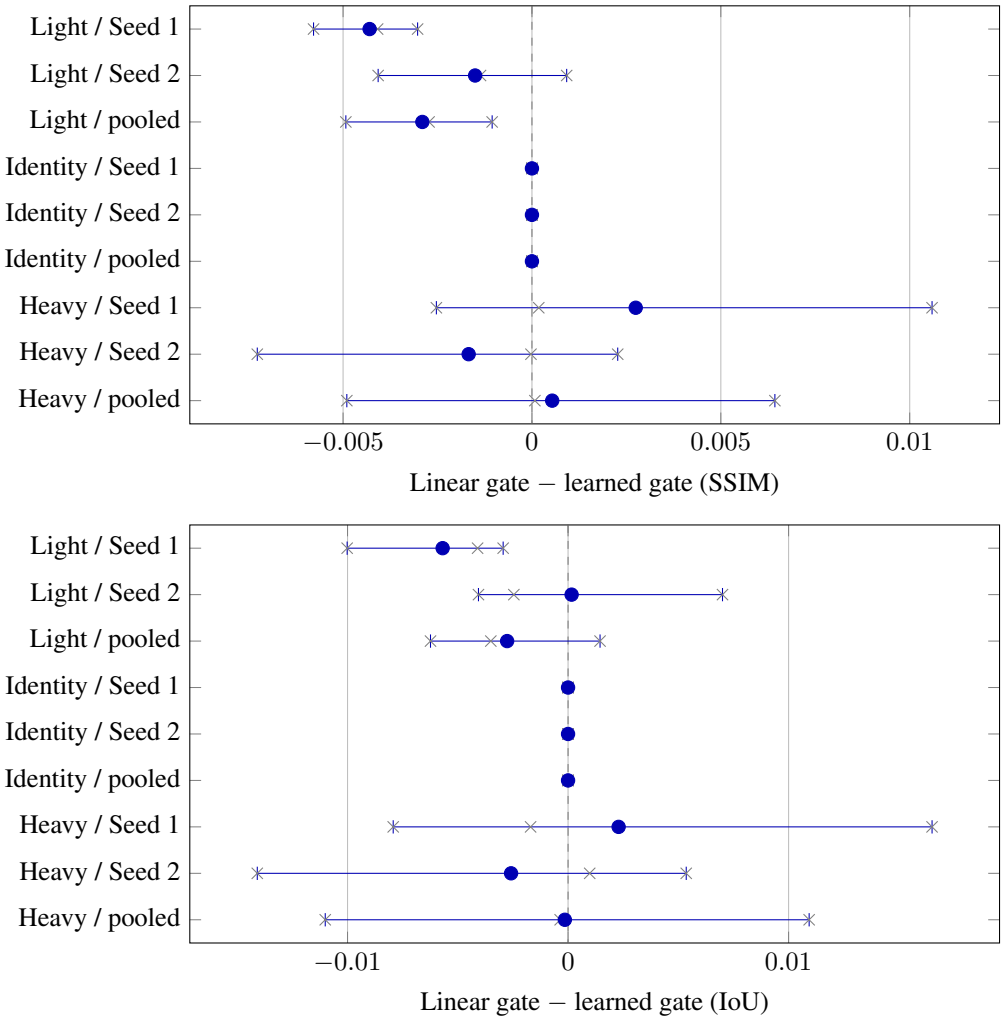



Figure 16: Direction- and seed-specific paired effects on the held-out-family panel. Dots show family-equal linear-minus-learned-gate means between separately trained gate variants on the same frozen generation anchor; crosses show all three family effects; bars are exploratory 95% family-bootstrap intervals. The two sampling seeds appear separately and pooled. Positive values favor the linear gate for these two fidelity metrics. The panel contains three family clusters.

**Fixed references** ($u_s = 500$) 

| | Seed 1 | | | Seed 2 | | |
|---|---|---|---|---|---|---|
| | $u_t = 200$ | $u_t = 500$ | $u_t = 900$ | $u_t = 200$ | $u_t = 500$ | $u_t = 900$ |
| GT |  |  |  |  |  |  |
| Ours, linear gate |  |  |  |  |  |  |
| Ours, learned gate |  |  |  |  |  |  |

Figure 17: Separately trained linear- and learned-gate residuals on the same frozen generation anchor, on the same first family/character as Figure 15: Sanmu Xiesong, U+4EBF. Minimum, source and maximum target weights are shown for both sampling seeds. The two other-character reference images are the actual inputs; source-weight target GT is displayed only. All cells are unedited original PNGs. This deterministic selection is for traceability.

### F.2 Families seen by the weight-conditioning stream

The seen-family panel contains 1,003 target-instance requests in nine families seen by the weight-conditioning stream, with 17 held-out target characters. Each request is evaluated with the prompt-only request, the numeric-conditioned request, and the displacement-conditioned request with the learned gate, under two sampling seeds: 6,018 outputs in total. The frozen inference protocol uses $256^2$ canvases, 40 steps, guidance 1.0, and two ordered reference glyphs at source weight 500. Table 16 reports direction- and seed-specific family-equal means. Only the two reference images are image inputs; target GT, Source500 and public content are display/evaluation material.

Figures 18 and 19 resolve the table into its nine families; the SSIM family breakdown is summarized by the table's family-equal means rather than plotted. Family codes are F1: B2B2; F2: Taiwan Rounded; F3: Jiangcheng Xiesong; F4: Jiangcheng Xiehei; F5: Jiangcheng Zhengyi; F6: Shiwei B2 Sugared Song; F7: Shiwei B2 Song; F8: Shiwei Sugared Song; F9: Shiwei Meatball. Font lineages were not independently audited.

**Coverage.** All 6,018 predictions, target GT images and actual references are present and identity-verified. For U+545C, the two-seed audit preserves 118 missing public-content display cells; actual references and predictions are complete.

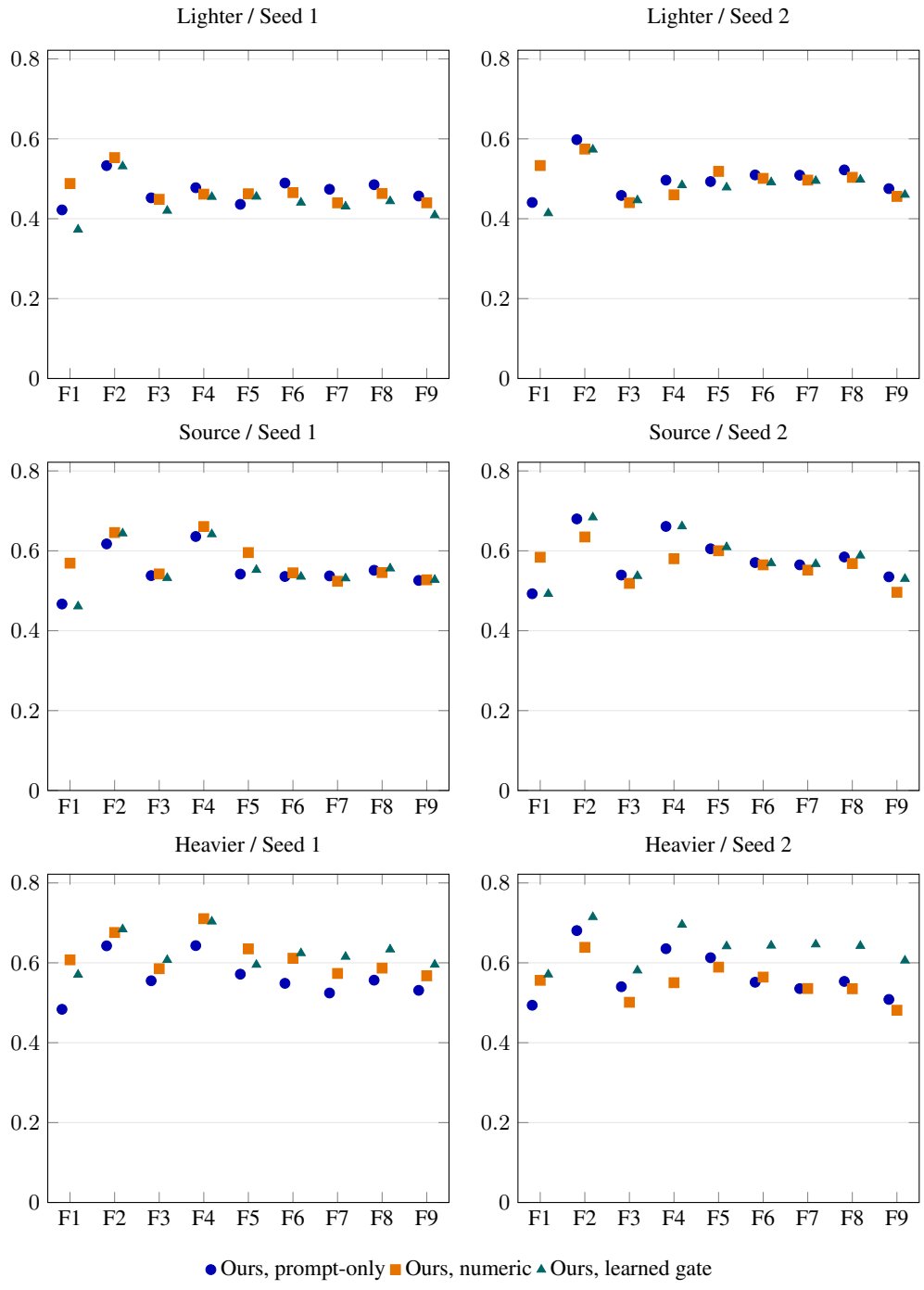



Figure 18: **Seen-family IoU ↑.** Each point averages all frozen target axes and held-out characters within one family, separately for direction and sampling seed. Family codes are defined in Section F.2. All nine families were seen in the weight-conditioning stream. All three rows share the frozen generation anchor and compare different request channels: the prompt-only and numeric requests name the weight in the instruction, while the learned gate takes nonnumeric text plus an explicit displacement. These are descriptive point estimates.

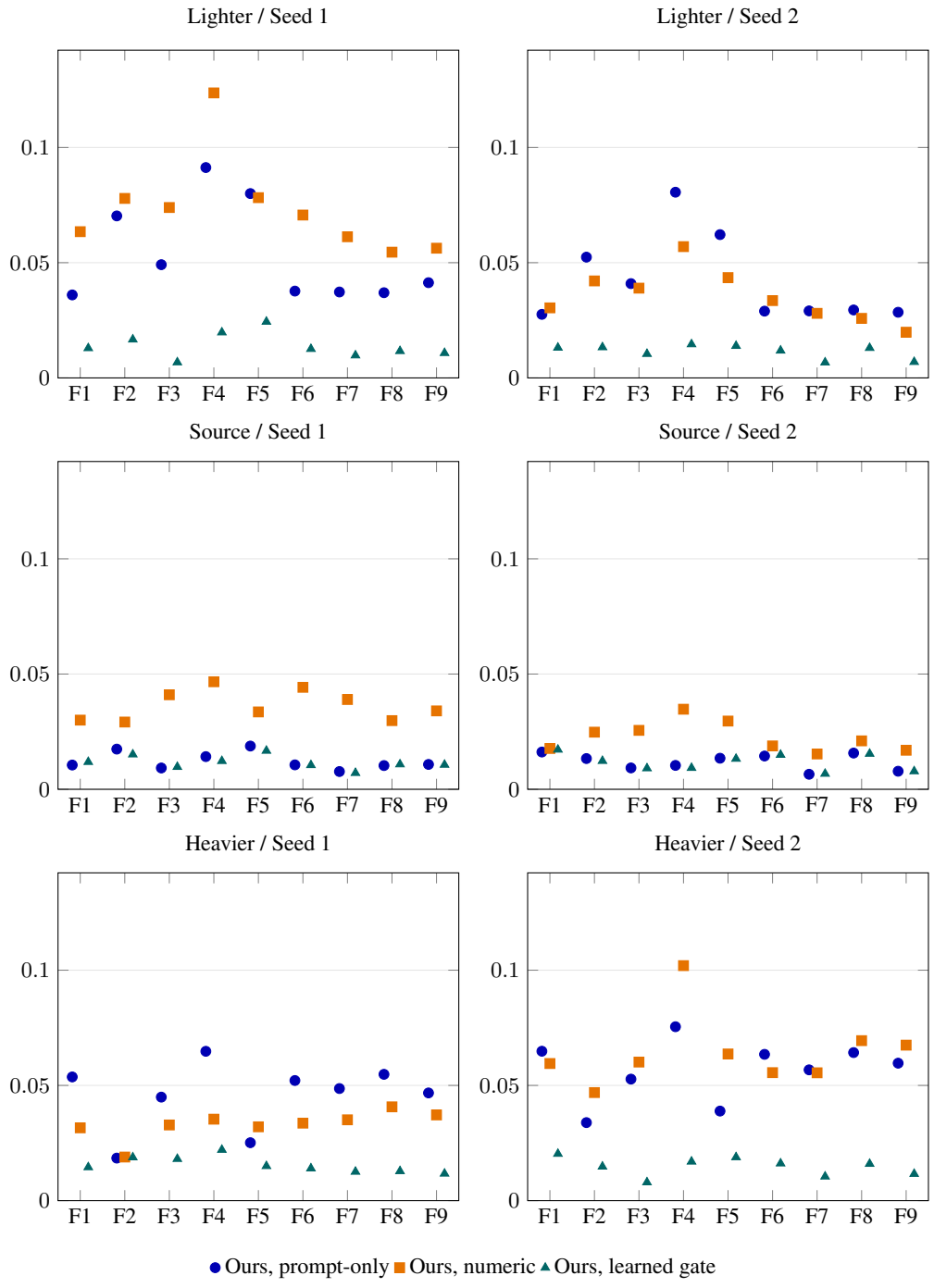



Figure 19: **Seen-family Absolute ink error ↓.** Each point averages all frozen target axes and held-out characters within one family, separately for direction and sampling seed. Family codes are defined in Section F.2. All nine families were seen in the weight-conditioning stream. All three rows share the frozen generation anchor and compare different request channels: the prompt-only and numeric requests name the weight in the instruction, while the learned gate takes nonnumeric text plus an explicit displacement. These are descriptive point estimates.

Table 16: Requested-weight results on the seen-family panel: nine families seen during the weight-conditioning stream and 17 held-out target characters. Each method has 1,003 target-instance rows per seed (6,018 outputs total). Means first average target axes within each family–character, then characters, then the nine families equally. Seed 1 and Seed 2 are the two sampling seeds; directions remain separate. The seen-family panel is not pooled with the held-out-family panel.

| | | SSIM↑ | | IoU↑ | | Ink error↓ | |
|---|---|---|---|---|---|---|---|
| Direction | Method | Seed 1 | Seed 2 | Seed 1 | Seed 2 | Seed 1 | Seed 2 |
| Lighter (493) | Ours, prompt-only | 0.7115 | 0.7307 | 0.4696 | 0.5003 | 0.0533 | 0.0422 |
| | Ours, numeric | 0.7025 | 0.7370 | 0.4691 | 0.4981 | 0.0733 | 0.0355 |
| | Ours, learned gate | 0.7272 | 0.7528 | 0.4398 | 0.4822 | 0.0139 | 0.0115 |
| Source (153) | Ours, prompt-only | 0.7204 | 0.7411 | 0.5500 | 0.5815 | 0.0122 | 0.0119 |
| | Ours, numeric | 0.7235 | 0.7373 | 0.5728 | 0.5665 | 0.0364 | 0.0227 |
| | Ours, learned gate | 0.7230 | 0.7415 | 0.5534 | 0.5820 | 0.0116 | 0.0118 |
| Heavier (357) | Ours, prompt-only | 0.7055 | 0.7147 | 0.5619 | 0.5679 | 0.0455 | 0.0567 |
| | Ours, numeric | 0.7269 | 0.7087 | 0.6170 | 0.5502 | 0.0330 | 0.0645 |
| | Ours, learned gate | 0.7200 | 0.7339 | 0.6254 | 0.6379 | 0.0155 | 0.0148 |

All Ours rows share the frozen generation anchor and compare request channels: *Ours, prompt-only* names the weight only in the instruction with no explicit displacement, *Ours, numeric* is the numeric-conditioned request, and *Ours, learned gate* is the displacement-conditioned request. The prompt-only and numeric requests receive numeric text; the learned gate receives nonnumeric text plus an explicit displacement. Ink error is absolute foreground-area-fraction error, not calibrated axis error; ink/IoU use gray $< 200/255$. Source-weight rows here do not by themselves establish the zero-displacement equality.

### F.3 Complete paired linear and learned gates for the seen-family panel

Table 17: Complete seen-family panel paired-gate study: 1,003 target instances, two gates, two seeds (4,012 outputs). Both arms use the same frozen generation anchor, with residuals trained separately under the two gate parameterizations; the *Linear* column uses the fixed linear gate and the *Gated* column the learned gate. Means average axes and seeds within family–character, then characters, then nine families equally. The *Gated* column re-samples the same learned-gate checkpoint in the paired launch, so its values differ slightly from the *Ours, learned gate* rows of Table 16; only the paired differences are read from it.

| Seed | Direction | SSIM Linear | SSIM Gated | IoU Linear | IoU Gated | Ink Linear | Ink Gated |
|---|---|---|---|---|---|---|---|
| Seed 1 | light | 0.7248 | 0.7266 | 0.4357 | 0.4383 | 0.0151 | 0.0141 |
| Seed 1 | identity | 0.7228 | 0.7228 | 0.5526 | 0.5526 | 0.0113 | 0.0113 |
| Seed 1 | heavy | 0.7225 | 0.7195 | 0.6289 | 0.6245 | 0.0148 | 0.0160 |
| Seed 2 | light | 0.7497 | 0.7527 | 0.4757 | 0.4820 | 0.0123 | 0.0117 |
| Seed 2 | identity | 0.7410 | 0.7410 | 0.5803 | 0.5803 | 0.0120 | 0.0120 |
| Seed 2 | heavy | 0.7361 | 0.7339 | 0.6424 | 0.6376 | 0.0150 | 0.0151 |
| Both | light | 0.7373 | 0.7397 | 0.4557 | 0.4602 | 0.0137 | 0.0129 |
| Both | identity | 0.7319 | 0.7319 | 0.5665 | 0.5665 | 0.0117 | 0.0117 |
| Both | heavy | 0.7293 | 0.7267 | 0.6357 | 0.6311 | 0.0149 | 0.0156 |

Table 18: Pooled-seed seen-family panel gate differences (learned gate minus linear gate). Both arms share the frozen generation anchor and use separately trained residuals. Exploratory 95% bootstrap intervals resample nine family clusters; they do not include retraining uncertainty.

| Direction | Metric | $\Delta$ | Family interval |
|---|---|---|---|
| light | SSIM | +0.00239 | [+0.00116, +0.00350] |
| light | IoU | +0.00444 | [+0.00141, +0.00790] |
| light | Ink error | -0.00073 | [-0.00130, -0.00028] |
| heavy | SSIM | -0.00259 | [-0.00348, -0.00149] |
| heavy | IoU | -0.00460 | [-0.00616, -0.00268] |
| heavy | Ink error | +0.00064 | [+0.00018, +0.00120] |

The 4,012-output paired-gate panel is distinct from the 6,018-output whole-interface panel above; each seed contributes 2,006 outputs. All 306 zero-displacement PNG and latent pairs are byte- and latent-exact to the generation adaptation the transfer panels evaluate. Public-content display material is absent in 1,888 paired audit rows; these are explicitly missing display cells, not generation failures or missing reference inputs. The anchor's generation training includes some exact targets from the seen-family panel, so these families are not whole-adaptation unseen.

### F.4 Ordinary-generation retention and scoring target

The 124-request retention panel consists of 62 correct-reference and 62 replacement-font-reference conditions. It reuses the fixed generation request pool and is a regression panel. Its 496 formal outputs are complete. The 248 displacement-conditioned formal outputs at $d = 0$ are byte- and latent-exact to the generation adaptation; all 124 four-arm initial-noise records match. Twelve learned-gate smoke checks cover zero input, zero displacement and separate-process reload for fixed correct and replacement examples.

Table 19: Ordinary nonnumeric generation, correct-reference subset (62 requests). All rows share the frozen generation anchor; request channels and trained control parameters differ. The displacement-conditioned requests use $d = 0$, which is the interface's zero-displacement path and returns the generation adaptation, while the numeric-conditioned request remains active. Exact equality is checked in both PNG bytes and final latent tensors. This subset includes evaluation and supervised diagnostic requests and is not pooled with the generation benchmark.

| Configuration | LPIPS ↓ | IoU ↑ | SSIM ↑ | Exact |
|---|---|---|---|---|
| Ours, prompt-only | 0.2524 | 0.4603 | 0.6874 | 62/62 |
| Ours, numeric | 0.2570 | 0.4669 | 0.6866 | 0/62 |
| Ours, linear gate ($d = 0$) | 0.2524 | 0.4603 | 0.6874 | 62/62 |
| Ours, learned gate ($d = 0$) | 0.2524 | 0.4603 | 0.6874 | 62/62 |

Every score in this panel uses the original target-font GT; the substituted face's own rendering appears only in the reference-swap diagnostic below. On the replacement subset, the prompt-only request, the linear-gate request and the learned-gate request give LPIPS, IoU, SSIM and L1 of 0.4000, 0.3358, 0.6340 and 0.1775; the numeric-conditioned request gives 0.3942, 0.3554, 0.6353 and 0.1767. On correct references the numeric-conditioned request's LPIPS is slightly worse while its IoU is higher (Table 19). The numeric-conditioned request thus changes ordinary outputs relative to the prompt-only request, with condition-dependent metric effects.

**Reference following against the substituted face.** A replacement request keeps the two reference characters and changes the face they come from, so the requested character is unchanged and only the implied font moves. Scoring those frozen outputs against the substituted face's own rendering of the requested character — same scorer, same convention, only the target image differs — shows which face the interface followed (Table 20). The prompt-only request, and both displacement-conditioned requests at $d = 0$ with it, moves from LPIPS 0.4000 against the original face to 0.2286 against the substituted one, with RMSE 0.4038 to 0.3339, and is closer to the substituted face on

55/62 requests by LPIPS and 48/62 by SSIM. The numeric-conditioned request is closer to the substituted face on 59/62 requests by LPIPS and 45/62 by SSIM. Its 0.2286 is not comparable to the correct-reference score of 0.2524, since those conditions have different target fonts. These observations support reference tracking on this panel. The paired diagnostic for the original generation requests is reported separately in Appendix C.4.

Table 20: Reference following under an ordinary generation protocol. The 62 replacement requests use the same reference characters as their correct-reference counterparts from a different face, so the substitution leaves the requested character unchanged and changes only the implied font. Both columns score the same outputs with the same metrics and convention; the left column uses the original target face, the right column the substituted face's own rendering of the requested character. All rows share the frozen generation anchor: *Ours, prompt-only* names the weight only in the instruction with no explicit displacement, *Ours, numeric* is the numeric-conditioned request, and the two gains are displacement-conditioned at $d = 0$; those $d = 0$ requests reproduce the generation adaptation byte- and latent-exactly, so those three rows are the generation line. Lower is better for LPIPS, L1 and RMSE.

| | Original target face | | | Substituted face | | |
|---|---|---|---|---|---|---|
| Configuration | LPIPS ↓ | SSIM ↑ | L1 ↓ | LPIPS ↓ | SSIM ↑ | L1 ↓ |
| Ours, prompt-only / linear / learned gate, $d = 0$ | 0.4000 | 0.6340 | 0.1775 | 0.2286 | 0.6961 | 0.1268 |
| Ours, numeric | 0.3942 | 0.6353 | 0.1767 | 0.2410 | 0.6863 | 0.1339 |

### F.5 Subspace-constraint evaluation

We evaluate two soft penalties and runtime hard projection against the exact Seed1 generation checkpoint used in the main generation experiment. These completed runs test Eq. 5. A is the same frozen Seed1 anchor with the linear gate; D is an ordinary control residual.

Table 21: **Control response and generation retention.** Full test set: 768 requests × two sampling seeds, with families weighted equally. Metrics compare against the requested-weight GT. A is the same frozen Seed1 anchor under identical zero-displacement inputs and noise. The final two rows verify B/C against Seed1 under identical zero-displacement inputs and noise; they do not measure nonzero-displacement fidelity.

| Method | Ink error ↓ | LPIPS ↓ | IoU ↑ |
|---|---|---|---|
| Generation anchor (Seed1) | 0.0451 | 0.1820 | 0.5472 |
| A: unconstrained | 0.0135 | 0.1876 | 0.5208 |
| B: soft constraint ($\lambda = 0.01$) | 0.0102 | 0.1537 | 0.5832 |
| C: hard projection | 0.0118 | 0.1565 | 0.5710 |
| Exact ordinary-generation recovery at $d = 0$ | | | B: 48/48; C: 48/48 |
| Exact test-image recovery at $d = 0$ | | | B: 240/240; C: 240/240 |

**Matched constrained runs.** Both soft candidates ($\lambda = 0.01, 0.1$) and hard projection freeze the Seed1 rank-16 adapter and train a rank-16 control residual with $g(d) = d$ and $d = (u_t - 500)/800$. The 6,018 source-500 training examples use the same ordered references, initial factors, noise schedule, and actual 16,384 sampled draws. Each run performs 256 updates at global batch 64, learning rate $10^{-4}$ and weight decay $10^{-4}$. Positive and negative displacements are trained together. The frozen base and generation factors are unchanged; 480 FP32 control factor tensors contain 23,592,960 trainable parameters. The exact composite rank is at most 32, not 16. Eight-rank checkpoint, optimizer-step, draw-trace, and zero-displacement prediction checks are retained.

For all 240 adapted attention projections, fixed effective column bases of the full $B_g A_g$ update are extracted using QR and a small SVD with relative singular-value threshold $10^{-6}$ and zero absolute threshold. The normalized soft loss uses $\varepsilon = 10^{-12}$. Hard projection applies $(I - U_g U_g^\top)\widetilde{B}_c$ in every training and inference forward pass; the saved raw factors alone are not its effective update.

Final mean energy overlap is $3.13 \times 10^{-4}$ for soft 0.01, $2.70 \times 10^{-5}$ for soft 0.1, and $1.56 \times 10^{-13}$ after hard projection. These are parameter-space diagnostics.

**Retention and verification.** Under identical zero-displacement conditions, Seed1 and all constrained endpoints have identical PNG, RGB, latent, noise, and prompt hashes on 48 ordinary-generation, 126 validation, and 240 test instances. This follows from the frozen anchor and zero gate, not orthogonality. All 6,144 primary test images and A's 2,394 images are recovered with file/pixel checksums; independent geometry rescoring agrees. Independent LPIPS and SSIM recomputation of the archived outputs agrees on 3,240 primary validation, 6,144 primary test, 1,620 A/D validation, and 3,072 A/D test outputs (tolerance $10^{-7}$); independent recomputation also agrees on 1,620 validation and 3,072 test outputs of the linear-gate/anchor pair. Complete GT, actual references, outputs, and failures are retained.

**Handwritten, calligraphic, and display-font supplement.** To extend coverage beyond the mostly regular test faces, we froze 18 complex faces (five handwritten, five calligraphic, eight display), ten target characters, displacements $d \in \{-0.25, 0, 0.25\}$, and the same two sampling seeds before this panel's sampling. All four control methods and both corresponding generation anchors are retained, giving 5,040 native images. This is an additional stress test after inspection of the original test results, not a new validation set or a basis for selecting $\lambda$. All ten characters occurred in generation-stage references; whole-adaptation unseen-character generalization is not established. Font lineages and pretraining exposure remain unaudited for these faces. These are static single-weight fonts: only $d = 0$ has a genuine target image, and no target-weight error is assigned at nonzero displacement. Model and input checksums are retained, and all scores use a common backend.

Table 22: Complex-font supplement: 18 faces, ten characters, two sampling seeds, and three displacements (1,080 images per control method). Scores weight faces equally. $\Delta I/I_0$ measures achieved ink response; LPIPS and IoU compare each output with its own $d = 0$ output, so they include the intended weight change and are not pure font-fidelity scores. A-linear changes only the inference gate.

| Method | $d$ | $\Delta I/I_0$ (%) | LPIPS to $d = 0$ | IoU to $d = 0$ |
|---|---|---|---|---|
| Soft $\lambda = 0.01$ | -0.25 | -16.74 | 0.0998 | 0.7631 |
| Soft $\lambda = 0.01$ | +0.00 | +0.00 | 0.0000 | 1.0000 |
| Soft $\lambda = 0.01$ | +0.25 | +19.57 | 0.1127 | 0.7812 |
| Hard projection | -0.25 | -17.01 | 0.1046 | 0.7564 |
| Hard projection | +0.00 | +0.00 | 0.0000 | 1.0000 |
| Hard projection | +0.25 | +19.78 | 0.1129 | 0.7826 |
| A: unconstrained | -0.25 | -16.79 | 0.1118 | 0.7602 |
| A: unconstrained | +0.00 | +0.00 | 0.0000 | 1.0000 |
| A: unconstrained | +0.25 | +19.06 | 0.1191 | 0.7869 |
| A, linear gate | -0.25 | -11.31 | 0.0836 | 0.8277 |
| A, linear gate | +0.00 | +0.00 | 0.0000 | 1.0000 |
| A, linear gate | +0.25 | +12.07 | 0.0815 | 0.8463 |

B01 and C have similar average relative ink responses: $-16.74\%/+19.57\%$ and $-17.01\%/+19.78\%$ for lighter/heavier requests. Strict three-point monotonicity holds for 360/360 B01 groups and 358/360 C groups; the two failures remain in the audit. A has 358/360 monotonic groups and A-linear 356/360. The latter's smaller distance to its own zero-displacement output accompanies weaker control ($-11.31\%/+12.07\%$, versus A's $-16.79\%/+19.06\%$), so it cannot be read as a preservation gain at equal control strength.

For B01 versus C, matching absolute differences in relative ink response at all three frozen tolerances (0.02, 0.05, 0.1) retains 603, 697, and 717 of 720 nonzero paired outputs. Averaging within each face and then across faces gives B01-minus-C LPIPS-to-own-zero differences of $-0.0008, -0.0020, -0.0028$, and IoU-to-own-zero differences of $+0.0004, +0.0018, +0.0031$. These are small descriptive differences conditioned on achieved response. The full category, direc-

tion, face, and coverage breakdowns are retained; sampling seeds are not training repetitions. All 5,040 original images, true references, and source-weight targets are retained and available. Image hashes and independent geometry checks pass; independent perceptual rescoring agrees for all four control methods (4,320 outputs).

**Stronger thinning on a fixed complex-font subset.** After inspecting the $d = -0.25$ images, the author requested stronger thinning. Before sampling this diagnostic, we fixed six faces (two each handwritten, calligraphic, and display), four structurally complex characters, the same two sampling seeds, and $d \in \{0, -0.25, -0.4, -0.5\}$. Table 23 includes all 576 newly generated outputs, including new zero and $-0.25$ references for each method. All runs retain runtime projection and the same requests, noise, 40 steps, CFG 1, and native resolution. This is a post-hoc stress diagnostic, not a validation set; it does not change the selected $\lambda$. The four characters have generation-reference exposure.

Table 23: Stronger-thinning diagnostic on six complex faces, four characters, and two sampling seeds: 48 outputs per method and displacement, 576 in total. Scores weight faces equally. LPIPS/IoU to each method's own zero output include the intended weight change; nonzero requests have no target-weight GT.

| Method | $d$ | $\Delta I/I_0$ (%) | LPIPS to $d = 0$ | IoU to $d = 0$ |
|---|---|---|---|---|
| Soft $\lambda = 0.01$ | +0.00 | +0.00 | 0.0000 | 1.0000 |
| Soft $\lambda = 0.01$ | -0.25 | -15.64 | 0.0949 | 0.7838 |
| Soft $\lambda = 0.01$ | -0.40 | -24.61 | 0.1487 | 0.6652 |
| Soft $\lambda = 0.01$ | -0.50 | -29.19 | 0.1791 | 0.6079 |
| Hard projection | +0.00 | +0.00 | 0.0000 | 1.0000 |
| Hard projection | -0.25 | -15.96 | 0.0977 | 0.7802 |
| Hard projection | -0.40 | -24.76 | 0.1554 | 0.6584 |
| Hard projection | -0.50 | -29.31 | 0.1840 | 0.6007 |
| A: unconstrained | +0.00 | +0.00 | 0.0000 | 1.0000 |
| A: unconstrained | -0.25 | -15.97 | 0.1245 | 0.7656 |
| A: unconstrained | -0.40 | -23.67 | 0.1647 | 0.6794 |
| A: unconstrained | -0.50 | -28.44 | 0.1860 | 0.6308 |

Stronger negative requests produce larger mean ink reductions. At $d = -0.5$, B01 and C reduce ink by 29.19% and 29.31% relative to their own zero outputs, versus 15.64% and 15.96% at $-0.25$. Strict four-level ink monotonicity holds for 46/48 groups for each constrained model and 48/48 for A; all failures are retained. Thus larger requests strengthen the average response but do not ensure monotonic behavior on every glyph or preserved legibility. The small soft–hard differences in distance to zero include the intended thinning. B01 and C match all five zero-displacement hashes on all 48 instances. Full-resolution images, actual references, per-face scores, and file/pixel checksums are retained; independent geometry checks agree. Independent LPIPS and SSIM recomputation of the archived outputs agrees for all 576 outputs (tolerance $10^{-7}$).

## G Failure Index and Visual Assessment

The raw examples preserve complete canvases, grayscale values, missing display cells, and unsuccessful predictions, with no sharpening, thresholding, repainting, selective stroke repair, or undisclosed crop applied to improve a research glyph. Figure selection and quantitative sample membership are separate: the full evaluation panel determines statistics even when only a fixed subset fits a display.

| Evidence location | Failure or limitation to inspect |
|---|---|
| Figure 2 | Character/style errors and image-like outputs. |
| Appendix D | Character-identity failures, including the recognizer's own ground-truth misreads and the pilot's blank output and confusable-character errors. |
| Table 20; Appendix F.4 | Reference tracking under a substituted face; the left column is the adverse score that motivates the right one. |
| Figure 6; Appendix C.1 | Retrained LF-Font official-inference errors and the AGDFont local substitute; complete raw failures are retained. |
| Figure 12 | Severe K8 degradation, retained under the same raw display policy. |
| Figure 14 | Template intervention failures across the same four conditions. |
| Figure 15; Table 14 | Direction/seed-dependent displacement behavior. |
| Figures 18, 19 | Family-level thinning costs and response benefits, kept separate from the held-out-family panel. |
| Appendix C.3 | Full six-family raw outputs. |

Pixel metrics and automated inspection are screening evidence; final character correctness and style consistency require human inspection of the actual outputs alongside target GT and the references consumed. No blind font-following study is included in this record; Appendix D records character-identity checks and recognizer limitations, and Appendix F.4 scores the replacement condition against the substituted face's own glyph.