GlyphPrinter: Region-Grouped Direct Preference Optimization for Glyph-Accurate Visual Text Rendering

Xincheng Shuai[1*] Ziye Li[1*] Henghui Ding[1✉] Dacheng Tao[2]
[1]Institute of Big Data, College of Computer Science and Artificial Intelligence, Fudan University, China
[2]Generative AI Lab, College of Computing and Data Science, Nanyang Technological University, Singapore
henghui.ding@gmail.com dacheng.tao@gmail.com
https://henghuiding.com/GlyphPrinter/

A. More Details of Stage 1

A.1. Construction of the Training Dataset

Following previous works [14, 15, 20], we train the GlyphPrinter with both a synthetic dataset and a collected realistic dataset. The synthetic dataset is used to improve the model's text rendering ability. Specifically, we use 560K images from LAION dataset as background, which are then overlaid with multilingual characters (languages include Latin, Chinese, Japanese, Korean, Arabic, and Thai) through the following steps: **1)** For each background image, we first use an optimization algorithm to obtain 1 to 4 non-overlapping bounding boxes, each with a random orientation, rotation angle, and size. **2)** For each bounding box, we randomly select a language and insert an appropriate number of characters based on the box size. **3)** We randomly select bounding boxes and apply distortion to them. Several samples are provided on the left of Fig. I.

To generate text images with diverse styles, we additionally collect 30K high-quality text images from the internet. These images are then annotated with an OCR model [4] and VLM [1] for text recognition and image caption, respectively. Following the previous work [15], placeholders such as `<sks1>`, `<sks2>`, ..., `<sksn>` are used in captions to represent the text from each text block. This dataset mainly contains Chinese and English texts, as illustrated on the right side of Fig. I.

(a) Synthetic Dataset **(b) Realistic Dataset**

Figure I. Samples from the **(a)** synthetic dataset and **(b)** realistic dataset used in Stage 1.

B. More Details of Stage 2

B.1. Construction of GlyphCorrector

In Stage 2, we construct GlyphCorrector to make the model learn from region-level preference pairs, improving the glyph accuracy. Each annotator labels their assigned samples by comparing generated characters with the synthetic rendered ones,

*Equal contribution.
✉ Corresponding author (henghui.ding@gmail.com).

Figure II. Samples from the proposed GlyphCorrector. The erroneous regions are highlighted with green boxes.

followed by cross-review and iterative correction until all errors are resolved. Some examples are shown in Fig. II.

The motivation for generating a group of images per condition. The traditional DPO objective [16] only considers a single preference pair in each training batch, which is insufficient for visual text rendering tasks. This is because a small number of samples are difficult to cover completely accurate glyphs, particularly for long or complex glyph conditions. For instance, given the glyph condition "12345678", the first sample contains accurate "1234", while the second one accurately renders "3456". However, the model still fails to learn the accurate "78" with this single preference pair. Therefore, generating a group of images for each condition enhances sample diversity, enabling the model to learn accurate glyphs from various samples and thereby improving model performance.

B.2. Derivation of Eq. (8)

Here, we derive that Eq. (8) is exactly the implicit reward model learned from Eq. (7). First, we reformulate Eq. (7) as:

$$
\begin{aligned}
&\min_{p_\theta} -E_{c,x_0\sim p_\theta(x_0|c)}[r_m(x_0,c,M)/\beta] + \Sigma_{t=1}^{T} E_{c,p_\theta(x_t|c)}[\mathcal{D}_{\mathrm{KL}}\big(p_\theta(M(x_{t-1})|x_t,c)||p_{\mathrm{ref}}(M(x_{t-1})|x_t,c)\big)] \\
=&\min_{p_\theta} -E_{c,p_\theta(x_{0:T}|c)}[\Sigma_{t=0}^{T-1} R_m(M(x_t),c)/\beta] + \Sigma_{t=1}^{T} E_{c,p_\theta(x_t|c)}[\mathcal{D}_{\mathrm{KL}}\big(p_\theta(M(x_{t-1})|x_t,c)||p_{\mathrm{ref}}(M(x_{t-1})|x_t,c)\big)] \\
=&\min_{p_\theta} -E_{c,p_\theta(x_{0:T}|c)}[\Sigma_{t=0}^{T-1} R_m(M(x_t),c)/\beta] + \Sigma_{t=1}^{T} E_{c,p_\theta(x_t|c)} E_{p_\theta(M(x_{t-1})|x_t,c)}\left[\log \frac{p_\theta(M(x_{t-1})|x_t,c)}{p_{\mathrm{ref}}(M(x_{t-1})|x_t,c)}\right] \\
=&\min_{p_\theta} -E_{c,p_\theta(x_{0:T}|c)}[\Sigma_{t=0}^{T-1} R_m(M(x_t),c)/\beta] + \Sigma_{t=1}^{T} E_{c,p_\theta(x_t|c)} E_{p_\theta(x_{t-1}|x_t,c)}\left[\log \frac{p_\theta(M(x_{t-1})|x_t,c)}{p_{\mathrm{ref}}(M(x_{t-1})|x_t,c)}\right] \\
=&\min_{p_\theta} -E_{c,p_\theta(x_{0:T}|c)}[\Sigma_{t=0}^{T-1} R_m(M(x_t),c)/\beta] + E_{c,p_\theta(x_{0:T}|c)}\left[\Sigma_{t=1}^{T} \log \frac{p_\theta(M(x_{t-1})|x_t,c)}{p_{\mathrm{ref}}(M(x_{t-1})|x_t,c)}\right] \\
=&\min_{p_\theta} E_{c,p_\theta(x_{0:T}|c)}\left[-\Sigma_{t=1}^{T} R_m(M(x_{t-1}),c)/\beta + \Sigma_{t=1}^{T} \log \frac{p_\theta(M(x_{t-1})|x_t,c)}{p_{\mathrm{ref}}(M(x_{t-1})|x_t,c)}\right],
\end{aligned}
\tag{i}
$$

where $r_m(x_0,c,M)$ is decomposed as $r_m(x_0,c,M) = E_{p_\theta(x_{1:T}|x_0,c)}[\Sigma_{t=0}^{T-1} R_m(M(x_t),c)]$, and $R_m(M(x_t),c)$ is the step-wise reward function.

Then, we formulate the sampling process of diffusion models under the multi-step RL framework as in [16]. We first define the initial state distribution ρ_0, state transition dynamics P_s, policy function π, action space $\mathcal{A}$, and the state space $\mathcal{S}$. Specifically, $P_s(s_{t+1}|s_t,a_t)$ takes the current state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$ as input, returning the distribution of the next state s_{t+1}. The policy $\pi(a_t|s_t)$ determines the action for the current state. Different from previous literature, we propose the

reward function $\hat{r}(s_t, \hat{a}_t)$, which receives s_t and the action $\hat{a}_t$ from a subspace of $\mathcal{A}$. Then, we have:

$$\begin{aligned} s_t &\triangleq (x_t, t, c) \\ \rho_0 &\triangleq (\mathcal{N}(\mathbf{0}, \mathbf{I}), \delta_T, p(c)) \\ a_t &\triangleq x_{t-1} \\ \hat{a}_t &\triangleq M(x_{t-1}) \\ P_s(s_{t+1}|s_t, a_t) &\triangleq (\delta_{x_{t-1}}, \delta_{t-1}, \delta_c) \\ \hat{r}(s_t, \hat{a}_t) &\triangleq R_m(M(x_{t-1}), c), \end{aligned} \tag{ii}$$

where δ is the Dirac delta function. Following [12], we can derive the optimal state-action value function Q_m^*, the optimal state value function V_m^*, and the optimal distribution p_θ^* from Eq. (i):

$$\begin{aligned} Q_m^*((x_t, t, c), M(x_{t-1})) &= R_m(M(x_{t-1}), c) + E_{p_\theta^*(x_{t-1}|x_t, c, M(x_{t-1}))}[V_m^*(x_{t-1}, t-1, c, M)], \\ V_m^*(x_t, t, c, M) &= \beta \log \int \exp(Q^*((x_t, t, c), M(x_{t-1}))/\beta) d(M(x_{t-1})), \\ p_\theta^*(M(x_{t-1})|x_t, c) &= p_{\text{ref}}(M(x_{t-1})|x_t, c) \exp\big((Q_m^*((x_t, t, c), M(x_{t-1})) - V_m^*(x_t, t, c, M))/\beta\big). \end{aligned} \tag{iii}$$

Therefore, we have:

$$\begin{aligned} Q_m^*((x_t, t, c), M(x_{t-1})) &= \beta \log \frac{p_\theta^*(M(x_{t-1})|x_t, c)}{p_{\text{ref}}(M(x_{t-1})|x_t, c)} + V_m^*(x_t, t, c, M), \\ R_m(M(x_{t-1}), c) &= Q_m^*((x_t, t, c), M(x_{t-1})) - E[V_m^*(x_{t-1}, t-1, c, M)] \\ &= \beta \log \frac{p_\theta^*(M(x_{t-1})|x_t, c)}{p_{\text{ref}}(M(x_{t-1})|x_t, c)} + V_m^*(x_t, t, c, M) - E[V_m^*(x_{t-1}, t-1, c, M)]. \end{aligned} \tag{iv}$$

The total regional reward function is:

$$\begin{aligned} r_m(x_0, c, M) &= E_{p_\theta^*(x_{1:T}|x_0, c)}[\Sigma_{t=0}^{T-1} R_m(M(x_t), c)] \\ &= E_{p_\theta^*(x_{1:T}|x_0, c)}[\Sigma_{t=1}^{T} R_m(M(x_{t-1}), c)] \\ &= E_{p_\theta^*(x_{1:T}|x_0, c)}\Big[\Sigma_{t=1}^{T} \beta \log \frac{p_\theta^*(M(x_{t-1})|x_t, c)}{p_{\text{ref}}(M(x_{t-1})|x_t, c)} + V_m^*(x_t, t, c, M) - E[V_m^*(x_{t-1}, t-1, c, M)]\Big] \\ &= \beta T E_{t, x_{t-1,t} \sim p_\theta^*(x_{t-1,t}|x_0, c)}\Big[\log \frac{p_\theta^*(M(x_{t-1})|x_t, c)}{p_{\text{ref}}(M(x_{t-1})|x_t, c)}\Big] + \\ &\qquad E_{p_\theta^*(x_{1:T}|x_0, c)}[\Sigma_{t=1}^{T} V_m^*(x_t, t, c, M) - E[V_m^*(x_{t-1}, t-1, c, M)]] \\ &= \beta T E_{t, x_{t-1,t} \sim p_\theta^*(x_{t-1,t}|x_0, c)}\Big[\log \frac{p_\theta^*(M(x_{t-1})|x_t, c)}{p_{\text{ref}}(M(x_{t-1})|x_t, c)}\Big] + Z_m(c, M), \end{aligned} \tag{v}$$

which is exactly the form of Eq. (8).

B.3. Alternative Approximate Derivation of Eq. (8).

We also provide an alternative approximate derivation of Eq. (8). Start from Eq. (i), we have:

$$
\begin{aligned}
&\min_{p_\theta} E_{c,p_\theta(x_{0:T}|c)}\Big[-\Sigma_{t=1}^{T} R_m(M(x_{t-1}),c)/\beta + \Sigma_{t=1}^{T}\log\frac{p_\theta(M(x_{t-1})|x_t,c)}{p_{\text{ref}}(M(x_{t-1})|x_t,c)}\Big]\\
=&\min_{p_\theta} E_{c,p_\theta(x_{0:T}|c)}\Big[-\Sigma_{t=1}^{T} R_m(M(x_{t-1}),c)/\beta + \log\frac{\prod_{t=1}^{T} p_\theta(M(x_{t-1})|x_t,c)}{\prod_{t=1}^{T} p_{\text{ref}}(M(x_{t-1})|x_t,c)}\Big]\\
=&\min_{p_\theta} E_{c,p_\theta(x_{0:T}|c)}\Big[\log\frac{\prod_{t=1}^{T} p_\theta(M(x_{t-1})|x_t,c)}{\prod_{t=1}^{T} p_{\text{ref}}(M(x_{t-1})|x_t,c)\exp(R_m(M(x_{t-1}),c)/\beta)}\Big]\\
=&\min_{p_\theta} E_{c,p_\theta(x_{0:T}|c)}\Big[\Sigma_{t=1}^{T}\log\frac{p_\theta(M(x_{t-1})|x_t,c)}{p_{\text{ref}}(M(x_{t-1})|x_t,c)\exp(R_m(M(x_{t-1}),c)/\beta)}\Big]\\
=&\min_{p_\theta} \Sigma_{t=1}^{T} E_{c,p_\theta(x_t|c)} E_{p_\theta(x_{t-1}|x_t,c)}\Big[\log\frac{p_\theta(M(x_{t-1})|x_t,c)}{p_{\text{ref}}(M(x_{t-1})|x_t,c)\exp(R_m(M(x_{t-1}),c)/\beta)}\Big]\\
=&\min_{p_\theta} \Sigma_{t=1}^{T} E_{c,p_\theta(x_t|c)} E_{p_\theta(M(x_{t-1})|x_t,c)}\Big[\log\frac{p_\theta(M(x_{t-1})|x_t,c)}{p_{\text{ref}}(M(x_{t-1})|x_t,c)\exp(R_m(M(x_{t-1}),c)/\beta)/Z_m^t(c,M)} - \log(Z_m^t(c,M))\Big]\\
=&\min_{p_\theta} \Sigma_{t=1}^{T} E_{c,x_t}[\mathcal{D}_{\text{KL}}\big(p_\theta(M(x_{t-1})|x_t,c)||p_{\text{ref}}(M(x_{t-1})|x_t,c)\exp(R_m(M(x_{t-1}),c)/\beta)/Z_m^t(c,M)\big)] - E[\log(Z_m^t(c,M))],
\end{aligned}
\tag{vi}
$$

where $Z_m^t(c,M) = \Sigma_{x_{t-1}} p_{\text{ref}}(M(x_{t-1})|x_t,c)\exp(R_m(M(x_{t-1}),c)/\beta)$ is the regional partition function for each step. Therefore, the optimal distribution $p_\theta^*(M(x_{t-1})|x_t,c)$ for each step can be approximated as:

$$
p_\theta^*(M(x_{t-1})|x_t,c) = p_{\text{ref}}(M(x_{t-1})|x_t,c)\exp(R_m(M(x_{t-1}),c)/\beta)/Z_m^t(c,M). \tag{vii}
$$

Then, we have:

$$
R_m(M(x_{t-1}),c) = \beta\log\frac{p_\theta^*(M(x_{t-1})|x_t,c)}{p_{\text{ref}}(M(x_{t-1})|x_t,c)} + \beta\log Z_m^t(c,M). \tag{viii}
$$

The total regional reward function is:

$$
\begin{aligned}
r_m(x_0,c,M) &= E_{p_\theta^*(x_{1:T}|x_0,c)}[\Sigma_{t=0}^{T-1} R_m(M(x_t),c)]\\
&= E_{p_\theta^*(x_{1:T}|x_0,c)}[\Sigma_{t=1}^{T} R_m(M(x_{t-1}),c)]\\
&= E_{p_\theta^*(x_{1:T}|x_0,c)}\Big[\beta\Sigma_{t=1}^{T}\log\frac{p_\theta^*(M(x_{t-1})|x_t,c)}{p_{\text{ref}}(M(x_{t-1})|x_t,c)} + \beta\Sigma_{t=1}^{T}\log Z_m^t(c,M)\Big]\\
&= E_{p_\theta^*(x_{1:T}|x_0,c)}\Big[\beta\Sigma_{t=1}^{T}\log\frac{p_\theta^*(M(x_{t-1})|x_t,c)}{p_{\text{ref}}(M(x_{t-1})|x_t,c)}\Big] + \beta E_{p_\theta^*(x_{1:T}|x_0,c)}[\Sigma_{t=1}^{T}\log Z_m^t(c,M)]\\
&= \beta T E_{t,x_{t-1,t}\sim p_\theta^*(x_{t-1,t}|x_0,c)}\Big[\log\frac{p_\theta^*(M(x_{t-1})|x_t,c)}{p_{\text{ref}}(M(x_{t-1})|x_t,c)}\Big] + Z_m(c,M),
\end{aligned}
\tag{ix}
$$

which is exactly the form of Eq. (8).

B.4. Derivation of Eq. (9)

The derivation of Eq. (9) is introduced below. We first substitute r_m terms from Eq. (9) with Eq. (8):

$$
\begin{aligned}
L_{\text{R-DPO}} &= -E_{c,x_0^w,x_0^l,M^w,M^l}\Big[\log\sigma\big(r_m(x_0^w,c,M^w) - r_m(x_0^l,c,M^l)\big)\Big]\\
&= -E_{c,x_0^w,x_0^l,M^w,M^l}\Big[\log\sigma\big(\beta T E_{t,x_t^w,x_t^l} E_{x_{t-1}^w,x_{t-1}^l}[\log\frac{p_\theta(M^w(x_{t-1}^w)|x_t^w,c)}{p_{\text{ref}}(M^w(x_{t-1}^w)|x_t^w,c)} -\\
&\log\frac{p_\theta(M^l(x_{t-1}^l)|x_t^l,c)}{p_{\text{ref}}(M^l(x_{t-1}^l)|x_t^l,c)}] + \Delta Z_m(c,M^w,M^l)\big)\Big],
\end{aligned}
\tag{x}
$$

where $\Delta Z_m(c, M^w, M^l)$ takes the form $Z_m(c, M^w) - Z_m(c, M^l)$. By Jensen's inequality, we have:

$$\begin{aligned} L_{\text{R-DPO}} \leq & -E_{c,t,x_0^w,x_0^l,M^w,M^l} E_{x_t^w,x_t^l} \Big[\log\sigma\big(\beta T E_{x_{t-1}^w \sim q(x_{t-1}|x_{0,t}^w), x_{t-1}^l \sim q(x_{t-1}|x_{0,t}^l)} [\log \frac{p_\theta(M^w(x_{t-1}^w)|x_t^w, c)}{p_{\text{ref}}(M^w(x_{t-1}^w)|x_t^w, c)} - \\ & \log \frac{p_\theta(M^l(x_{t-1}^l)|x_t^l, c)}{p_{\text{ref}}(M^l(x_{t-1}^l)|x_t^l, c)}] + \Delta Z_m(c, M^w, M^l)\big)\Big] \\ = & -E_{c,t,x_0^w,x_0^l,M^w,M^l} E_{x_t^w,x_t^l} \Big[\log\sigma\big(\beta T E_{M^w(x_{t-1}^w) \sim q(M^w(x_{t-1})|x_{0,t}^w), M^l(x_{t-1}^l) \sim q(M^l(x_{t-1})|x_{0,t}^l)} \\ & [\log \frac{p_\theta(M^w(x_{t-1}^w)|x_t^w, c)}{p_{\text{ref}}(M^w(x_{t-1}^w)|x_t^w, c)} - \log \frac{p_\theta(M^l(x_{t-1}^l)|x_t^l, c)}{p_{\text{ref}}(M^l(x_{t-1}^l)|x_t^l, c)}] + \Delta Z_m(c, M^w, M^l)\big)\Big], \end{aligned} \tag{xi}$$

where we approximate p_θ with forward process q as in [19]. Then, we have:

$$\begin{aligned} L_{\text{R-DPO}} \leq & -E_{c,t,x_0^w,x_0^l,M^w,M^l} E_{x_t^w,x_t^l} \Big[\log\sigma\big(\beta T E_{M^w(x_{t-1}^w), M^l(x_{t-1}^l)} [\log p_\theta(M^w(x_{t-1}^w)|x_t^w, c) - \log q(M^w(x_{t-1}^w)|x_{0,t}^w) - \\ & (\log p_{\text{ref}}(M^w(x_{t-1}^w)|x_t^w, c) - \log q(M^w(x_{t-1}^w)|x_{0,t}^w)) - (\log p_\theta(M^l(x_{t-1}^l)|x_t^l, c) - \log q(M^l(x_{t-1}^l)|x_{0,t}^l) - \\ & (\log p_{\text{ref}}(M^l(x_{t-1}^l)|x_t^l, c) - \log q(M^l(x_{t-1}^l)|x_{0,t}^l)))] + \Delta Z_m(c, M^w, M^l)\big)\Big] \\ = & -E_{c,t,x_0^w,x_0^l,M^w,M^l} E_{x_t^w,x_t^l} \Big[\log\sigma\Big(- \beta T \big(\mathcal{D}_{\text{KL}}\big(q(M^w(x_{t-1}^w)|x_{0,t}^w)||p_\theta(M^w(x_{t-1}^w)|x_t^w, c)\big) - \\ & \mathcal{D}_{\text{KL}}\big(q(M^w(x_{t-1}^w)|x_{0,t}^w)||p_{\text{ref}}(M^w(x_{t-1}^w)|x_t^w, c)\big) - \big(\mathcal{D}_{\text{KL}}\big(q(M^l(x_{t-1}^l)|x_{0,t}^l)||p_\theta(M^l(x_{t-1}^l)|x_t^l, c)\big) - \\ & \mathcal{D}_{\text{KL}}\big(q(M^l(x_{t-1}^l)|x_{0,t}^l)||p_{\text{ref}}(M^l(x_{t-1}^l)|x_t^l, c)\big)\big)\Big) + \Delta Z_m(c, M^w, M^l)\Big)\Big]. \end{aligned} \tag{xii}$$

From previous literature [10], we have:

$$E_{c,t,x_0} E_{x_t}[\mathrm{D}_{\text{KL}}(q(x_{t-1}|x_{0,t})||p_\theta(x_{t-1}|x_t, c))] = E_{c,t,x_0,\epsilon}[\omega_t||\epsilon - \epsilon_\theta(x_t, t, c)||^2], \tag{xiii}$$

where $p_\theta(x_{t-1}|x_t, c)$ takes the form $\mathcal{N}(x_{t-1}|\mu_\theta, \sigma_t^2\mathbf{I})$, and $q(x_{t-1}|x_{0,t}) := \mathcal{N}(x_{t-1}|\mu_t, \sigma_t^2\mathbf{I})$. Since M is a linear transformation, $p_\theta(M(x_{t-1})|x_t, c)$ and $q(M(x_{t-1})|x_{0,t})$ become to $\mathcal{N}(x_{t-1}|M(\mu_\theta), \sigma_t^2\mathbf{I})$ and $\mathcal{N}(x_{t-1}|M(\mu_t), \sigma_t^2\mathbf{I})$, respectively. Therefore, we can extend the Eq. (xiii) into following regional variant:

$$E_{c,t,x_0,M} E_{x_t}[\mathrm{D}_{\text{KL}}(q(M(x_{t-1})|x_{0,t})||p_\theta(M(x_{t-1})|x_t, c))] = E_{c,t,x_0,M,\epsilon}[\omega_t||M(\epsilon - \epsilon_\theta(x_t, t, c))||^2], \tag{xiv}$$

Since $||v - v_\theta(x_t, t, c)||^2 \propto ||\epsilon - \epsilon_\theta(x_t, t, c)||^2$ [13], we can get following objective by substituting the Eq. (xiv) into each KL-divergence term of Eq. (xii):

$$\begin{aligned} L_{\text{R-DPO}} \leq & -E_{c,t,x_0^w,x_0^l,M^w,M^l,\epsilon^w,\epsilon^l} \Big[\log\sigma\Big(- \beta T \omega_t \big(||M^w(v^w - v_\theta(x_t^w, t, c))||_2^2 - ||M^w(v^w - v_{\text{ref}}(x_t^w, t, c))||_2^2 - \\ & (||M^l(v^l - v_\theta(x_t^l, t, c))||_2^2 - ||M^l(v^l - v_{\text{ref}}(x_t^l, t, c))||_2^2)\big) + \Delta Z_m(c, M^w, M^l)\Big)\Big], \end{aligned} \tag{xv}$$

which is exactly the form of Eq. (9). According to the formulation of $Z_m(c, M)$ from Eq. (v) and Eq. (ix), $\Delta Z_m(c, M^w, M^l)$ equals 0 when $M^w = M^l$.

B.5. Comparison of R-GDPO with Previous Methods

Recently, several studies [8, 21] have introduced their DPO variants for video generation models and Large Language Models, to address the inefficiency of applying overall preferences [16] in their tasks. For example, DenseDPO [21] assigns preference labels to each frame for two video samples, while Mask-DPO [8] leverages sentence-level preference annotations to enable the model to only learn from the correct facts in winning answers and the incorrect contents in losing answers. In contrast, our R-GDPO is proposed for image generation and constructs region-level preference pairs within the spatial dimension. Moreover, while providing a comprehensive derivation, we extend the conventional single preference pair to a group-wise setting, thereby enhancing sample diversity and data utilization efficiency.

For image generation, the most related work to ours is PatchDPO [11]. Similarly, it introduces a patch-level DPO objective for customization tasks to improve subject consistency. However, our R-GDPO differs from the method in the following

aspects. **1).** Essentially, PatchDPO employs a weighted diffusion loss, which can be regarded as a softer variant of our Mask-SFT discussed in the ablation studies. Specifically, it assigns high/low weights to superior/inferior patches within an image, with weights normalized to $[0, 1]$. However, the objective inherently lacks explicit penalty signals from losing samples, leading to suboptimal model performance, as discussed in our ablation studies. In contrast, our R-GDPO aligns the model outputs with winning glyph regions, while distancing them from losing ones, thereby learning localized glyph correctness. **2).** Similar to previous works [8, 16, 21], PatchDPO objective only considers a single preference pair for each batch, which limits the sample diversity. Our R-GDPO objective, on the other hand, generates a group of images per condition, allowing the model to learn from the superior regions across different samples and thereby improving overall model performance.

C. More Details of RRG

First, we prove that Eq. (12) derives the optimal distribution with adjustable regularization weight. As shown in Eq. (iii) and Eq. (vii), since $p_\theta(x_t|c) \propto p_{\text{ref}}(x_t|c)\exp(r(x_t, c)/\beta)$, we can reformulate Eq. (12) as :

$$\begin{aligned}
\hat{s}_{\text{ref},\theta}(x_t, \omega, c) &= \nabla_{x_t} \log\big(p_{\text{ref}}(x_t|c)^{(1-\omega)} p_\theta(x_t|c)^{\omega}\big) \\
&= \nabla_{x_t} \log\big(p_{\text{ref}}(x_t|c)^{(1-\omega)} (p_{\text{ref}}(x_t|c)\exp(r(x_t, c)/\beta))^{\omega}\big) \\
&= \nabla_{x_t} \log\Big(p_{\text{ref}}(x_t|c)\exp(\frac{r(x_t, c)}{\beta/\omega})\Big).
\end{aligned} \tag{xvi}$$

The $p_{\text{ref}}(x_t|c)\exp(\frac{r(x_t,c)}{\beta/\omega})$ in the last row corresponds to the optimal distribution to be sampled. β/ω is the adjustable regularization weight, controlling the glyph accuracy during inference. Then, we give the full derivation of Eq. (13):

$$\begin{aligned}
v^*(x_t, t, c) &= a_t x + b_t\big(\nabla_{x_t} \log\big(p_{\text{ref}}(x_t|c)^{(1-\omega)} p_\theta(x_t|c)^{\omega}\big)\big) \\
&= a_t x + b_t\big((1-\omega)\nabla_{x_t} \log p_{\text{ref}}(x_t|c) + \omega \nabla_{x_t} \log p_\theta(x_t|c)\big) \\
&= a_t x + b_t\big((1-\omega)\frac{v_{\text{ref}}(x_t, t, c) - a_t x}{b_t} + \omega \frac{v_\theta(x_t, t, c) - a_t x}{b_t}\big) \\
&= a_t x + (1-\omega) v_{\text{ref}}(x_t, t, c) + \omega v_\theta(x_t, t, c) - a_t x \\
&= (1-\omega) v_{\text{ref}}(x_t, t, c) + \omega v_\theta(x_t, t, c).
\end{aligned} \tag{xvii}$$

Algorithm 1 demonstrates the pipeline of our RRG.

Algorithm 1: Algorithm of RRG.

Input: Initial noisy image x_T sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$; condition c; Classifier-Free Guidance scale ω; model v_{ref} from Stage 1; model v_θ from Stage 2; sampling step T;
Output: The generated image x;
Get the overall text region mask $\hat{\mathbf{P}}$ from c ;
for t in $\{T, T-1 \ldots, 1\}$ **do**
  $v_{t,\text{ref}} = v_{\text{ref}}(x_t, t, c)$;
  $v_{t,\theta} = v_\theta(x_t, t, c)$;
  $v_t^* = (1-\omega) v_{t,\text{ref}} + \omega v_{t,\theta}$;
  $\hat{v}_t^* = \hat{\mathbf{P}} v_t^* + (\mathbf{I} - \hat{\mathbf{P}}) v_{t,\theta}$;
  $x_{t-1} = x_t + \hat{v}_t^* dt$;
$x = x_0$;
Return: The generated image x.

Comparison of RRG With Previous Methods. The early work Flow-NRG[13] introduces reward guidance in inference time, enhancing model performance by sampling from the optimal distribution. Inspired by the classifier-guidance method [5], Flow-NRG trains an additional reward network to assess noisy images. During inference, it leverages the gradient of the reward network to modulate the velocity field, guiding the sampling process. In contrast, our method builds upon Classifier-Free Guidance (CFG) [9], which combines the predicted velocity fields from models at different stages, enabling sampling from a controllable optimal distribution without training an auxiliary network.

Figure III. Samples from the constructed benchmarks **(a)** GlyphAcc-Multilingual and **(b)** GlyphAcc-Complex.

Inspired by CFG, some concurrent works [3, 6] have also introduced their reward guidance approaches. While they arrive at similar conclusions, our RRG is derived by a different approach. Furthermore, we extend the conventional image-level guidance to the region-level variant, which better preserves the quality of the background content.

D. More Details of Benchmarks

We construct two benchmarks, GlyphAcc-Multilingual and GlyphAcc-Complex, to evaluate the model performance in rendering multilingual texts and characters with complex glyphs, respectively. GlyphAcc-Multilingual consists of 370 test cases spanning seven languages: English, Chinese, Japanese, Korean, French, Vietnamese, and Thai. In contrast, GlyphAcc-Complex contains 97 test cases, focusing on complex Chinese characters. For both benchmarks, we instruct Gemini [17] to provide the image description and the coordinates of 1 to 4 bounding boxes for placing texts. Furthermore, the texts to be rendered are also obtained via Gemini. Representative examples are shown in Fig. III. The prompt for querying Gemini is:

> You are an image layout expert, specializing in designing which areas of a 1024×1024 image should contain text. You will output rectangular text bounding boxes to indicate the positions of the text in the image, in the format:
>
> $$(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$$
>
> (top-left, top-right, bottom-right, bottom-left). For each image, output 1–4 text bounding boxes. The content inside each bounding box should be text in English, Chinese, Japanese, Korean, French, Vietnamese, or Thai. Finally, provide an overall prompt for the image, using placeholders such as `<sks1>`, `<sks2>`, ..., `<sksn>` to denote the text that will be rendered. Output the result in a JSON file.

Figure IV. Comparison results on GlyphAcc-Multilingual.

E. More Details of Evaluation Metrics

We leverage Qwen2.5-VL [1] as VLM to evaluate the following metrics.

1. Text accuracy. To calculate text accuracy metrics, such as Normalized Edit Distance (**NED**) and sentence accuracy (**Sen.Acc**), we first instruct VLM to recognize texts through the following query prompt:

```
You are an expert in text recognition. Please recognize the text in the image and
output it line by line.
```

2. Image aesthetic and text-image alignment. We also employ VLM [1] to evaluate image aesthetic (**Aes**) and text-image alignment (**Text.Align**), using the following query prompt:

```
Please evaluate the provided text-image pair according to the following two criteria.
The input consists of a prompt and a generated image.
1. Image Aesthetic: Assess the visual quality of the image, including factors such as
   color harmony, contrast, and overall appeal.
2. Text-Image Alignment: Evaluate how well the generated image aligns with the given
prompt. For each criterion, provide:
- A score from 1 to 100 (where 1 = poor, 100 = excellent).
- A brief explanation justifying the score.
Return your evaluation in the following JSON-like format:
{
  "Image Aesthetic": {
```

```
    "score": <score>,
    "comment": "<explanation>"
  },
  "Text-Image Alignment": {
    "score": <score>,
    "comment": "<explanation>"
  }
}
```

F. More Details of the User Study

We employ 20 volunteers to conduct the user study with the generated images from GlyphAcc-Multilingual and GlyphAcc-Complex benchmarks. They are asked to assess each image in terms of image aesthetic, text-image alignment, and glyph accuracy. For glyph accuracy, the participants need to compare the rendered characters in the glyph image with the generated ones. All of the above scores are within the range of 1 to 10. For each score, we average the results.

G. More Comparison Results

We provide more comparison results in this section. Fig. IV and Fig. V represent the results from GlyphAcc-Multilingual and GlyphAcc-Complex, respectively. As shown in Fig. IV, most of *prompt-guided* methods fail to generate accurate glyphs for some infrequent languages, as shown in the 3rd row of the figure. For the examples from Fig. V, existing methods exhibit poor performance in rendering complex glyphs. In contrast, our GlyphPrinter outperforms in glyph accuracy across all these

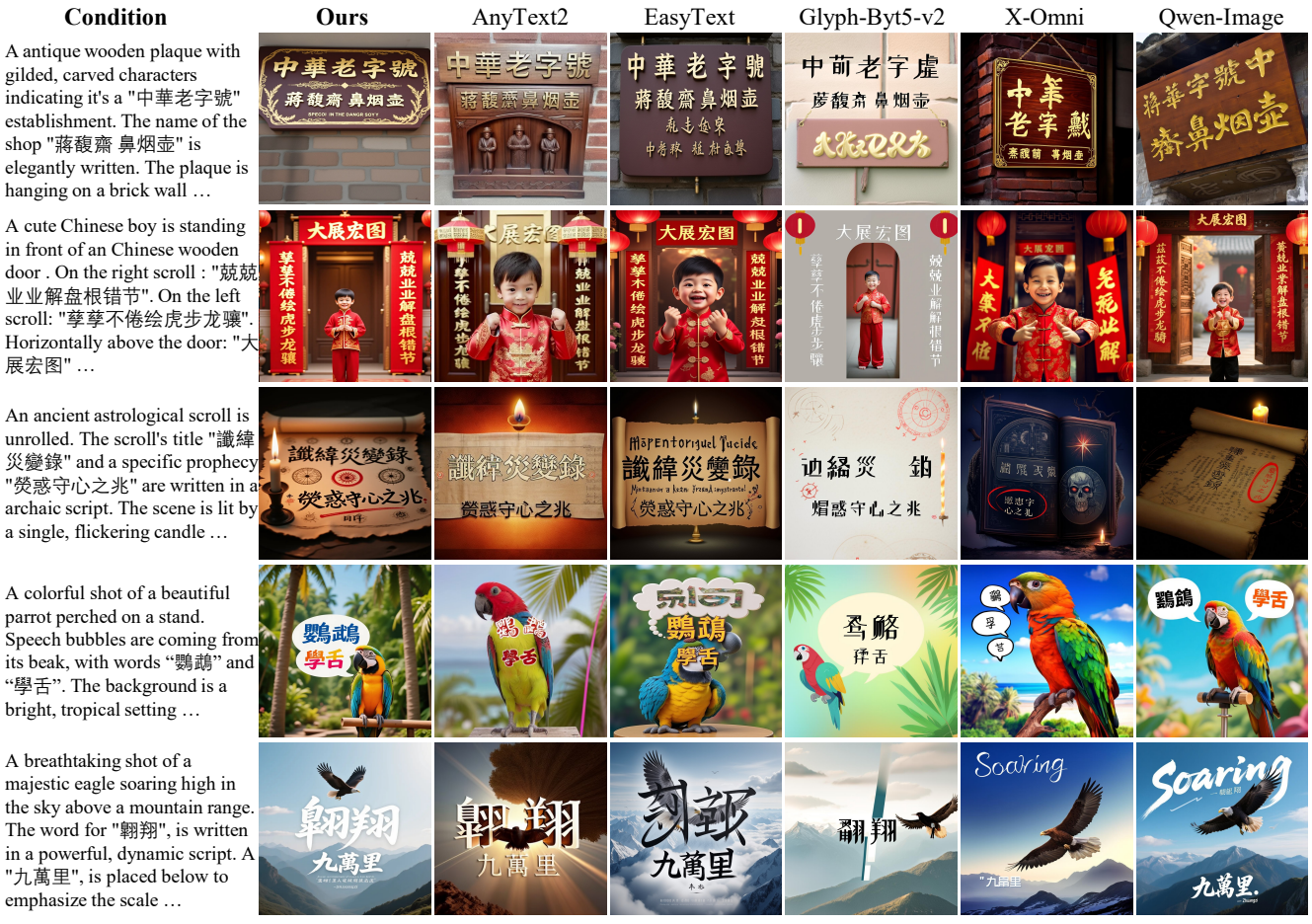

Figure V. Comparison results on GlyphAcc-Complex.

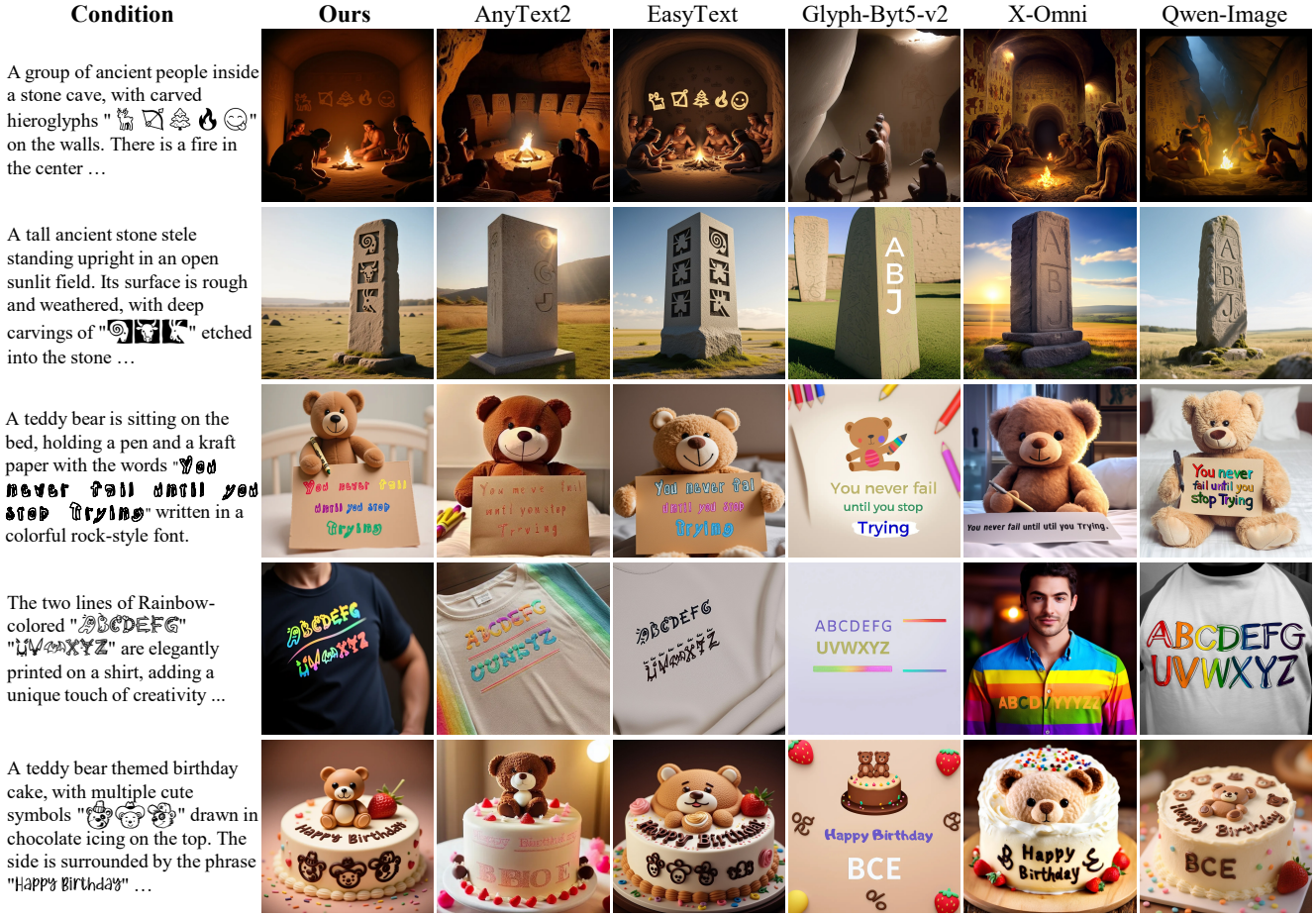

Figure VI. Comparison results under out-of-domain conditions.

Figure VII. More results of ablation studies. The erroneous regions of each image are highlighted with green boxes.

cases, achieving a good balance between the stylization and precision.

In addition, we also illustrate the model performance on out-of-domain inputs, where we construct glyph conditions with emojis and characters with stylized fonts. As shown in Fig. VI, only our method preserves fine-grained structural details of

the condition.

Fig. VII presents more results of ablation studies, demonstrating the effects of our key designs.

We further evaluate the performance of our GlyphPrinter against other comparison methods on the OneIG benchmark [2]. We first leverage Gemini [17] to generate the corresponding layouts using the query prompt below.

> You are an image layout expert, specializing in designing which areas of a 1024×1024 image should contain text. You will output rectangular text bounding boxes to indicate the positions of the text in the image, in the format:
>
> $$(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$$
>
> (top-left, top-right, bottom-right, bottom-left). For each image, generate bounding boxes corresponding to the number of sentences that need to be rendered from the prompt. The content within each bounding box should be taken directly from the quoted prompt. You may split a sentence into smaller segments if it is too long, but each split segment must also generate its own independent bounding box. Finally, provide an overall prompt for the image, using placeholders such as `<sks1>`, `<sks2>`, ..., `<sksn>` to denote the text that will be rendered. Output the result in a JSON file.

As shown in Tab. I, our method achieves the best performance on both English and Chinese scenarios.

Table I. The quantitative results on the OneIG [2] benchmark.

Language	**GlyphPrinter (Ours)**		AnyText2 [18]		EasyText [15]		Glyph-Byt5-v2 [14]		X-Omni [7]		Qwen-Image [20]	
	NED	**Sen.Acc**	**NED**	**Sen.Acc**	**NED**	**Sen.Acc**	**NED**	**Sen.Acc**	**NED**	**Sen.Acc**	**NED**	**Sen.Acc**
English	**0.9704**	**0.8853**	0.6314	0.5301	<u>0.9571</u>	<u>0.8741</u>	0.8060	0.7650	0.8930	0.6353	0.9432	0.8327
Chinese	**0.9771**	**0.8932**	0.7642	0.5089	<u>0.9589</u>	<u>0.8808</u>	0.9287	0.8274	0.7705	0.4199	0.9424	0.8630

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv*, 2025. 1, 8

[2] Jingjing Chang, Yixiao Fang, Peng Xing, Shuhan Wu, Wei Cheng, Rui Wang, Xianfang Zeng, Gang Yu, and Hai-Bao Chen. Oneig-bench: Omni-dimensional nuanced evaluation for image generation. In *NeurIPS*, 2025. 11

[3] Min Cheng, Fatemeh Doudi, Dileep Kalathil, Mohammad Ghavamzadeh, and Panganamala R Kumar. Diffusion blend: Inference-time multi-preference alignment for diffusion models. *arXiv*, 2025. 7

[4] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. *arXiv*, 2025. 1

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 6

[6] Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Diffusion guidance is a controllable policy improvement operator. *arXiv*, 2025. 7

[7] Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, et al. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv*, 2025. 11

[8] Yuzhe Gu, Wenwei Zhang, Chengqi Lyu, Dahua Lin, and Kai Chen. Mask-dpo: Generalizable fine-grained factuality alignment of llms. In *ICLR*, 2025. 5, 6

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv*, 2022. 6

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 5

[11] Qihan Huang, Long Chan, Jinlong Liu, Wanggui He, Hao Jiang, Mingli Song, and Jie Song. Patchdpo: Patch-level dpo for finetuning-free personalized image generation. In *CVPR*, 2025. 5

[12] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv*, 2018. 3

[13] Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Wenyu Qin, Menghan Xia, et al. Improving video generation with human feedback. *arXiv*, 2025. 5, 6

[14] Zeyu Liu, Weicong Liang, Yiming Zhao, Bohan Chen, Lin Liang, Lijuan Wang, Ji Li, and Yuhui Yuan. Glyph-byt5-v2: A strong aesthetic baseline for accurate multilingual visual text rendering. *arXiv*, 2024. 1, 11

[15] Runnan Lu, Yuxuan Zhang, Jiaming Liu, Haofan Wang, and Yiren Song. Easytext: Controllable diffusion transformer for multilingual text rendering. In *AAAI*, 2025. 1, 11

[16] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023. 2, 5, 6

[17] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv*, 2023. 7, 11

[18] Yuxiang Tuo, Yifeng Geng, and Liefeng Bo. Anytext2: Visual text generation and editing with customizable attributes. *arXiv*, 2024. 11

[19] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *CVPR*, 2024. 5

[20] 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*, 2025. 1, 11

[21] Ziyi Wu, Anil Kag, Ivan Skorokhodov, Willi Menapace, Ashkan Mirzaei, Igor Gilitschenski, Sergey Tulyakov, and Aliaksandr Siarohin. Densedpo: Fine-grained temporal preference optimization for video diffusion models. *arXiv*, 2025. 5, 6