# Supplementary Material

## Contents


**A Additional Results of HG-DPO** **1**
A.1. Text-to-Image Generation . . . . . 1
A.2 Personalized Text-to-Image Generation . . . 1

**B Additional Analysis on the Easy Stage** **3**
B.1. Effectiveness of the Easy Stage . . . . . 3
B.2. Image Pool with AI Feedback . . . . . 3
B.3. Statistics Matching Loss . . . . . 4
B.3.1. Hypothesis test . . . . . 4
B.3.2. What causes the color shift artifacts? 4
B.3.3. Why is it sufficient to match only the mean? . . . . . 4
B.3.4. Importance of the statistics matching loss . . . . . 5

**C Additional Analysis on the Normal Stage** **5**
C.1. Effectiveness of the Normal Stage . . . . . 5
C.2. Intermediate Domains . . . . . 6
C.2.1. Intermediate domains with SDRecon 6
C.2.2. Winning images from the intermediate domains . . . . . 6

**D Additional Analysis on the Hard Stage** **8**
D.1. Effectiveness of the Hard Stage . . . . . 8
D.2 Winning Images of the Hard Stage . . . . . 8
D.3. Effectiveness of the Enhanced Text Encoder . 8

**E Limitations** **8**

**F. Implementation Details** **8**
F.1. Details on Supervised Fine-Tuning . . . . . 8
F.2. Details on HG-DPO Training . . . . . 15
F.2.1 . Architecture . . . . . 15
F.2.2 . Loss function . . . . . 15
F.2.3 . Optimization . . . . . 15
F.2.4 . Dataset . . . . . 16
F.3. Adaptation to Personalized T2I model . . . . 16
F.4. Details on Image Sampling . . . . . 16
F.5. Details on User Study . . . . . 16

**G Broader Impacts** **16**


## A. Additional Results of HG-DPO

In this section, we present additional qualitative and quantitative results of HG-DPO to demonstrate the effectiveness of HG-DPO.

### A.1. Text-to-Image Generation

As demonstrated in Figure S1, HG-DPO successfully generates high-quality human images with diverse actions, appearances, group sizes, and backgrounds. This is made possible by HG-DPO's effective enhancement of the base model, as demonstrated by extensive experimental results in our manuscript and Figure S2.

As a result, in Table S1, HG-DPO outperforms other existing methods. Table S1 is similar to Table 1 in the manuscript but differs in two key aspects: it includes additional baselines, DPOK [3] and D3PO [20], and uses 10 random seeds instead of a single one. To train DPOK and D3PO, we use our training prompt set $\mathcal{P}$ and PickScore [10] as the reward model. While D3PO originally uses human feedback, we follow the authors' setup by using the reward model instead. The results in Table S2, which converts Table S1 to samplewise win rates, further highlight the effectiveness of HG-DPO.

Furthermore, HG-DPO significantly outperforms the base model and the previous approaches in the user study, as shown in Figure S3. In the user study, we evaluated a selected subset of the baselines introduced in Section 4 of our manuscript against HG-DPO. Specifically, since the model trained with HPD [19] yields results similar to the model trained with Pick-a-Pic [10] (see Figure 4 in our manuscript), we compared HG-DPO exclusively with the model trained using Pick-a-Pic [10], which is widely used in DPO-related studies. Furthermore, we excluded Diffusion-DPO [17], NCP-DPO [4], and MAPO [7] from the user study because these models often failed to generate images reliably and exhibited severe artifacts (see Figure 4 in our manuscript).

### A.2. Personalized Text-to-Image Generation

HG-DPO significantly improves personalized text-to-image (PT2I) generation. As shown in Figure S4, this allows the generation of high-quality images that accurately reflect specific identities. Notably, these improvements are

Figure S1. **Qualitative results of HG-DPO.** HG-DPO is capable of effectively generating high-quality human images that encompass a wide range of actions, appearances, group sizes, and backgrounds.

| Model | P-Score (↑) | HPS (↑) | I-Reward (↑) | AES (↑) | CLIP (↑) | FID (↓) | CI-Q (↑) | CI-S (↑) | ATHEC (↑) |
|---|---|---|---|---|---|---|---|---|---|
| HPD v2 | 21.7211 | 0.2821 | -0.1353 | 6.0928 | 29.71 | 39.53 | 0.8856 | 0.9507 | 17.45 |
| Pick-a-Pic v2 | 21.6778 | 0.2821 | -0.1352 | 6.0999 | 29.72 | 40.85 | 0.8614 | 0.9383 | 17.43 |
| Diffusion-DPO | 18.0731 | 0.2408 | -1.9616 | 5.0637 | 23.49 | 160.11 | 0.6638 | 0.8715 | **40.31** |
| NCP-DPO | 17.4631 | 0.2327 | -2.0222 | 4.7983 | 21.53 | 184.81 | 0.6342 | 0.8236 | 12.09 |
| MAPO | 20.3971 | 0.2692 | -0.5150 | 5.4260 | 28.22 | 63.33 | 0.6459 | 0.7566 | 30.71 |
| Curriculum-DPO | 22.4298 | 0.2868 | 0.5823 | 6.1874 | 31.43 | 37.02 | 0.8857 | 0.9528 | 21.63 |
| AlignProp | **22.8933** | 0.2843 | 0.0693 | **6.2670** | 29.50 | 53.87 | 0.8534 | 0.9609 | 15.67 |
| DPOK | 21.6709 | 0.2809 | -0.2344 | 6.0998 | 29.25 | 41.52 | 0.8756 | 0.9332 | 15.68 |
| D3PO | 21.6905 | 0.2810 | -0.1914 | 6.0764 | 29.59 | 41.26 | 0.8902 | 0.9508 | 17.41 |
| HG-DPO (Ours) | 22.5781 | **0.2871** | **0.7384** | 6.1758 | **31.53** | **30.91** | **0.9327** | **0.9852** | 28.28 |

Table S1. **Quantitative comparison with the previous methods.** HG-DPO achieves superior performance over the existing methods across nearly all evaluation metrics. **Bold** text and underlined text indicate the best and second-best results, respectively. The row corresponding to our final model, HG-DPO, is highlighted in blue. For a more accurate comparison, we evaluate using 10 random seeds.

| Model | P-Score (↑) | HPS (↑) | I-Reward (↑) | AES (↑) | CLIP (↑) | CI-Q (↑) | CI-S (↑) | ATHEC (↑) |
|---|---|---|---|---|---|---|---|---|
| vs HPD v2 | 85.13 % | 76.44 % | 82.25 % | 62.17 % | 73.15 % | 82.31 % | 86.64 % | 93.75 % |
| vs Pick-a-Pic v2 | 86.03 % | 76.14 % | 82.08 % | 61.06 % | 72.64 % | 89.63 % | 90.87 % | 93.79 % |
| vs Diffusion-DPO | 99.97 % | 99.96 % | 99.67 % | 96.91 % | 96.48 % | 96.74 % | 88.17 % | 27.24 % |
| vs NCP-DPO | 99.97 % | 99.85 % | 99.78 % | 95.04 % | 98.95 % | 99.61 % | 96.94 % | 97.02 % |
| vs MAPO | 97.92 % | 98.35 % | 88.51 % | 96.26 % | 84.78 % | 98.89 % | 98.10 % | 42.07 % |
| vs Curriculum-DPO | 60.85 % | 51.86 % | 57.10 % | 49.51 % | 50.21 % | 84.35 % | 88.66 % | 82.74 % |
| vs AlignProp | 33.82 % | 62.12 % | 75.80 % | 37.73 % | 74.02 % | 95.35 % | 85.45 % | 97.98 % |
| vs DPOK | 86.19 % | 80.71 % | 84.06 % | 61.80 % | 77.67 % | 85.67 % | 91.82 % | 95.91 % |
| vs D3PO | 85.90 % | 81.70 % | 83.69 % | 64.16 % | 74.82 % | 81.09 % | 87.46 % | 92.90 % |

Table S2. **Samplewise win rates (%) of HG-DPO against the previous methods.** HG-DPO achieves superior performance over the existing methods across nearly all evaluation metrics. This table converts Table S1 into win rates, which means that these results are also calculated using 10 random seeds.

achieved without compromising the identity injection capability of the base PT2I model.

## B. Additional Analysis on the Easy Stage

In this section, we present additional experimental results and further analysis of the easy stage.

### B.1. Effectiveness of the Easy Stage

In the easy stage, we refine the base model to generate images that align more closely with human preferences as shown in Figure S5. Specifically, the model is improved to produce images with undistorted poses and anatomies and stronger alignment with the given prompts.

### B.2. Image Pool with AI Feedback

In our manuscript, we propose a method for selecting winning and losing images from the image pool using AI feedback (PickScore [10]). This method assumes that a larger PickScore difference between the winning and losing images indicates greater semantic differences, which are crucial for enhancing the model through DPO and align better with actual human preferences. As shown in Figure S6,


Figure S2. **Qualitative enhancements in text-to-image generation through HG-DPO.** HG-DPO improves the base model's capability to generate human images with more realistic poses and anatomies that align more accurately with the given prompt.

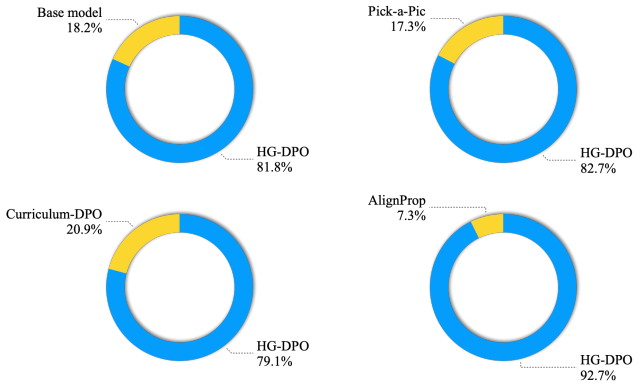



Figure S3. **User studies comparing HG-DPO and baselines.** HG-DPO demonstrates superior performance compared to the base model and previous approaches in human evaluation. Participants were tasked with choosing the image that exhibited higher realism and better alignment with the given prompt from the outputs of the two models. The detailed process for conducting the user study is described in Section F.5.

comparing the image with the highest PickScore to the image with the $l$-th highest PickScore reveals that the semantic differences between the two images (e.g., anatomy, pose, and text-image alignment) become more pronounced as $l$ increases. By choosing the images with the highest and the 20th highest PickScores as the winning and losing images, respectively, we accentuate the semantic differences between them, better reflecting human preferences.

## B.3. Statistics Matching Loss

In this section, we further analyze the statistics matching loss.

### B.3.1. Hypothesis test

Here, we validate the hypothesis underlying the statistics matching loss, $\mathcal{L}_{stat}$. Let us denote the model obtained by training $\epsilon_{base}$ through the easy stage without $\mathcal{L}_{stat}$ as $\hat{\epsilon_{\mathbb{E}}}$. $\hat{\epsilon_{\mathbb{E}}}$ is a model that suffers from the color shift artifacts. As explained in our manuscript, we hypothesize that the cause of the color shift artifacts is the divergence between the latent statistics sampled by $\hat{\epsilon_{\mathbb{E}}}$ and those of $\epsilon_{base}$ during inference. $\mathcal{L}_{stat}$ is designed to prevent such divergence based on this assumption.

To verify our hypothesis more directly, we design an inference-time statistics matching approach called *latent adaptive normalization* (*LAN*). If the gaps in the channel-wise statistics of the latents during inference cause the color shift artifacts, then eliminating those gaps should resolve those artifacts.

Let $\hat{h}_{\mathbb{E}}^{t-1}$ and $h_{base}^{t-1}$ denote the latents sampled from the same random noise using $\hat{\epsilon_{\mathbb{E}}}$ and $\epsilon_{base}$ at inference time with timestep $t$, respectively. Formally, we define

$$\hat{h}_{\mathbb{E}}^{t-1} = \psi(h_{\mathbb{E}}^{t}, p, t, \hat{\epsilon_{\mathbb{E}}}) \tag{1}$$

$$h_{base}^{t-1} = \psi(h_{base}^{t}, p, t, \epsilon_{base}) \tag{2}$$

where $\psi$ denotes a inference-time latent sampler and $p$ denotes an inference prompt. Then, we define LAN as follows:

$$h_{\mathbb{E}}^{t-1} = \left( \frac{\hat{h}_{\mathbb{E}}^{t-1} - \mu(\hat{h}_{\mathbb{E}}^{t-1})}{\sigma(\hat{h}_{\mathbb{E}}^{t-1})} \right) \sigma(h_{base}^{t-1}) + \mu(h_{base}^{t-1}) \tag{3}$$

where $\mu$ and $\sigma$ calculate the channel-wise mean and standard deviation from the input, respectively. $h_{\mathbb{E}}^{t-1}$ is used in Eq. (1) of the supplementary material at the next inference timestep.

Table S3 reveals that $N > 2$ ($\hat{\epsilon_{\mathbb{E}}}$) + LAN significantly reduces the hue distance compared to $N > 2$ ($\hat{\epsilon_{\mathbb{E}}}$). Furthermore, $N > 2$ ($\hat{\epsilon_{\mathbb{E}}}$) + LAN achieves comparable performance to $N > 2$ ($\hat{\epsilon_{\mathbb{E}}}$) in human preference metrics (P-Score, HPS, I-Reward, and AES) and image-text alignment (CLIP). These findings validate LAN's effectiveness in addressing the color shift artifacts and support the hypothesis underlying the design of $\mathcal{L}_{stat}$.

However, because LAN requires additional sampling from $\epsilon_{base}$ during inference, it incurs higher computational costs during inference compared to $N > 2 + \mathcal{L}_{stat}$. For this reason, we propose $\mathcal{L}_{stat}$ as a more computationally efficient solution to mitigate the color shift artifacts.

### B.3.2. What causes the color shift artifacts?

The color shift artifacts arise from the deviation of the channel-wise statistics of latents sampled using $\hat{\epsilon_{\mathbb{E}}}$ from those sampled using $\epsilon_{base}$, as demonstrated by the effectiveness of LAN in the previous paragraph. Here, to find the cause of this deviation, we further analyze the winning and losing images used in the easy stage. Specifically, we calculate the cosine distance of channel-wise statistics of encoded latents of winning and losing images. In Table S4, the results reveal that the cosine distance between the latents' means for the winning and losing images is *0.2035*, while the cosine distance for their standard deviations is *0.005*. Since DPO trains the model to learn the differences between winning and losing images, it can be inferred that the differences in the channel-wise **mean** values of latents present in the dataset are also learned by the model. This can encourage the model to shift the mean of the sampled latents far from that of the losing image and close to that of the winning image.

### B.3.3. Why is it sufficient to match only the mean?

$\mathcal{L}_{stat}$ mitigates the color shift by preventing the aforementioned mean shift through the mean matching loss. Interestingly, as reported in the previous paragraph, we can observe


Figure S4. **Qualitative advancements achieved through in personalized text-to-image (PT2I) generation through HG-DPO.** HG-DPO improves the base model's capability to generate human images with more realistic poses and anatomies that align more accurately with the given prompt, and these improvements extend to PT2I generation. As a result, we can produce high-quality images that accurately reflect the identity of the concept image shown in the bottom left.

that the cosine distance of standard deviation between the latents of winning and losing images is close to zero. We believe this is why matching only the mean in $\mathcal{L}_{stat}$ is sufficient to prevent the color shift artifacts.

#### B.3.4. Importance of the statistics matching loss

As illustrated in Figure S7, the absence of $\mathcal{L}_{stat}$ results in generated images appearing unnatural due to the color shift artifacts. Incorporating $\mathcal{L}_{stat}$ effectively eliminates these artifacts, producing noticeably more natural images.

## C. Additional Analysis on the Normal Stage

In this section, we present additional experimental results and further analysis of the normal stage.

### C.1. Effectiveness of the Normal Stage

We further explore the role of the normal stage, which refines $\epsilon_{\mathbb{E}}$, derived from the easy stage, to produce $\epsilon_{\mathbb{N}}$. While the easy stage enables $\epsilon_{\mathbb{E}}$ to generate images aligned with human preferences resulting in undistorted anatomical features and poses, they still fall short of achieving the realism found in real human portrait images. For example, as shown in Figure S8, although the poses are largely free from distortion, they still appear somewhat unnatural compared to those in real photographs. The normal stage enhances $\epsilon_{\mathbb{E}}$ by improving its ability to generate compositions and poses that are not only distortion-free but also realistic, closely mirroring those found in the real dataset. Figure S8 illustrates that $\epsilon_{\mathbb{N}}$ achieves significantly more realistic composi-

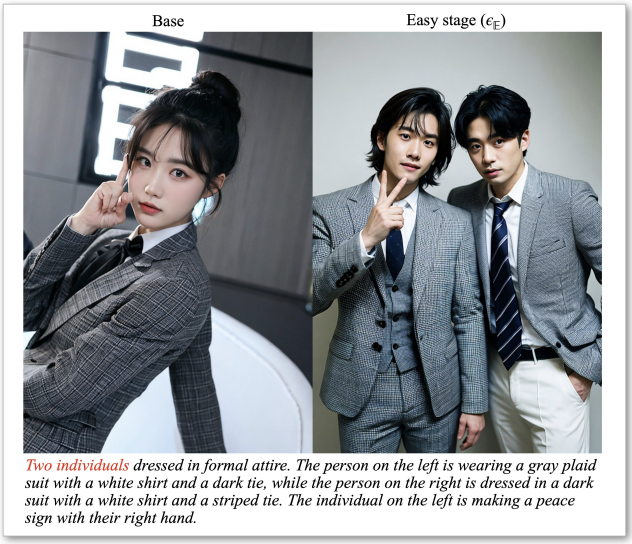



Figure S5. **Qualitative advancements achieved through the easy stage.** We enhance the base model through the easy stage to generate images that better align with human preferences. Specifically, the model is improved to produce images with undistorted poses and anatomies and stronger alignment with the given prompts.

| Model | P-Score (↑) | HPS (↑) | I-Reward (↑) | AES (↑) | CLIP (↑) | FID (↓) | CI-Q (↑) | CI-S (↑) | ATHEC (↑) | Hue (↓) |
|---|---|---|---|---|---|---|---|---|---|---|
| Base ($\epsilon_{base}$) | 21.7364 | 0.2819 | -0.0665 | 6.1061 | 29.72 | 37.34 | 0.9058 | **0.9573** | 18.73 | - |
| $N = 2$ | 22.1939 | 0.2854 | 0.3610 | 6.1408 | 30.66 | **34.44** | 0.8887 | 0.9472 | 18.96 | **10.24** |
| $N > 2$ ($\hat{\epsilon_{\mathbb{E}}}$) | 22.5688 | 0.2872 | **0.7830** | **6.2544** | 31.50 | 37.29 | 0.8879 | 0.9471 | **27.20** | 98.54 |
| $N > 2$ ($\hat{\epsilon_{\mathbb{E}}}$) + $\beta \uparrow$ | 22.2506 | 0.2864 | 0.5435 | 6.1129 | 31.30 | 36.00 | 0.8416 | 0.9141 | 19.17 | 23.77 |
| $N > 2$ ($\hat{\epsilon_{\mathbb{E}}}$) + LAN | **22.6474** | **0.2885** | 0.7677 | 6.1899 | **31.60** | 37.08 | **0.9086** | 0.9521 | 18.65 | 16.13 |
| $N > 2 + \mathcal{L}_{stat}$ ($\epsilon_{\mathbb{E}}$) | 22.5384 | 0.2878 | 0.7146 | 6.1775 | 31.56 | 36.00 | 0.9057 | 0.9547 | 19.58 | 27.94 |

Table S3. **Quantitative analysis of the easy stage.** For $\mathcal{D}_{\mathbb{E}}$, $N = 2$ generates exactly two images per prompt, while $N > 2$ builds an image pool. $N > 2 + \beta \uparrow$, $N > 2+$ LAN, and $N > 2 + \mathcal{L}_{stat}$ add regularization to address the color shift artifacts in $N > 2$. Specifically, $N > 2 + \beta \uparrow$ applies a higher $\beta$, which is a strength of the original regularization in $\mathcal{L}_{D\text{-}DPO}$, $N > 2+$ LAN applies latent adaptive normalization (Section B.3), and $N > 2 + \mathcal{L}_{stat}$ integrates $\mathcal{L}_{stat}$. **Bold** text and underlined text indicate the best and second-best results, respectively. The row corresponding to the proposed training configuration in the easy stage is highlighted in blue.

| | Mean | Standard deviation |
|---|---|---|
| Cosine distance | 0.2035 | 0.0005 |

Table S4. **Difference of channel-wise statistics between winning and losing images.** Cosine distance of channel-wise statistics of encoded latents of winning and losing images. For the encoding, we use the encoder of VAE [9] used in HG-DPO.

tions and poses, derived from real human portrait images, than $\epsilon_{\mathbb{E}}$.

## C.2. Intermediate Domains

In the normal stage, we introduce intermediate domains for winning images. Figure S9 illustrates the outcomes of the *SDRecon* operation used to create these intermediate domains, along with the winning images employed during the normal stage.

### C.2.1. Intermediate domains with SDRecon

As shown in Figure S9, we use 10 intermediate domains, labeled from $t_1$ to $t_T$. While $t_1$ is nearly identical to a real image, $t_T$ resembles a generated image, retaining little of the real image's original features. As the transition progresses from $t_1$ to $t_T$, the characteristics of the real image gradually fade, increasingly resembling those of a generated image. Specifically, fine-detail information is lost first, followed by the loss of pose information.

### C.2.2. Winning images from the intermediate domains

As depicted in Figure S9, we select four intermediate domains, $t_4$ through $t_7$, as candidates for the winning images in the normal stage. This is because the our qualitative analysis reveals that these domains generally retain the realistic pose of the real image while exhibiting fine details resembling those of generated images. Among these candidates, the image with the highest PickScore [10] is chosen as the


Figure S6. **Visualization of the image pool.** This figure shows the image pool with the size of 20 for the prompt in the leftmost column. The column labeled as 1st contains images with the highest PickScore, while the column labeled as 20th contains images with the 20th highest PickScore, i.e., the lowest PickScore, in the image pool. By selecting the image with the highest PickScore from this image pool as the winning image and the image with the 20th highest PickScore as the losing image, we magnify the semantic differences between the winning and losing images.


Figure S7. **Qualitative enhancements achieved with the statistics matching loss.** The statistics matching loss effectively removes the color shift artifacts, leading to the generation of significantly more natural images.

winning image.

# D. Additional Analysis on the Hard Stage

In this section, we present additional experimental results and further analysis of the hard stage.

## D.1. Effectiveness of the Hard Stage

We investigate the impact of the hard stage, which refines $\epsilon_{\mathbb{N}}$, obtained from the normal stage, to produce $\epsilon_{\mathbb{H}}$. While $\epsilon_{\mathbb{N}}$ achieves realistic composition and poses during the normal stage, it struggles to generate fine details. For instance, as shown in Figures S10, S11, S12, and S13, $\epsilon_{\mathbb{N}}$ 1) fails to accurately depict fine facial features such as eyes and lips, 2) requires better shading, and 3) suffers from image blurriness. Although these details may seem minor, they play a crucial role in achieving overall image realism. The hard stage addresses these limitations by enhancing $\epsilon_{\mathbb{N}}$, resulting in $\epsilon_{\mathbb{H}}$, which excels in generating realistic fine details. Figures S10, S11, S12, and S13 illustrate that $\epsilon_{\mathbb{H}}$ effectively produces fine details that $\epsilon_{\mathbb{N}}$ cannot, significantly improving image realism. As shown in Figure S14, in a user study comparing $\epsilon_{\mathbb{N}}$ and $\epsilon_{\mathbb{H}}$, $\epsilon_{\mathbb{H}}$ is rated higher, further demonstrating its effectiveness.

## D.2. Winning Images of the Hard Stage

In the hard stage, we employ images from the intermediate domain $t_1$ as winning images instead of real images. As illustrated in Figure S9, images from the intermediate domain $t_1$ are visually nearly indistinguishable from real human portrait images, making this approach effectively comparable to using real images directly as winning images. This choice is motivated by the observation that, while real images and intermediate domain $t_1$ images appear almost identical to the human eye, utilizing intermediate domain images leads to slightly better quantitative performance. Specifically, as demonstrated in Table S5, the model trained with intermediate domain $t_1$ images achieves results similar to those trained with real images, with a slight improvement in CI-Q scores.

## D.3. Effectiveness of the Enhanced Text Encoder

We train the text encoder during the easy stage to enhance image-text alignment and employ it alongside $\epsilon_{\mathbb{H}}$, derived from the hard stage, during inference. As shown in Figure S15, the enhanced text encoder effectively improves image-text alignment without compromising the image quality achieved by $\epsilon_{\mathbb{H}}$.

# E. Limitations

Through a three-stage training pipeline, HG-DPO enhances the base model to generate not only realistic anatomical features and poses but also fine details with greater realism. Despite these improvements, HG-DPO does not address the generation of realistic fingers. As shown in Figure S16, HG-DPO produces an image with sharper and more realistic fine details compared to the base model. However, the generated fingers remain notably unrealistic.

# F. Implementation Details

In this section, we provide implementation details on training and inference.

## F.1. Details on Supervised Fine-Tuning

First, we introduce the method for obtaining $\epsilon_{base}$ through supervised fine-tuning.

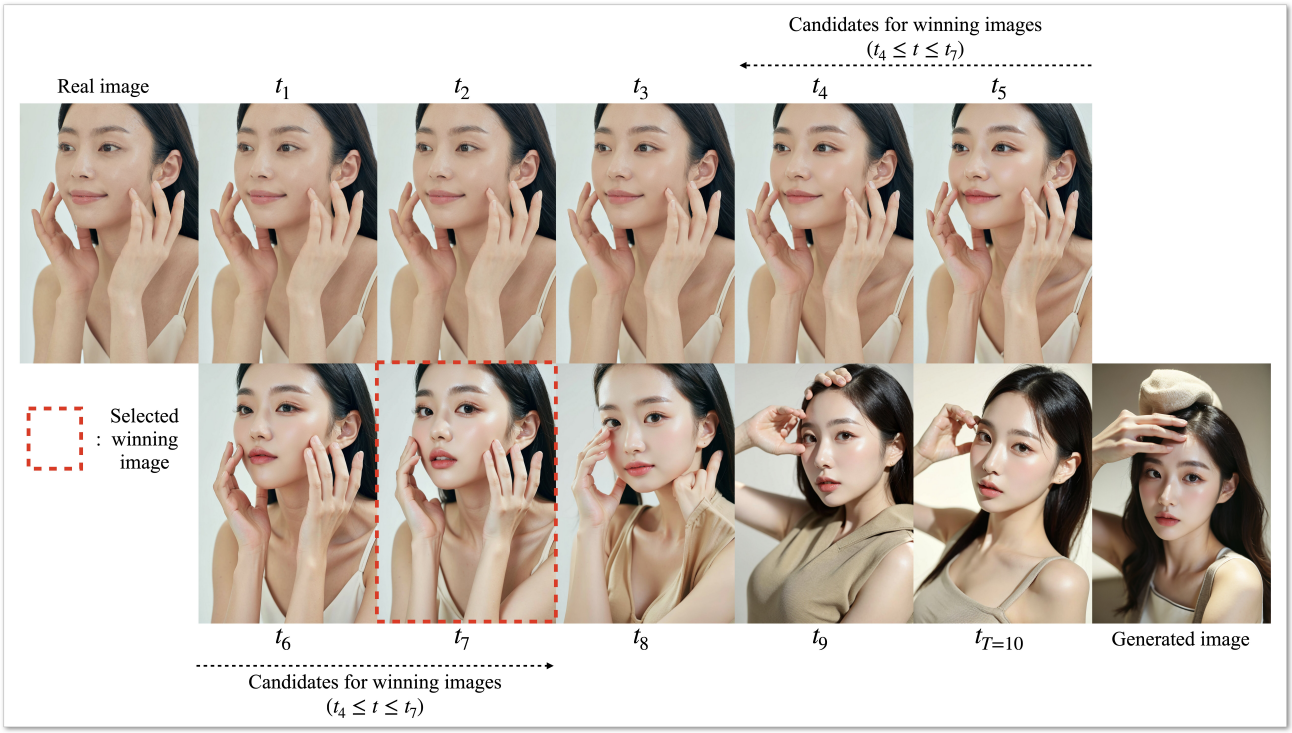
**Text-to-image dataset.** We collected approximately 300k high-quality human images. Each image has a resolution of $704 \times 1024$. We use `LLaVa` [11] to generate text prompts for all the collected images for training. This text-to-image dataset corresponds to $\mathcal{D}_{\text{real}}$ in our manuscript.


Figure S8. **Qualitative advancements achieved through the normal stage.** $\epsilon_{\mathbb{N}}$, derived by refining $\epsilon_{\mathbb{E}}$ through the normal stage, generates images with more **realistic compositions and poses** compared to $\epsilon_{\mathbb{E}}$.

| Model | P-Score (↑) | HPS (↑) | I-Reward (↑) | AES (↑) | CLIP (↑) | FID (↓) | CI-Q (↑) | CI-S (↑) | ATHEC (↑) |
|---|---|---|---|---|---|---|---|---|---|
| Real | **22.4773** | 0.2857 | 0.5388 | 6.1953 | 30.99 | **28.56** | 0.9298 | **0.9885** | 29.13 |
| Intermediate $t_1$ | 22.4698 | **0.2867** | **0.5791** | **6.1955** | **31.15** | 28.66 | **0.9365** | 0.9859 | **30.08** |

Table S5. **Quantitative results based on the type of images used as winning images in the hard stage.** The row labeled *Real* displays the results for the model trained with real images as winning images, while the row labeled *Intermediate* $t_1$ shows the results for the model trained using images from the intermediate domain $t_1$ as winning images. **Bold** text indicates the best results. The row corresponding to the proposed training configuration in the hard stage is highlighted in blue.

Furthermore, we use Qwen2-VL [18] for visual question answering to analyze distribution of this dataset, which includes 40.7% male and 59.3% female, and 24.45% child, 2.82% teenager, 41.00% youth, 31.61% adult, and 0.12% elderly. While the proportions of teenagers and elderly appear small, images in these groups may have been reasonably classified into adjacent categories (*e.g.*, teenagers as child/youth, elderly as adult).



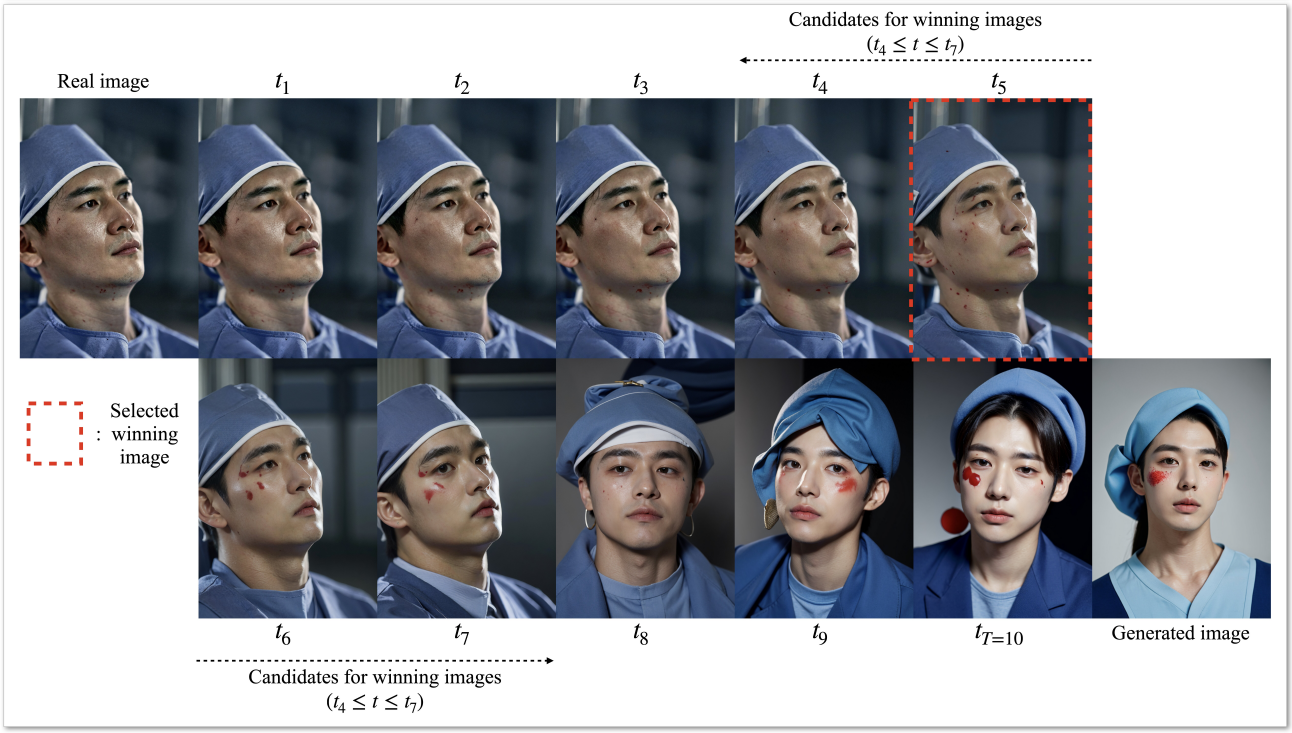




Figure S9. **Visualization of the intermediate domains.** The images labeled $t_1$ to $t_T$ are reconstructed from real images using the *SDRecon* operation. The image labeled generated image is produced via text-to-image generation based on the caption of the real image. As the labels progress toward $t_T$, SDRecon applies increasingly stronger noise to the real image, causing it to lose more of its original characteristics and resemble the generated image more closely. For the normal stage, we select four intermediate domains, $t_4$ to $t_7$, as candidates for winning images, because they maintain the realistic pose of the real image while adopting the fine details typical of the generated image. The image with the highest PickScore among these candidates is chosen as the winning image.


Figure S10. **Qualitative advancements achieved through the hard stage.** $\epsilon_{\mathbb{H}}$, derived by refining $\epsilon_{\mathbb{N}}$ through the hard stage, generates finer details, especially more realistic depictions of the **eyes**, compared to $\epsilon_{\mathbb{N}}$ as shown in the red box.


Figure S11. **Qualitative advancements achieved through the hard stage.** $\epsilon_{\mathbb{H}}$, derived by refining $\epsilon_{\mathbb{N}}$ through the hard stage, generates finer details, especially more realistic depictions of the **gaze**, compared to $\epsilon_{\mathbb{N}}$ as shown in the red box.


Figure S12. **Qualitative advancements achieved through the hard stage.** $\epsilon_{\mathbb{H}}$, derived by refining $\epsilon_{\mathbb{N}}$ through the hard stage, generates finer details, especially more realistic depictions of the **lips**, compared to $\epsilon_{\mathbb{N}}$ as shown in the red box.


Figure S13. **Qualitative advancements achieved through the hard stage.** $\epsilon_{\mathbb{H}}$, derived by refining $\epsilon_{\mathbb{N}}$ through the hard stage, generates **sharper** images with improved fine details, particularly exhibiting more vivid and realistic **shading**, compared to $\epsilon_{\mathbb{N}}$.

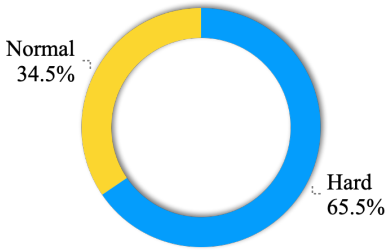



Figure S14. **User study comparing a model trained up to the normal stage ($\epsilon_{\mathbb{N}}$) with one trained through the hard stage ($\epsilon_{\mathbb{H}}$).** Participants were tasked with choosing the image that exhibited higher realism and better alignment with the given prompt from the outputs of the two models. The model trained through the hard stage achieves higher human evaluation scores due to its ability to generate finer details with greater realism compared to the model trained only up to the normal stage.

**Architecture.** We employ Stable Diffusion 1.5 (SD1.5) [13], which is pre-trained with large text-to-image datasets, as our backbone model. More specifically, we use majicmix-v7 [1], a fine-tuned model of SD1.5 specialized in human generation. We further fine-tune this backbone model with $\mathcal{D}_{\text{real}}$, to obtain our base model, $\epsilon_{base}$.

**Loss function.** For fine-tuning, we use the noise prediction loss [6]. Also, we use DDPM noise scheduler [6] for the forward diffusion process during training.

## F.2. Details on HG-DPO Training

In this section, we provide details on how to improve $\epsilon_{base}$ using HG-DPO.

### F.2.1. Architecture

**U-Net.** Instead of training the all parameters of $\epsilon_{base}$ through HG-DPO, we attach LoRA [8] layers to the all linear layers in the attention modules and only train them. We set LoRA rank as 8.

**Text encoder.** When training the text encoder, we also attach LoRA [8] layers to the all linear layers in the attention modules and only train them. For the text encoder, we set LoRA rank as 64.

### F.2.2. Loss function

**DPO loss.** We adopt the objective function of Diffusion-DPO ($\mathcal{L}_{D\text{-}DPO}$) [17] with $\beta = 2500$. For $\mathcal{L}_{D\text{-}DPO}$, we use DDPM noise scheduler [6] for the forward diffusion process.


Figure S15. **Qualitative advancements achieved through the text encoder enhancement.** By training the text encoder through the easy stage and incorporating it with $\epsilon_{\mathbb{H}}$ during inference, we achieve improved image-text alignment compared to using $\epsilon_{\mathbb{H}}$ alone. Moreover, the use of the enhanced text encoder does not compromise the image quality produced by $\epsilon_{\mathbb{H}}$.

**Statistics matching loss.** For the statistics matching loss ($\mathcal{L}_{stat}$), we set $\lambda_{stat} = 10000$. Also, for the latent sampling in $\mathcal{L}_{stat}$, we use DDPM sampler [6]. We tried DDIM sampler [15], but there was no significant difference. In addition, classifier-free guidance [5] is not used during the latent sampling in $\mathcal{L}_{stat}$.

### F.2.3. Optimization

For the optimization, we set the local batch size to four, which corresponds to the total batch size to 16 because we used four NVIDIA A100 GPUs. As an optimizer, we use the 8-bit Adam optimizer [2] with $\beta_1$ and $\beta_2$ of the Adam optimizer to 0.9 and 0.999, respectively, and the learning rate to $1e-5$. Additionally, we utilize mixed precision for


Figure S16. **Qualitative results illustrating the limitations of HG-DPO.** While HG-DPO significantly improves the base model in generating more realistic human images, it still struggles to accurately synthesize fingers.

efficient training. For the easy, normal, and hard stages, we update the model for 300k, 20k, and 20k steps, respectively.

#### F.2.4. Dataset

**Image pool.** For the image pool generation, we simply use the prompt set from $\mathcal{D}_{\text{real}}$. Furthermore, as shown in Figure S6, we generate 20 images per prompt for the image pool, which corresponds to $N = 20$ in our manuscript.

**Intermediate domains.** For the intermediate domains, we introduce 10 intermediate domains from $t_1$ to $t_{T=10}$ as shown in Figure S9. These 10 domains are generated by evenly dividing the diffusion timesteps from 1 to 1000 into 10 intervals. Specifically, we set $t_1 = 100$, $t_2 = 200$, ..., $t_T = 1000$. Then, we set $t_r = t_4$ and $t_g = t_7$ for candidates of winning images as shown in Figure S9.

### F.3. Adaptation to Personalized T2I model

To adapt HG-DPO to the personalized T2I model, we firstly trained InstantBooth [14] using $\epsilon_{base}$ as the backbone. After training InstantBooth, we can seamlessly adapt the pre-trained HG-DPO LoRA layers to InstantBooth because they share the same backbone, $\epsilon_{base}$.

### F.4. Details on Image Sampling

**Sampling method.** *DPMSolverMultistepScheduler* [12] in `diffusers` [16] is used with the step size of 50 for sampling the images, using classifier-free guidance [5] with the guidance scale of 5.0.

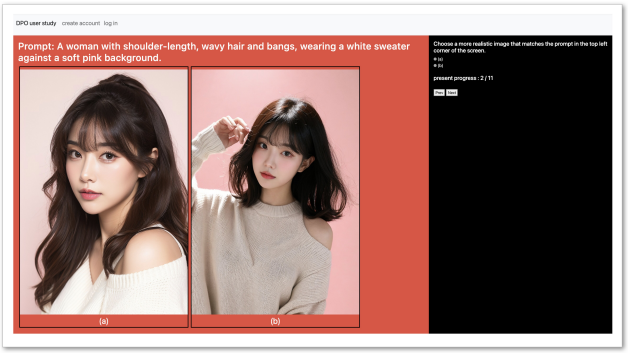



Figure S17. **User study interface.** We conduct the user study by providing a prompt and two images, asking users to choose the one that appeared more realistic considering the given prompt.

**LoRA configuration.** In addition, the LoRA weight of 0.5 is applied to both the U-Net and the text encoder. The LoRA layers in the text encoder are specifically trained to improve image-text alignment rather than visual quality, so they are applied only to a subset of inference timesteps near the noise. Specifically, the text encoder's LoRA layers are activated during inference timesteps 900 to 1000. Additionally, as $\epsilon_{\mathbb{H}}$ focuses on enhancing visual fine details, its LoRA layers are applied solely to the upsampling blocks of the U-Net, while the remaining U-Net blocks are frozen. This approach is chosen because qualitative analysis suggested that applying $\epsilon_{\mathbb{H}}$'s LoRA layers to all U-Net blocks reduces image diversity. This method allows for improved image quality while preserving diversity as much as possible.

### F.5. Details on User Study

In Figures S3 and S14, we present the results of user studies. Each participant was tasked with selecting one of two images that best aligned with the given prompt and appeared more realistic. Here, these two images are generated by the models being compared. Evaluations were conducted using a web-based user interface, as illustrated in Figure S17.

## G. Broader Impacts

We recognize the potential negative societal impacts of our work. Since our method can generate high-quality human images, it could be misused to create malicious fake images, especially when combined with personalized T2I models. It can cause significant harm to specific individuals. However, our work can also have positive impacts on society when used beneficially, such as in the entertainment or film industries. For instance, users can create desired high-quality profile pictures using text input. It highlights the beneficial uses of our work.

## References


[1] majicmix realistic. https://civitai.com/models/43331/majicmix-realistic. 15

[2] Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*, 2021. 15

[3] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023. 1

[4] Alexander Gambashidze, Anton Kulikov, Yuriy Sosnin, and Ilya Makarov. Aligning diffusion models with noise-conditioned perception. *arXiv preprint arXiv:2406.17636*, 2024. 1

[5] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 15, 16

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 15

[7] Jiwoo Hong, Sayak Paul, Noah Lee, Kashif Rasul, James Thorne, and Jongheon Jeong. Margin-aware preference optimization for aligning diffusion models without reference. *arXiv preprint arXiv:2406.06424*, 2024. 1

[8] 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. 15

[9] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6

[10] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3, 6

[11] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 8

[12] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022. 16

[13] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 15

[14] Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. *arXiv preprint arXiv:2304.03411*, 2023. 16

[15] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 15

[16] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 16

[17] 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. *arXiv preprint arXiv:2311.12908*, 2023. 1, 15

[18] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 9

[19] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 1

[20] Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8941–8951, 2024. 1