

## 6. Typography Attribute Space

This section details the typography attribute space used by *UniLayout Generator*. For each text element  $t$  on the canvas, the model predicts a style vector  $y_t = (f_t, w_t, s_t, c_t, a_t, \ell_t, k_t, \gamma_t, \theta_t, h_t)$ , where the individual components are summarized in Table 4.

**Attribute definitions.** Font family  $f_t$  selects one typeface from a finite catalog (e.g., serif, sans-serif, script, display), while weight  $w_t$  chooses among nine standard levels (100–900). The size attribute  $s_t$  is discretized into 16 bins that approximately cover typical point sizes for titles, subtitles and body text. Color  $c_t$  is represented in continuous RGB/HSV space and is the only non-categorical attribute, which allows precise control of hue and contrast for WCAG-aware optimization. Alignment  $a_t$  specifies left/center/right alignment of each block. Leading  $\ell_t$  and letter spacing  $k_t$  control vertical and horizontal spacing, respectively, and are both quantized into 16 levels to balance expressiveness and model complexity. Capitalization  $\gamma_t$  encodes whether the text is rendered in lower case, title case or all caps. The angle attribute  $\theta_t$  captures small rotations in a range of  $-30^\circ$  to  $30^\circ$  with 5-degree steps, which is sufficient for stylized headings while avoiding extreme distortions. Finally, hierarchy  $h_t$  indicates the semantic role of the element (e.g., main title, subtitle, body, call-to-action), and is used both for prompt conditioning and for enforcing coherent typographic structure.

**Design rationale.** The combination of discrete and continuous attributes in Table 4 is chosen to satisfy three criteria: (i) cover common design operations used by human designers (font choice, emphasis, spacing, rotation); (ii) remain compact enough to be predicted reliably from multimodal prompts and content; and (iii) expose explicit, interpretable controls that can be analyzed and manipulated in downstream applications (e.g., accessibility checks, style transfer, or human-in-the-loop editing). In the main paper we use this attribute space for both supervised training and preference-based alignment under D-DPO.

## 7. Dynamic Direct Preference Optimization (D-DPO)

We adopt Direct Preference Optimization (DPO) to align UniLayout Generator with typography-aware preferences. Let  $x$  denote a natural-language style prompt,  $c$  the structured content, and  $\pi_\theta(y \mid x, c)$  the layout policy with parameters  $\theta$ . Given a preferred layout  $y^+$  and a less preferred layout  $y^-$  under the same  $(x, c)$ , standard DPO optimizes

$$\mathcal{L}_{\text{DPO}}(\theta) = -\log \sigma\left(\beta[\Delta_\theta(x, c) - \Delta_{\text{ref}}(x, c)]\right), \quad (13)$$

Table 4. Typography attributes predicted by UniLayout Generator.

Attribute	Type	Range / Example
Font family $f_t$	Categorical	up to 256 families
Weight $w_t$	Categorical	9 levels (100–900)
Size $s_t$	Categorical	16 bins (point size)
Color $c_t$	Numeric	RGB / HSV
Alignment $a_t$	Categorical	left / center / right
Leading $\ell_t$	Categorical	16 bins
Letter spacing $k_t$	Categorical	16 bins
Capitalization $\gamma_t$	Categorical	lower / title / upper
Angle $\theta_t$	Categorical	13 bins ( $-30^\circ:5^\circ:30^\circ$ )
Hierarchy $h_t$	Categorical	4–6 levels (title $\rightarrow$ body)

where  $\Delta_\theta(x, c) = \log \pi_\theta(y^+ \mid x, c) - \log \pi_\theta(y^- \mid x, c)$  and  $\Delta_{\text{ref}}(x, c) = \log \pi_{\text{ref}}(y^+ \mid x, c) - \log \pi_{\text{ref}}(y^- \mid x, c)$ ,  $\pi_{\text{ref}}$  is a frozen reference model, and  $\beta > 0$  is a temperature.

### 7.1. Multi-signal preference scores

For each candidate layout  $y$  we compute a vector of normalized scores  $\mathbf{u}(y) \in [0, 1]^4$ :

$$u_{\text{ocr}}(y) = \text{OCR-F1}(y), \quad (14)$$

$$u_{\text{perc}}(y) = 1 - \text{LPIPS}(y), \quad (15)$$

$$u_{\text{wcag}}(y) = \text{WCAG-AA}(y)/100, \quad (16)$$

$$u_{\text{gpt}}(y) = \text{GPT-4o preference prob.}(y), \quad (17)$$

corresponding to readability, perceptual similarity, accessibility, and LLM-based (GPT-4o) aesthetic preference. For a pair  $(y^+, y^-)$  we define the per-metric gaps

$$\delta_m = u_m(y^+) - u_m(y^-), \quad m \in \{\text{ocr}, \text{perc}, \text{wcag}, \text{gpt}\}. \quad (18)$$

Only positive gaps are treated as evidence in favor of  $y^+$ :

$$g_m = \max(0, \delta_m). \quad (19)$$

### 7.2. Dynamic weighting across signals

Let  $\lambda_m > 0$  be base coefficients reflecting the nominal importance of each metric (e.g.,  $\lambda_{\text{ocr}} = \lambda_{\text{wcag}} = 1.0$ ,  $\lambda_{\text{perc}} = 0.5$ ,  $\lambda_{\text{gpt}} = 1.0$ ). For a specific pair, D-DPO computes a *dynamic* normalized weight for each signal:

$$\alpha_m = \frac{\lambda_m g_m}{\varepsilon + \sum_j \lambda_j g_j}, \quad \sum_m \alpha_m = 1, \quad (20)$$

where  $\varepsilon$  is a small constant (e.g.  $10^{-6}$ ) for numerical stability. Signals that strongly disagree between  $y^+$  and  $y^-$  (large  $g_m$ ) receive larger  $\alpha_m$ , while metrics with similar scores (small  $g_m$ ) are down-weighted.

The per-pair aggregate margin used in the DPO objective is then

$$\Delta R(x, c) = \sum_m \alpha_m \delta_m, \quad (21)$$

**Algorithm 1** Dynamic DPO for typography-aware layout

---

```

1: Input: prompt  $x$ , content  $c$ , policy  $\pi_\theta$ , reference  $\pi_{\text{ref}}$ ,
   metrics  $\{u_m\}$ , base weights  $\{\lambda_m\}$ 
2: Output: updated parameters  $\theta$ 
3: for each training step do
4:   Sample minibatch  $\{(x, c)\}$ 
5:   for each  $(x, c)$  in minibatch do
6:     Sample  $K$  candidates  $y^{(1:K)} \sim \pi_\theta(\cdot | x, c)$ 
7:     Compute  $u_m(y^{(k)})$  and  $R(y^{(k)})$ 
8:      $y^+ \leftarrow \arg \max_k R(y^{(k)})$ ,  $y^- \leftarrow \arg \min_k R(y^{(k)})$ 
9:     Compute  $\delta_m, g_m, \alpha_m, \Delta R$ 
10:    Accumulate D-DPO loss term
11:   end for
12:   Update  $\theta$  using  $\nabla_\theta \mathcal{L}$ 
13: end for

```

---

and we simply rescale the DPO temperature by this margin:

$$\mathcal{L}_{\text{D-DPO}}(\theta) = -\log \sigma\left(\beta \Delta R(x, c) [\Delta_\theta(x, c) - \Delta_{\text{ref}}(x, c)]\right). \quad (22)$$

Thus, pairs that are clearly preferred under multiple metrics ( $\Delta R$  large) induce stronger updates, while ambiguous pairs have a weaker effect.

### 7.3. Preference pair construction

For each  $(x, c)$  in the training set, we sample  $K$  candidate layouts  $y^{(1)}, \dots, y^{(K)}$  from the current policy (or a replay buffer), and evaluate the four metrics  $\mathbf{u}(y^{(k)})$ . We compute the aggregated score

$$R(y^{(k)}) = \sum_m \lambda_m u_m(y^{(k)}), \quad (23)$$

and select the best and worst candidates to form a preference pair:

$$y^+ = \arg \max_k R(y^{(k)}), \quad y^- = \arg \min_k R(y^{(k)}). \quad (24)$$

In practice, we reject pairs with  $R(y^+) - R(y^-) < \tau$  for a small margin  $\tau$  to avoid training on near-ties.

### 7.4. Pseudo-code

The overall Dynamic DPO procedure is summarized in Algorithm 1.

## 8. Qualitative Style Diversity and Failure Cases

Figure 6 illustrates the range of typography styles that *UniLayout Generator* can produce under different natural-language prompts. Each column corresponds to a fixed background image and content, while rows show typography-only edits driven by style descriptions such as

“soft pastel script for babies”, “clean clinical sans serif”, or “cinematic blockbuster title”.

For the baby formula and baby bottle posters (first two rows), the system varies font family, weight, color, and hierarchy while preserving the underlying illustration and spatial arrangement. This results in a spectrum of styles, from rounded friendly headings to more minimal, high-contrast sans serif layouts, demonstrating that the attribute space in Table 4 can support diverse yet design-consistent outputs.

The projector advertisements (bottom row) highlight a different aspect of the model’s behavior. Here we intentionally use prompts that ask for strong emphasis and multiple type styles (e.g., “cinematic title with bold tagline and highlighted feature list”). Although the generated text remains readable and aligned with the product semantics, some variants combine several font families and accent colors within a small region, leading to visually cluttered typography. These examples illustrate both the flexibility of the framework and a typical failure mode where expressive prompts can push the system toward overly busy designs.



Figure 6. Qualitative typography editing results on three product categories (baby formula posters, baby bottles, and home projectors). For each background image, UniLayout Generator produces diverse style variants while keeping the non-text content fixed (top two rows). The bottom row shows more aggressive prompts that mix multiple font families, weights and emphases in a small area, illustrating that our method can sometimes generate visually cluttered typography despite preserving legibility.