APPENDIX

Table of Contents

A THEORETICAL EXPLANATIONS

In this section, we provide theoretical justifications for the validity of our proposed distribution π_{HERO} in Eq. (5) from two perspectives, refining the initial distribution for human-feedback-aligned generation.

A.1 CONCENTRATION OF HUMAN-SELECTED NOISES IN SD'S PRIOR DISTRIBUTION

It is known that the initial distribution of SD sampling is typically the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_D)$, which yields a random vector that concentrates around the sphere of radius $\sqrt{D}$ with high probability. In the following proposition, we show that a random vector drawn from our proposed distribution π_{HERO} also concentrates around the sphere of radius $\sqrt{D}$ with high probability, provided that the variance $\varepsilon_0 > 0$ of the Gaussian mixture is sufficiently small. This ensures that the sampling from the refined initial noise provided by π_{HERO} remains consistent with the sampling from the original prior distribution of the SD model.

Proposition A.1 (Concentration of π_{HERO})**.** *Let π be a Gaussian mixture with each component as $\mathcal{N}(\boldsymbol{\mu}_i, \varepsilon_0^2\mathbf{I}_D)$, where each mean $\boldsymbol{\mu}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$, and $\varepsilon_0 > 0$ is a small constant. Let $\mathbf{y} \sim \pi$ be a random vector drawn from π. Then, for any $\delta > 0$, we have the following concentration if ε_0 is sufficiently small:*

$$\mathbb{P}\left(\sqrt{D}(1-\varepsilon_0) \leqslant \|\mathbf{y}\| \leqslant \sqrt{D}(1+\varepsilon_0)\right) \geqslant 1-\delta.$$

Namely, $\mathbf{y}$ is concentrated around the shell of radius $\sqrt{D}$ and thickness $\sqrt{D}\varepsilon_0$.

Proof. We will show that the overall probability mass is concentrated in a shell around radius $\sqrt{D}$, which means that for a sample $\mathbf{y}$ from the GMM π, $\|\mathbf{y}\| \approx \sqrt{D}$ with high probability.

From the properties of high-dimensional Gaussians (Vershynin, 2018), we know that the norm of each mean $\boldsymbol{\mu}_i$ concentrates around $\sqrt{D}$. Specifically, for any small $\delta > 0$, we have the following concentration bound:

$$\mathbb{P}\left(\sqrt{D}(1-\delta) \leqslant \|\boldsymbol{\mu}_i\| \leqslant \sqrt{D}(1+\delta)\right) \geqslant 1 - 2\exp\left(-\frac{\delta^2 D}{8}\right) \tag{6}$$

This means that the means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_n$ are likely to lie within a thin shell of radius $\sqrt{D}$ and width proportional to $\delta\sqrt{D}$.

Now consider the Gaussian component corresponding to $\boldsymbol{\mu}_i$, which is distributed as $\mathcal{N}(\boldsymbol{\mu}_i, \varepsilon_0^2 \mathbf{I}_D)$. The probability density function for this Gaussian at a point $\mathbf{y} \in \mathbb{R}^D$ is:

$$p_i(\mathbf{y}) = \frac{1}{(2\pi\varepsilon_0^2)^{D/2}} \exp\left(-\frac{\|\mathbf{y} - \boldsymbol{\mu}_i\|^2}{2\varepsilon_0^2}\right)$$

We need to analyze the concentration of this Gaussian around $\boldsymbol{\mu}_i$. The squared distance $\|\mathbf{y} - \boldsymbol{\mu}_i\|^2$ follows a chi-squared distribution with D degrees of freedom, scaled by ε_0^2. Specifically, for any $\delta > 0$, using a concentration inequality (e.g., Chernoff's bound), we can show that:

$$\mathbb{P}\left(\left|\|\mathbf{y} - \boldsymbol{\mu}_i\|^2 - D\varepsilon_0^2\right| \geqslant \delta D\varepsilon_0^2\right) \leqslant 2\exp\left(-\frac{\delta^2 D}{8}\right)$$

This implies that $\|\mathbf{y} - \boldsymbol{\mu}_i\|$ is concentrated around $\varepsilon_0\sqrt{D}$ with high probability. For small ε_0, the samples from the Gaussian will be tightly concentrated around $\boldsymbol{\mu}_i$, and the typical distance from $\boldsymbol{\mu}_i$ will be approximately $\varepsilon_0\sqrt{D}$.

Next, we want to understand the behavior of $\|\mathbf{y}\|$, where $\mathbf{y}$ is a sample from the GMM π. Since $\mathbf{y}$ is a sample from one of the Gaussian components, say $\mathcal{N}(\boldsymbol{\mu}_i, \varepsilon_0^2 \mathbf{I}_D)$, we have:

$$\mathbf{y} = \boldsymbol{\mu}_i + \mathbf{z}, \quad \text{where } \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \varepsilon_0^2 \mathbf{I}_D).$$

We analyze the expression

$$\|\mathbf{y}\|^2 = \|\boldsymbol{\mu}_i + \mathbf{z}\|^2 = \|\boldsymbol{\mu}_i\|^2 + 2\langle \boldsymbol{\mu}_i, \mathbf{z}\rangle + \|\mathbf{z}\|^2$$

term by term.

For $\|\boldsymbol{\mu}_i\|^2$ term, we know from Ineq. (6) that $\|\boldsymbol{\mu}_i\|^2$ concentrates around D, meaning:

$$\|\boldsymbol{\mu}_i\|^2 = D(1 + \mathcal{O}(\delta)).$$

For the cross term $\langle \boldsymbol{\mu}_i, \mathbf{z}\rangle$ term, since $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \varepsilon_0^2 \mathbf{I}_D)$ and $\boldsymbol{\mu}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$, we have that $\langle \boldsymbol{\mu}_i, \mathbf{z}\rangle$ is a sum of independent normal random variables with mean 0 and variance ε_0^2. Hence, $\langle \boldsymbol{\mu}_i, \mathbf{z}\rangle \sim \mathcal{N}(\mathbf{0}, \varepsilon_0^2 D)$, and we can apply a concentration inequality (e.g., Hoeffding's inequality) to show that:

$$\mathbb{P}\left(|\langle \boldsymbol{\mu}_i, \mathbf{z}\rangle| \geqslant t\right) \leqslant 2\exp\left(-\frac{t^2}{2\varepsilon_0^2 D}\right).$$

Therefore, with high probability, the cross term is small:

$$\langle \boldsymbol{\mu}_i, \mathbf{z}\rangle = \mathcal{O}(\varepsilon_0\sqrt{D}).$$

For $\|\mathbf{z}\|^2$ term, it is the squared norm of a Gaussian random vector with covariance $\varepsilon_0^2 \mathbf{I}_D$, and hence follows a chi-squared distribution with D degrees of freedom, scaled by ε_0^2. We know that:

$$\mathbb{E}[\|\mathbf{z}\|^2] = D\varepsilon_0^2, \quad \mathrm{Var}[\|\mathbf{z}\|^2] = 2D\varepsilon_0^4$$

Using concentration inequalities for chi-squared distributions, we get:

$$\mathbb{P}\left(\left|\|\mathbf{z}\|^2 - D\varepsilon_0^2\right| \geqslant \delta D\varepsilon_0^2\right) \leqslant 2\exp\left(-\frac{\delta^2 D}{8}\right)$$

Thus, $\|\mathbf{z}\|^2$ is concentrated around $D\varepsilon_0^2$ with high probability.

Combining these terms:

$$\|\mathbf{y}\|^2 = \|\boldsymbol{\mu}_i\|^2 + 2\langle \boldsymbol{\mu}_i, \mathbf{z}\rangle + \|\mathbf{z}\|^2$$

we have:

$$\begin{aligned}\|\mathbf{y}\|^2 &= D(1+\mathcal{O}(\delta)) + \mathcal{O}(\varepsilon_0\sqrt{D}) + D\varepsilon_0^2(1+\mathcal{O}(\delta)) \\ &= D(1+\varepsilon_0^2) + \mathcal{O}\big(D(1+\varepsilon_0^2)\delta\big) + \mathcal{O}(\varepsilon_0\sqrt{D}).\end{aligned}$$

Therefore, whenever ε_0 is sufficiently small, this shows that $\|\mathbf{y}\| \approx \sqrt{D}$ with high probability.

□

A.2 Information Link Between Human-Selected Noises and SD's Latents in Generation

We consider the general form of the backward SDE for diffusion model sampling (Song et al., 2020b; Lai et al., 2023a;b):

$$d\mathbf{z}_t = \big(f(t)\mathbf{z}_t - g^2(t)\nabla \log p_t(\mathbf{z}_t)\big)\,dt + g(t)d\bar{\mathbf{w}}_t, \quad \mathbf{z}_T \sim \pi_{\text{HERO}}, \tag{7}$$

where $f\colon \mathbb{R} \to \mathbb{R}$ is the drift scaling term, $g\colon \mathbb{R} \to \mathbb{R}_{\geqslant 0}$ is the diffusion term determined by the forward diffusion process, and $\bar{\mathbf{w}}_t$ represents the time-reversed Wiener process.

In the following proposition, we demonstrate that if $\Delta t \not\approx 0$, then the initial condition $\mathbf{z}_T \sim \pi_{\text{HERO}}$ and the solution $\mathbf{z}_0$ obtained from a finite-step numerical solver will possess mutual information. This suggests that the information of either $\mathbf{z}_0$ or $\mathbf{z}_T$ is preserved during SDE solving with common forward designs, such as the variance-preserving SDE (Ho et al., 2020; Song et al., 2020b) in SD. Typical choices include the Ornstein–Uhlenbeck process $\big(f(t), g(t)\big) = (-1, \sqrt{2})$, or $\big(f(t), g(t)\big) = \Big(-\frac{1}{2}\beta(t), \sqrt{\beta(t)}\Big)$, where $\beta(t) := \beta_{\min} + t(\beta_{\max} - \beta_{\min})$, with $\beta_{\min} = 0.1$ and $\beta_{\max} = 20$.

We consider discretized time using a uniform partition (Kim et al., 2024a; Hu, 1996; Kim et al., 2024b) $0 = t_n < t_{n-1} < \ldots < t_0 = T$ with $\Delta t = t_{k+1} - t_k$ for our analysis. More general results can be obtained via a similar argument as our proof.

Proposition A.2 (Information Link Between $\mathbf{z}_T$ and Generated $\mathbf{z}_0$)**.** *Let $\mathbf{z}_T \sim \pi_{\text{HERO}}$. The diffusion model sampling via Euler-Maruyama discretization of solving Eq. (7) with uniform stepsize Δt will lead to the following form:*

$$\mathbf{z}_0 = \mathbf{z}_T e^{\sum_{k=0}^{n-1} f(t_k)\Delta t} - \sum_{k=0}^{n-1} g^2(t_k)\nabla \log p_{t_k}(\mathbf{y}_k)\Delta t e^{\sum_{j=k+1}^{n-1} f(t_j)\Delta t} + R(\Delta t),$$

where $R(\Delta t)$ is the residual term concerning the accumulated stochastic component $g(t_n)\Delta\bar{\mathbf{w}}_n$ and stepsize Δt. Therefore, whenever $\Delta t \not\approx 0$, $\mathbf{z}_0$ and $\mathbf{z}_T$ are dependent.

Proof. For the simplicity of notations, we write $\mathbf{y}_n := \mathbf{z}_{t_n}$ (i.e., $\mathbf{y}_0 = \mathbf{z}_T$). Applying the Euler-Maruyama scheme, we obtain:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \big(f(t_n)\mathbf{y}_n - g^2(t_n)\nabla \log p_{t_n}(\mathbf{y}_n)\big)\,\Delta t + g(t_n)\Delta\bar{\mathbf{w}}_n,$$

where $\mathbf{y}_0 \sim \pi_{\text{HERO}}$, and $\Delta\bar{\mathbf{w}}_n \sim \mathcal{N}(\mathbf{0}, \Delta t\mathbf{I})$ represents the increment of the Wiener process.

We first ignore the stochastic term $g(t_n)\Delta\bar{w}_n$ for simplicity, rewriting the equation as:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \big(f(t_n)\mathbf{y}_n - g^2(t_n)\nabla \log p_{t_n}(\mathbf{y}_n)\big)\,\Delta t.$$

This can be rearranged into:

$$\mathbf{y}_{n+1} = \mathbf{y}_n(1 + f(t_n)\Delta t) - g^2(t_n)\nabla \log p_{t_n}(\mathbf{y}_n)\Delta t.$$

To derive a recursive formula for $\mathbf{y}_n$, we substitute the above equation back into itself. Starting from $\mathbf{y}_0$:

$$\begin{aligned}\mathbf{y}_1 &= \mathbf{y}_0(1 + f(t_0)\Delta t) - g^2(t_0)\nabla \log p_{t_0}(\mathbf{y}_0)\Delta t, \\ \mathbf{y}_2 &= \mathbf{y}_1(1 + f(t_1)\Delta t) - g^2(t_1)\nabla \log p_{t_1}(\mathbf{y}_1)\Delta t.\end{aligned}$$

By continuing this process, we express $\mathbf{y}_n$ recursively as:

$$\mathbf{y}_n = \mathbf{y}_{n-1}(1 + f(t_{n-1})\Delta t) - g^2(t_{n-1})\nabla \log p_{t_{n-1}}(\mathbf{y}_{n-1})\Delta t.$$

Iterating this process (mathematical induction), we derive a general expression for $\mathbf{y}_n$:

$$\mathbf{y}_n = \mathbf{y}_0 \prod_{k=0}^{n-1}(1 + f(t_k)\Delta t) - \sum_{k=0}^{n-1} g^2(t_k)\nabla \log p_{t_k}(\mathbf{y}_k)\Delta t \prod_{j=k+1}^{n-1}(1 + f(t_j)\Delta t).$$

We can utilize the exponential Taylor expansion

$$e^{f(t)\Delta t} = (1 + f(t)\Delta t) + \mathcal{O}((\Delta t)^2).$$

to reduce the above expression to:

$$\mathbf{y}_n = \mathbf{y}_0 e^{\sum_{k=0}^{n-1} f(t_k)\Delta t} - \sum_{k=0}^{n-1} g^2(t_k)\nabla \log p_{t_k}(\mathbf{y}_k)\Delta t e^{\sum_{j=k+1}^{n-1} f(t_j)\Delta t} + \mathcal{O}((\Delta t)^2)$$

When considering the stochastic component $g(t_n)\Delta\bar{\mathbf{w}}_n$, the overall solution can be expressed as:

$$\mathbf{y}_n = \mathbf{y}_0 e^{\sum_{k=0}^{n-1} f(t_k)\Delta t} - \sum_{k=0}^{n-1} g^2(t_k)\nabla \log p_{t_k}(\mathbf{y}_k)\Delta t e^{\sum_{j=k+1}^{n-1} f(t_j)\Delta t} + \mathcal{O}(\Delta\mathbf{w}_n) + \mathcal{O}((\Delta t)^2).$$

Therefore, the solution presented indicates that the state variable retains the memory of its initial condition for a finite time, influenced by both deterministic drift and stochastic components if $\Delta t \not\approx 0$. □

B Additional Evaluation Metrics

In this section, we present evaluation metrics beyond task success rates and supplement the results of these measurements during inference time in Appendix B.1, as well as during training in Appendix B.2.

B.1 Measurement in Inference

Results of samples from the final epoch for aesthetic quality, image diversity, and text-to-image alignment are presented in Figure 8. The descriptions of each measurement are detailed as follows.

Aesthetic Quality. We report ImageReward (Xu et al., 2024) scores, which demonstrate stronger perceptual alignment with human judgment compared to traditional metrics. Higher scores reflect better aesthetic quality. Although human evaluators prioritized task success based on the criteria in Appendix D over aesthetic quality and were not instructed to consider aesthetics, HERO demonstrates comparable aesthetic performance to the baselines, surpassing them in 3 out of 5 tasks.

Image Diversity. Following Section 4.3.3 of von Rütte et al. (2023), we compute "In-Batch Diversity", defined as the complement of the average similarity of CLIP image embeddings (Radford et al., 2021) between pairs of images in a generated batch. Specifically, for a batch of N generated images $I_1, I_2, \ldots, I_N$, and the cosine similarity $\text{CLIPSim}(I_i, I_j)$ of their embeddings in the CLIP feature space, the in-batch diversity is calculated as:

$$D_{\text{batch}} = 1 - \frac{2}{N(N-1)} \sum_{1 \leqslant i < j \leqslant N} \text{CLIPSim}(I_i, I_j),$$

where $1 - \text{CLIPSim}(I_i, I_j)$ represents the dissimilarity between two images. A higher D_{batch} signifies greater diversity. Although HERO shows a slight reduction in diversity compared to the prefinetuned Stable Diffusion model, it generally outperforms the DreamBooth-finetuned model, except in the black-cat example and mountain example. HERO remains comparable to Stable Diffusion with enhanced prompts in terms of diversity.

Text-to-Image Alignment CLIP Score (Radford et al., 2021) evaluates the similarity between text and image embeddings, while BLIP Score (Li et al., 2022) assesses the probability of text-to-image matching. Together, these metrics provide a quantitative measure of how well the generated images align with the given prompts. Higher scores on both metrics indicate better alignment between the generated images and the prompts. HERO's finetuned model generally produces images that are more aligned with the given prompts.

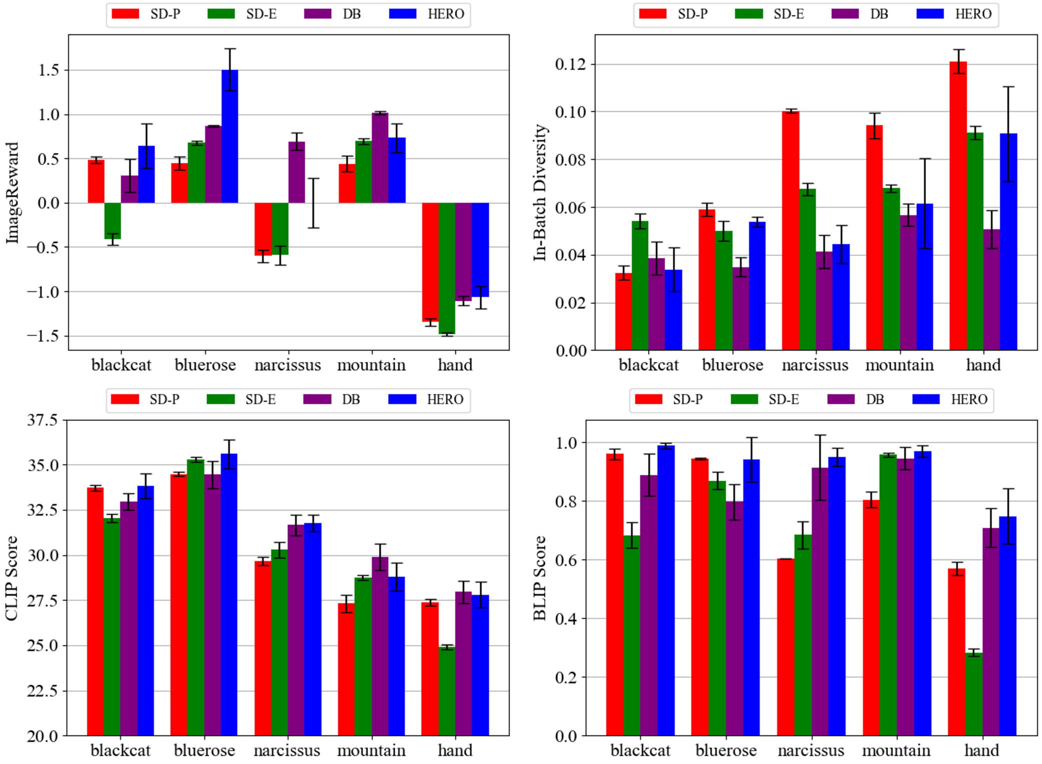
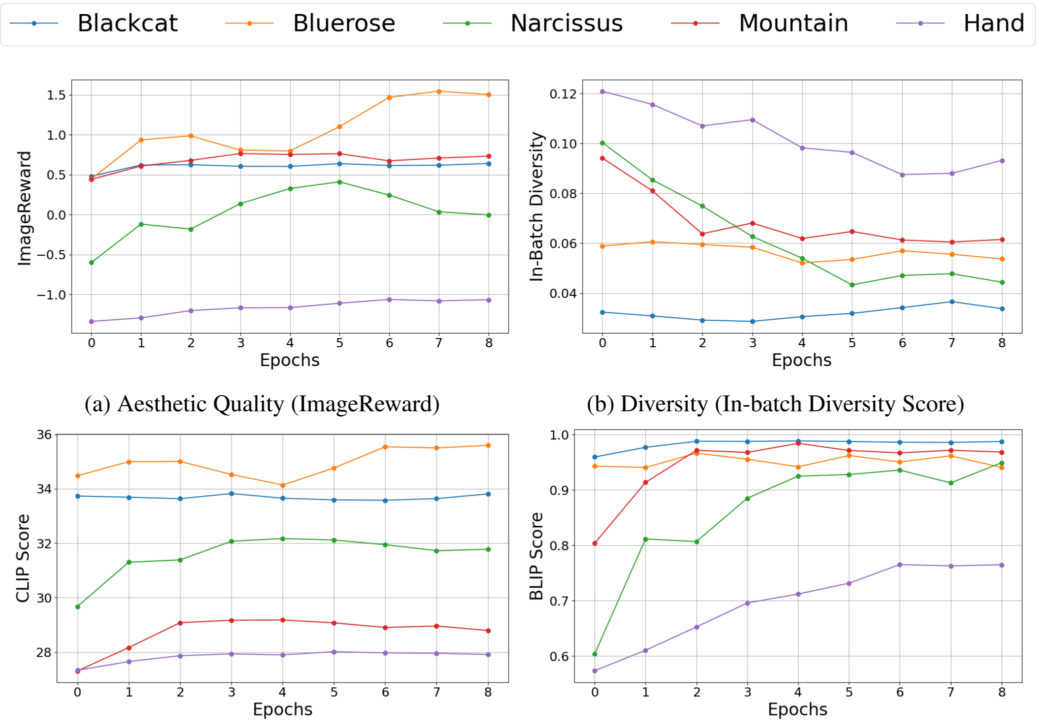


Figure 8: Additional evaluation results. For all metrics, a higher value indicates better performance. **Top Left.** Aesthetic quality measured with ImageReward (Xu et al., 2024). **Top Right.** In-Batch Diversity computation following Radford et al. (2021). **Bottom.** CLIP (Radford et al., 2021) and BLIP (Li et al., 2022) Text-to-image alignment scores.

B.2 Measurements in Training Progress

We also provide supplementary results showing different metrics versus training epochs to observe the influence of the number of feedback samples. As shown in Figure 9, we present results from samples generated during the first 8 epochs, where we observe the following trends:

- **Aesthetic Quality** (measured with ImageReward): Aesthetic quality is generally maintained throughout the fine-tuning process, demonstrating that HERO does not compromise aesthetic appeal even with increased human feedback.
- **Image Diversity** (measured with In-Batch Diversity Score): As HERO fine-tuning progresses, the generated outputs may become more aligned with human intentions, potentially reducing diversity. This aligns with the common phenomenon where stronger guidance often leads to lower diversity. Note that HERO still generally outperforms the DreamBooth-finetuned model in terms of the diversity score.
- **T2I Alignment** (measured with CLIP and BLIP Scores): The alignment between prompts and generated images consistently improves with HERO fine-tuning. This provides implicit evidence that HERO fine-tuning effectively converges toward human intention, as reflected in the prompts.



(a) Aesthetic Quality (ImageReward)

(b) Diversity (In-batch Diversity Score)

(c) T2I Alignment (CLIP Score)

(d) T2I Alignment (BLIP Score)

Figure 9: **Addtional Evaluation Measurements across Training Progress.** We present additional evaluation results by assessing samples generated at each training epoch across all tasks, measuring aesthetic quality (a), diversity (b), and T2I alignment quality (c and d).

C Additional Experiments

C.1 RL Fine-tuning with Existing Reward Models

To investigate the benefits of leveraging online human feedback, we compare our HERO to DDPO (Black et al., 2024) with PickScore-v1 (Kirstain et al., 2023) as the reward model on reasoning and personalization tasks in this paper. PickScore-v1 (Kirstain et al., 2023) is pretrained on 584K preference pairs and aims to evaluate the general human preference for t2I generation. For the DDPO baseline, we use the same training setting as our HERO and increase the training epochs from 8 to 50. The success rate is calculated using 200 evaluation images.

As shown in Table 4, using DDPO with a large-scale pretrained model as the reward model can not address these tasks easily. Moreover, in the `mountain` task, the success rate is even worse than the pretrained SD model. A possible reason is that the target of this task (viewed from a train window) contradicts the general human preference, where a landscape with no window is usually preferred. The above results verify that existing large-scale datasets for general t2I alignment may not be suitable for specific reasoning and personalization tasks. Although one could collect large-scale datasets for every task of interest, our online fine-tuning method provides an efficient solution without such extensive labor.

Table 4: Success rates of RL fine-tuning with existing reward models

Method	`blue-rose`	`black-cat`	`narcissus`	`mountain`
SD-Pretrained	0.354	0.422	0.406	0.412
DDPO + PickScore-v1	0.710	0.555	0.615	0.375
HERO (ours)	**0.807**	**0.750**	**0.912**	**0.995**

C.2 Imporve Time Efficiency for Online Finetuning

Inspired by Clark et al. (2024), we only consider the last $K + 1$ ($\leqslant T$) steps of the denoising trajectories during loss computation in Equation (2) to accelerate training and reduce the workload for human evaluators:

$$\nabla_\phi \mathcal{L}_{\text{DDPO-K}}(\phi) = \mathbb{E}_{\mathbf{z}_T \sim \mathcal{Z}_T} \sum_{t=0}^{K} \left[\frac{p_\phi(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c})}{p_{\phi_{\text{old}}}(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c})} \nabla_\phi \log p_\phi(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c}) R(\mathbf{z}_0) \right]. \tag{8}$$

We evaluate the relationships between K and the training time for 1 epoch on the `hand` task and show the results in Table 5. Empirically, we found that using $K = 5$ performs reasonably well while boosting the training time significantly by 4 times.

Table 5: The impact of update steps K on training time

K	1	2	5	10	20
Training time(s)	30.34	60.24	149.58	298.55	595.49

C.3 DreamBooth Prompting Experiments

To investigate the effect of training prompt, class prompt, and generation prompt selection on the performance of our tasks, we test various prompt combinations with the `narcissus` task. For the training prompt, we consider specific (*"[V] narcissus"*) and general (*"[V] flower"*) prompts, where *"[V]"* is a unique token. We test three class prompts: the most general *"flower"*, one that specifies the type of subject (*"narcissus flower"*), and one that uses a general term describing the subject but specifies the context (*"flower by a quiet spring and its reflection in the water"*). Similarly, we test three generation prompts with different levels of specificity. Results are shown in Table 6. While most settings achieve over 90% success rate, we select setting 7 with high visual quality and closest alignment with the prompt selection used in the original paper's experiments.

Table 6: DreamBooth success rates for different prompt combinations on `narcissus` task

	Training Prompt	Class Prompt	Generation Prompt	Success Rate
1	*"[V] narcissus"*	*"flower"*	*"[V] narcissus by a quiet spring and its reflection in the water"*	0.43
2	*"[V] narcissus"*	*"flower"*	*"[V] narcissus"*	0.94
3	*"[V] narcissus"*	*"narcissus flower"*	*"[V] narcissus"*	0.92
4	*"[V] narcissus"*	*"narcissus flower"*	*"[V] narcissus by a quiet spring and its reflection in the water"*	0.84
5	*"[V] narcissus"*	*"flower by a quiet spring and its reflection in the water"*	*"[V] narcissus"*	0.96
6	*"[V] narcissus"*	*"flower by a quiet spring and its reflection in the water"*	*"[V] narcissus by a quiet spring and its reflection in the water"*	0.91
7	*"[V] flower"*	*"flower"*	*"[V] flower"*	0.95
8	*"[V] narcissus"*	*"narcissus"*	*"[V] narcissus"*	0.92

D Details of Tasks and Task Categories

Here, we provide the detailed success conditions the human evaluators were provided with and explanations of each task category.

Detailed Task Success Conditions

- `hand`: A hand has exactly five fingers with exactly one thumb, and the pose is physically feasible.

- `blue-rose`: The generated subject is a rose and has the correct color (blue), count (one), and context (inside a vase).
- `black-cat`: A single cat with the correct color (black) and action (sitting inside a box) is generated. The cat's pose is feasible, with no parts of the body penetrating the box. The cardboard is shaped like a functional box.
- `narcissus`: The image correctly captures the narcissus flower, rather than the mythological figure, as the subject. Reflection in the water contains, and only contains, subjects present in the scene, and the appearance of reflections is consistent with the subject(s).
- `mountain`: View of the mountains is from a train window. The body of the train the mountain is seen from is not in the view. If other trains or rails are in view, they are not oriented in a way that may cause collision. Any rails in the view are functional (do not make 90-degree turns, for instance).

Description of Task Categories

- Correction: Removing distortions or defects in the generated image. For example, generating non-distorted human limbs.
- Reasoning: Capturing object attributes (e.g., color or texture), spatial relationships (e.g., on top of, next to), and non-spatial relationships (e.g., looking at, wearing).
- Counting: Generating the correct number of specified objects.
- Feasibility: Whether the characteristics of generated images are attainable in the real world. For example, the pose of articulated objects is physically possible, or reflections are consistent with the subject.
- Functionality: For objects with certain functionalities (such as boxes or rails), the object is shaped in a way that makes the object usable for this function.
- Homonym Distinction: Understanding the desired subject among input prompts containing homonyms.
- Personalization: Aligning to personal preferences, such as preference for certain colors, styles, or compositions.

E HERO Implementation

E.1 HERO Detailed Algorithm

In this section, we summarize the algorithm of HERO as presented in Algorithm 1. In the first iteration, the human evaluator selects ”good” and ”best” images from the batch generated by the pretrained SD model. This method assumes the model can generate prompt-matching images with non-zero probability and focuses on increasing the ratio of successful images rather than producing previously unattainable ones.

Algorithm 1 HERO's Training

```
Require: pretrained SD weights $\phi$, best image ratio $\beta$, feedback budget $N_{\text{fb}}$
Initialize: learnable weights $\theta$, # of feedback $n_{\text{fb}} = 0$, latent distribution $\pi_{\text{HERO}} = \mathcal{N}(\mathbf{z}_T; \mathbf{0}, \mathbf{I})$
while $n_{\text{fb}} < N_{\text{fb}}$ do
    Sample $n_{\text{batch}}$ noise latents $\mathbf{z}_T$ from $\pi_{\text{HERO}}$    ▷ Feedback-Guided Image Generation
    Perform denoising process for each $\mathbf{z}_T$ to obtain trajectory $\{\mathbf{z}_T, \mathbf{z}_{T-1}, \cdots, \mathbf{z}_0\}$.
    Decode $\mathcal{Z}_0$ with SD decoder for images $\mathcal{X}$.
    Query human feedback on $\mathcal{X}$, and save corresponding $\mathcal{Z}_T^+$, $\mathcal{Z}_T^-$, $\mathbf{z}_T^{\text{best}}$ .
    Update $\theta$ of $E_\theta$ and $g_\theta$ by minimizing Eq. (3).    ▷ Feedback-Aligned Representation Learning
    Compute reward $R(\mathbf{z}_0)$ according to Eq. (4).
    Update $\phi$ via DDPO by minimizing Eq. (8).
    Update latents distribution $\pi_{\text{HERO}}$ using Eq. (5).
    $n_{\text{fb}} \mathrel{+}= n_{\text{batch}}$.
end while
```

E.2 HERO Training Parameters

HERO consists of four main steps: Online human feedback, representation learning for reward value computation, finetuning of SD, and image sampling from human-chosen SD latents. In π_{HERO}, we choose its variance as $\varepsilon_0^2 = 0.1$ accross all experiments. Table 7 lists the parameters used in each step.

Representation learning network architecture. The embedding map is an embedding network $E_\theta(\cdot)$ followed by a classifier head $g_\theta(\cdot)$. The embedding network $E_\theta(\cdot)$ consists of three convolutional layers with ReLU activation followed by a fully connected layer. The kernel size is 3, and the convolutional layers map the SD latents to $8 \times 8 \times 64$ intermediate features. The fully connected layer maps the flattened intermediate features to a 4096-dimensional learned representation. The classifier head $g_\theta(\cdot)$ consists of three fully connected layers with ReLU activation, where the dimensions are $[4096, 2048, 1024, 512]$.

Table 7: HERO training parameters

Embedding Network $E_\theta(\cdot)$ and Classifier Head $g_\theta(\cdot)$	
Learning rate	$1e^{-5}$
Optimizer	Adam (Kingma & Ba, 2015) ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay $= 0$)
Batch size	2048
Triplet margin α	0.5
SD Finetuning	
Learning rate	$3e^{-4}$
Optimizer	Adam (Kingma & Ba, 2015) ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay $= 1e^{-4}$)
Batch size	2
Gradient accumulation steps	4
DDPO clipping parameter	1e^{-4}
Update steps for loss computation K	5
Image Sampling	
Diffusion steps	50 (20 for `hand`)
DDIM sampler parameter η	1.0
Classifier free guidance weight	5.0
Best image ratio β	0.5

F Baseline Implementations

F.1 DreamBooth Training Settings

Here, we discuss the DreamBooth (Ruiz et al., 2023) experiment design.

Input Images. Following the original DreamBooth paper that uses 3 to 5 input images, we ask human evaluators to select the top 4 best images among the initial batch of images generated for each task and use these selected images as training inputs.

Hyperparameters. We follow the common practice of training DreamBooth with LoRA (Hu et al., 2022). Training hyperparameters are listed in Table 8.

Table 8: DreamBooth training parameters

Parameters	Values
Learning rate	$1e^{-5}$
Training epochs	250
Optimizer	Adam (Kingma & Ba, 2015) ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay $= 0.01$)
Batch size	2
Prior presevation loss weight	1.0

Prior Preservation Loss (PPL). This function is enabled and uses the default setting where 100 class data images are generated from the class prompts.

Prompts. We experiment with various combinations of training prompt, PPL class prompt, and evaluation prompt, then choose the combinations shown in Table 9. See Appendix C.3 for details on prompting experiments.

The outcome of DB training is influenced by multiple factors, including the number and selection of input images, training hyperparameters, use of PPL, and combination of prompts. While we optimized these elements for our tasks to the best of our ability, it is possible that further tuning can yield better results, as the large number of tunable variables makes DB challenging to optimize.

Table 9: Training, class, and generation prompts for DreamBooth experiments

Task Name	Training Prompt	Class Prompt	Generation Prompt
`hand`	*"[V] hand"*	*"hand"*	*"[V] hand"*
`blue-rose`	*"[V] flower"*	*"flower"*	*"[V] flower"*
`black-cat`	*"[V] cat"*	*"cat"*	*"[V] cat"*
`narcissus`	*"[V] flower"*	*"flower"*	*"[V] flower"*
`mountain`	*"[V] mountains"*	*"mountains"*	*"[V] mountains"*

Table 10: Enhanced prompts used in SD-Enhanced baseline

Task Name	Generation Prompt	Enhanced Prompt
`hand`	*"1 hand"*	*"A close-up of a beautifully detailed hand with five fingers, featuring delicate and lifelike skin texture, fingers gracefully extended. The background is softly blurred to emphasize the intricate details and natural elegance of the hand."*
`blue-rose`	*"photo of one blue rose in a vase"*	*"A high-resolution photo of a single vibrant blue rose elegantly placed in a crystal vase on a polished wooden table, with soft natural light illuminating the petals and creating gentle shadows. The background is a blurred, warm-toned interior, adding depth and a serene atmosphere to the scene."*
`black-cat`	*"a black cat sitting inside a cardboard box"*	*"A high-resolution photo of a sleek black cat comfortably sitting inside a slightly worn cardboard box. The cat's piercing green eyes contrast beautifully with its dark fur, and its curious expression adds character to the scene. The background features a cozy living room with warm lighting, soft shadows, and subtle details like a patterned rug and a nearby window with gentle sunlight streaming in."*
`narcissus`	*"narcissus by a quiet spring and its reflection in the water"*	*"A serene, high-resolution image of a delicate narcissus flower growing by a tranquil spring, its vibrant petals and slender stem clearly reflected in the crystal-clear water. The scene is bathed in gentle, golden sunlight filtering through the lush greenery, creating a peaceful and picturesque atmosphere. Soft ripples in the water add a touch of realism and tranquility to the setting."*
`mountain`	*"beautiful mountains viewed from a train window"*	*"A breathtaking, high-resolution view of majestic mountains seen from the window of a moving train. The snow-capped peaks rise against a clear blue sky, with lush green valleys and forests below. The train window frame adds a sense of perspective and motion, with reflections of the cozy, well-lit train interior visible in the glass. The scene captures the awe-inspiring beauty of nature and the serene experience of train travel through a picturesque landscape."*

F.2 Prompt Enhancement with a Large VLM

In the SD-enhanced baselines, we prompt the Stable Diffusion v1.5 model with a prompt enhanced by GPT-4 (Brown, 2020; Achiam et al., 2023). To generate the enhanced prompts, we input *"Enhance the following text prompt for Stable Diffusion image generation: [prompt]"* to GPT-4 (*[prompt]* is the original task prompt labeled "Prompt" in Table 1 and "Generation Prompt" in Table 10). Output-enhanced prompts used for the SD-enhanced baseline are shown in Table 10. Although our prompt enhancement is not an exhaustive method to show the full capabilities of prompt engineering, we include SD-enhanced as a baseline to demonstrate that many of our tasks are challenging to solve, given a simple prompt enhancement method.

G Additional Elaboration of HERO's Mechanisms

In this section, we elaborate on HERO's mechanism, highlighting its cost-effective trainable embeddings and the application of contrastive learning.

About Trainable Embedding. While HERO introduces additional training for a human-aligned embedding to convert binary feedback into informative continuous reward signals, this mechanism is both efficient and effective in significantly reducing the need for online human feedback, compared to D3PO. To further illustrate the efficient training of this embedding, consider the hand deformation correction task in Figure 3. HERO requires only 1152 samples and 144 update iterations (batch size 8), compared to D3PO, which needs 5000 samples and 500 update iterations (batch size 10). Moreover, HERO's embedding map is implemented using a simple network with three CNN layers and one fully connected layer, making its training far less complex than fine-tuning Stable Diffusion.

About Trainable Embedding with Selected "Best". Below, we also provide an estimated runtime comparison. The process of selecting a single "best" image from all "good" images requires minimal extra effort from the evaluators. While providing binary "good"/"bad" labels, the evaluators are already exposed to all candidate images. With only 64 to 128 images presented at a time, evaluators typically have a general sense of which image to select as the "best" by the time they complete the binary evaluations. To provide a concrete estimate, we measured the time spent by evaluators during feedback. Evaluators spent approximately 0.5 seconds per image for binary "good"/"bad" evaluations. The time required to select the "best" image among candidates ranged from 3 to 5 seconds, depending on the number of candidates. For the upper limit of 128 candidates in our setup, the selection process took approximately 10 seconds. In terms of time, providing the "best" image label is roughly equivalent to giving feedback on 5–20 binary labels. For example, in the hand anomaly correction experiment, human evaluators provided feedback over 9 epochs with 128 feedback instances per epoch, resulting in a total of $9 \times 128 = 1152$ binary feedback labels. If we estimate the effort of "best" image feedback as $20\times$ that of binary feedback, this adds $9 \times 20 = 180$ additional feedback, for an approximate total of 1332 feedback labels. This is still significantly less than the $5000+$ feedback labels required by D3PO to achieve a comparable success rate.

About the Usage of Contrastive Learning. We emphasize the distinction in HERO's use of contrastive learning, which focuses on learning relationships among human-annotated samples through triplet loss. This differs from the contrastive learning literature (Chen et al., 2020; He et al., 2020; Caron et al., 2020), which primarily emphasizes unsupervised learning with large-scale unlabeled datasets. Specifically, HERO employs feedback-aligned representation learning by leveraging human annotations (e.g., "good", "bad", and "best") to structure embedded representations into distinct clusters using triplet loss. This approach enables efficient fine-tuning using continuous rewards derived from the similarity to the human-selected "best" samples. As a result, HERO significantly reduces the need for online human feedback, requiring only $0.5 - 1$K samples, compared to baselines such as D3PO, which require at least 5K.

H Additional Results

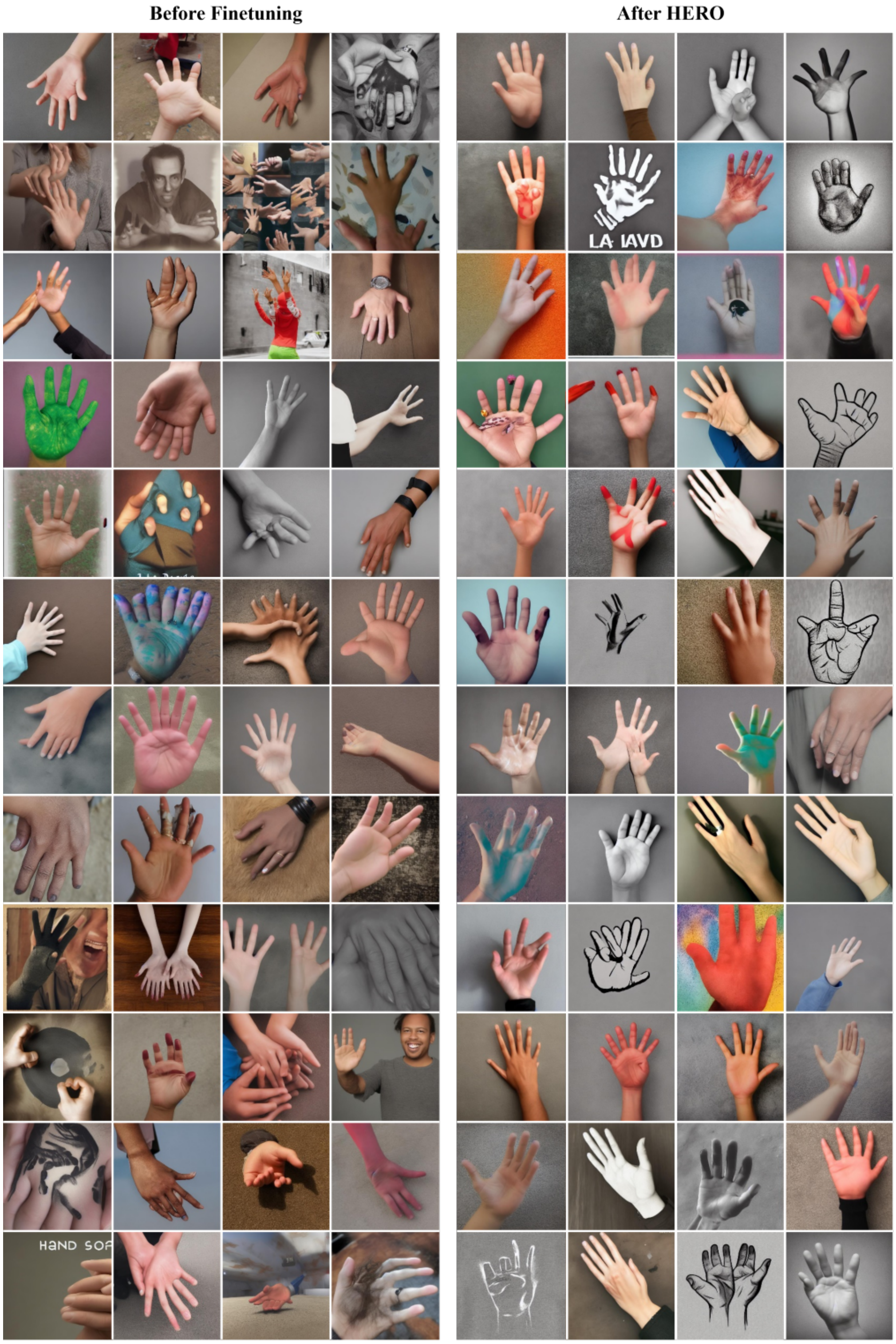


Figure 10: Randomly generated samples from pretrained SD and HERO for `hand` task.

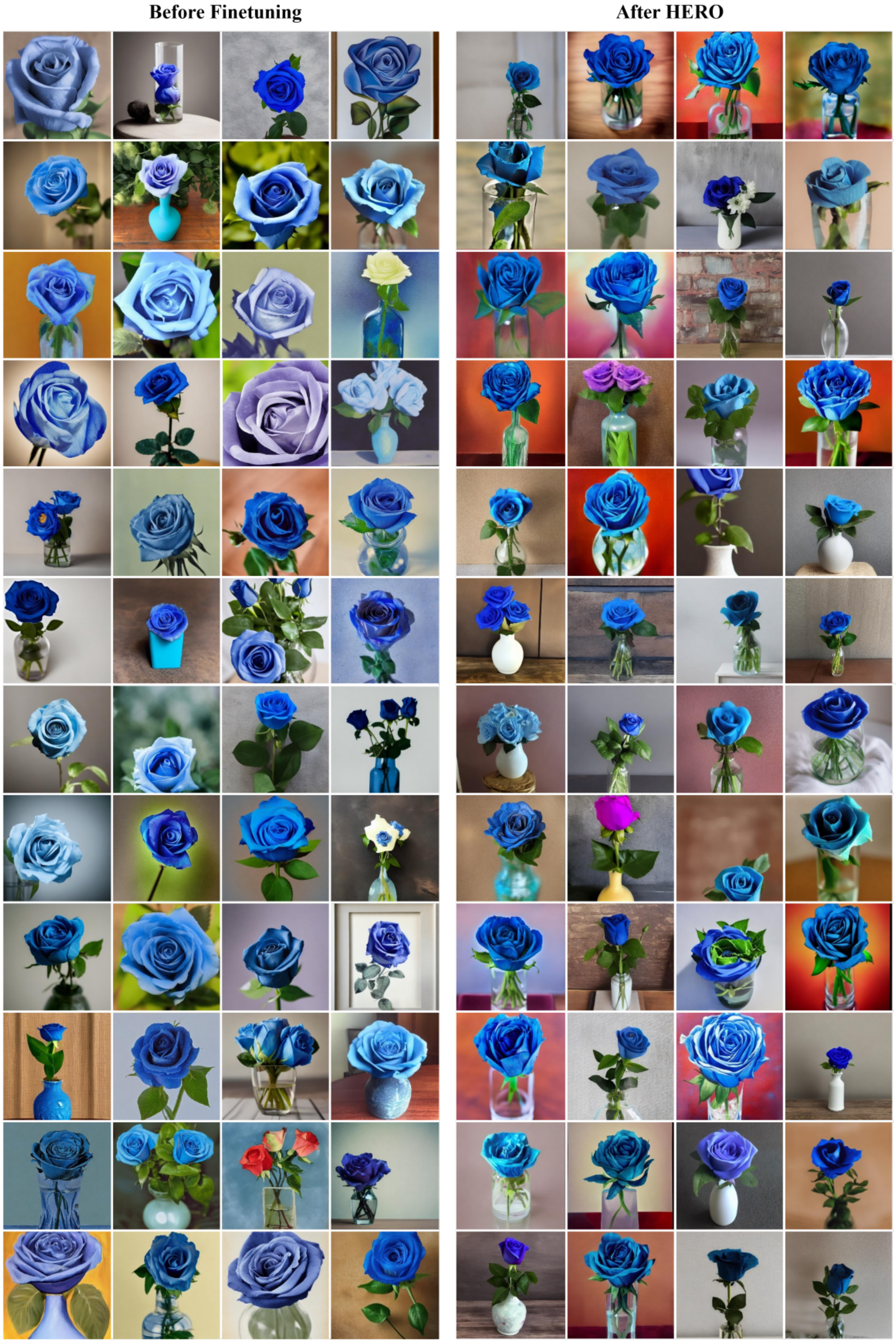


Figure 11: Randomly generated samples from pretrained SD and HERO for `blue-rose` task.

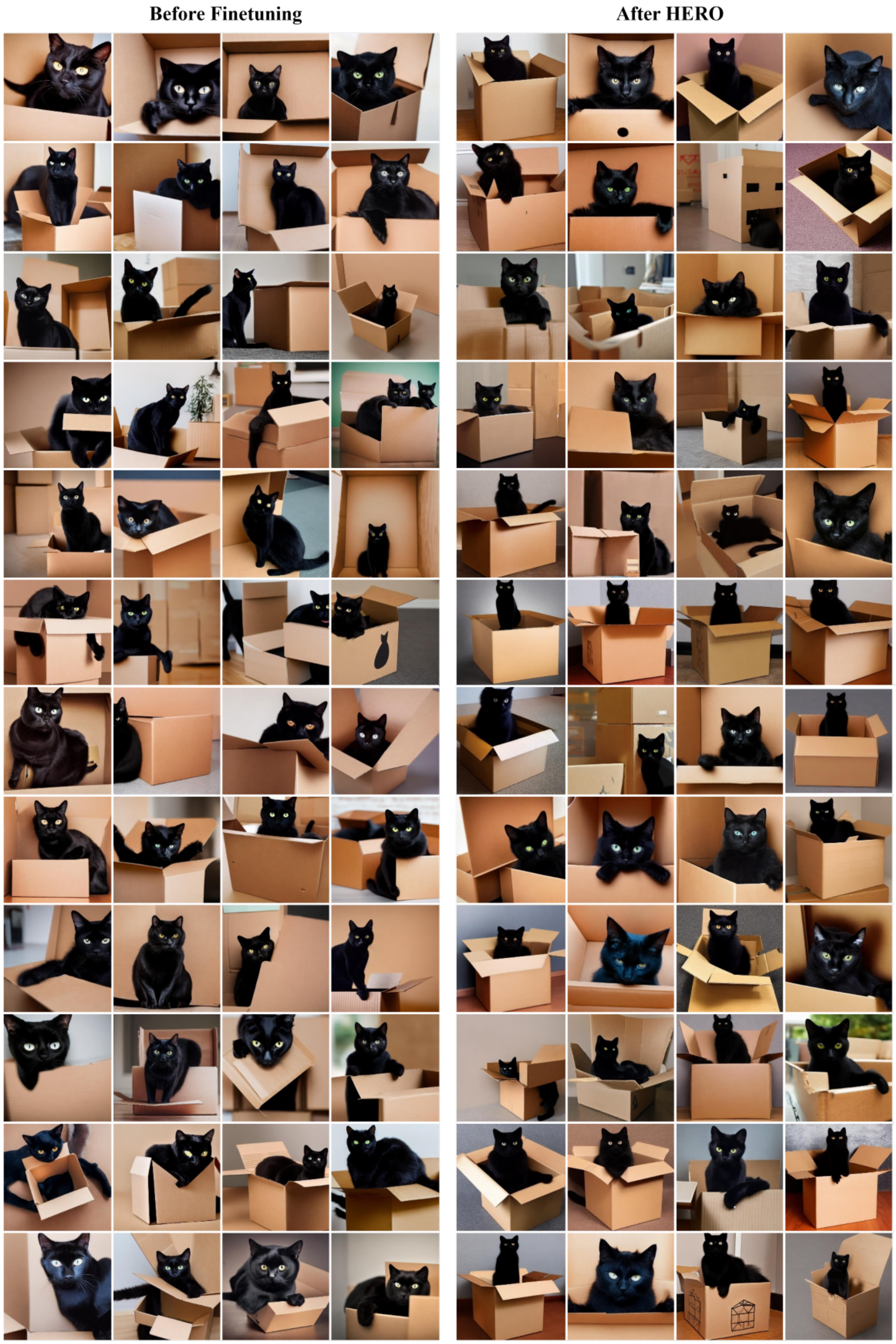
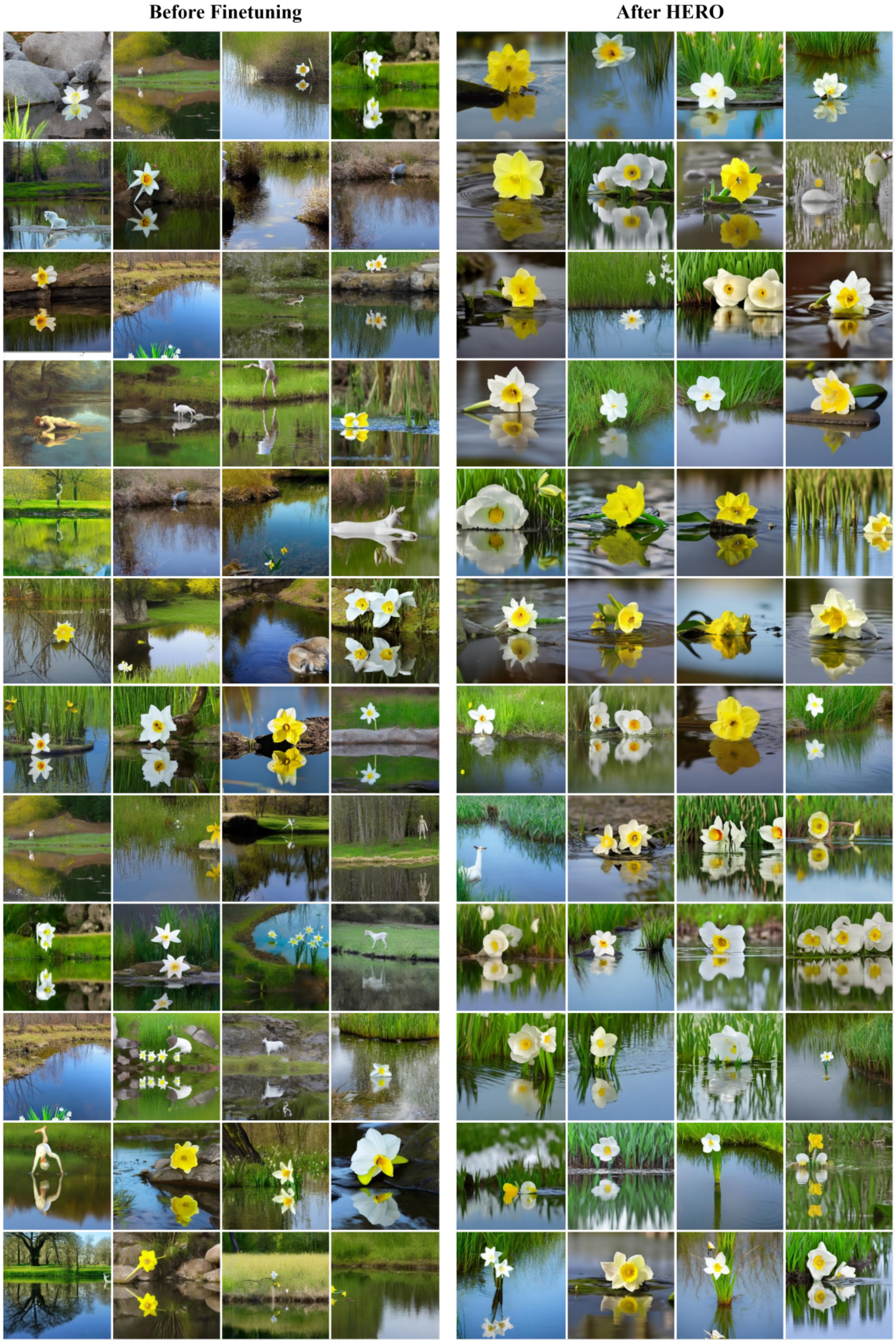


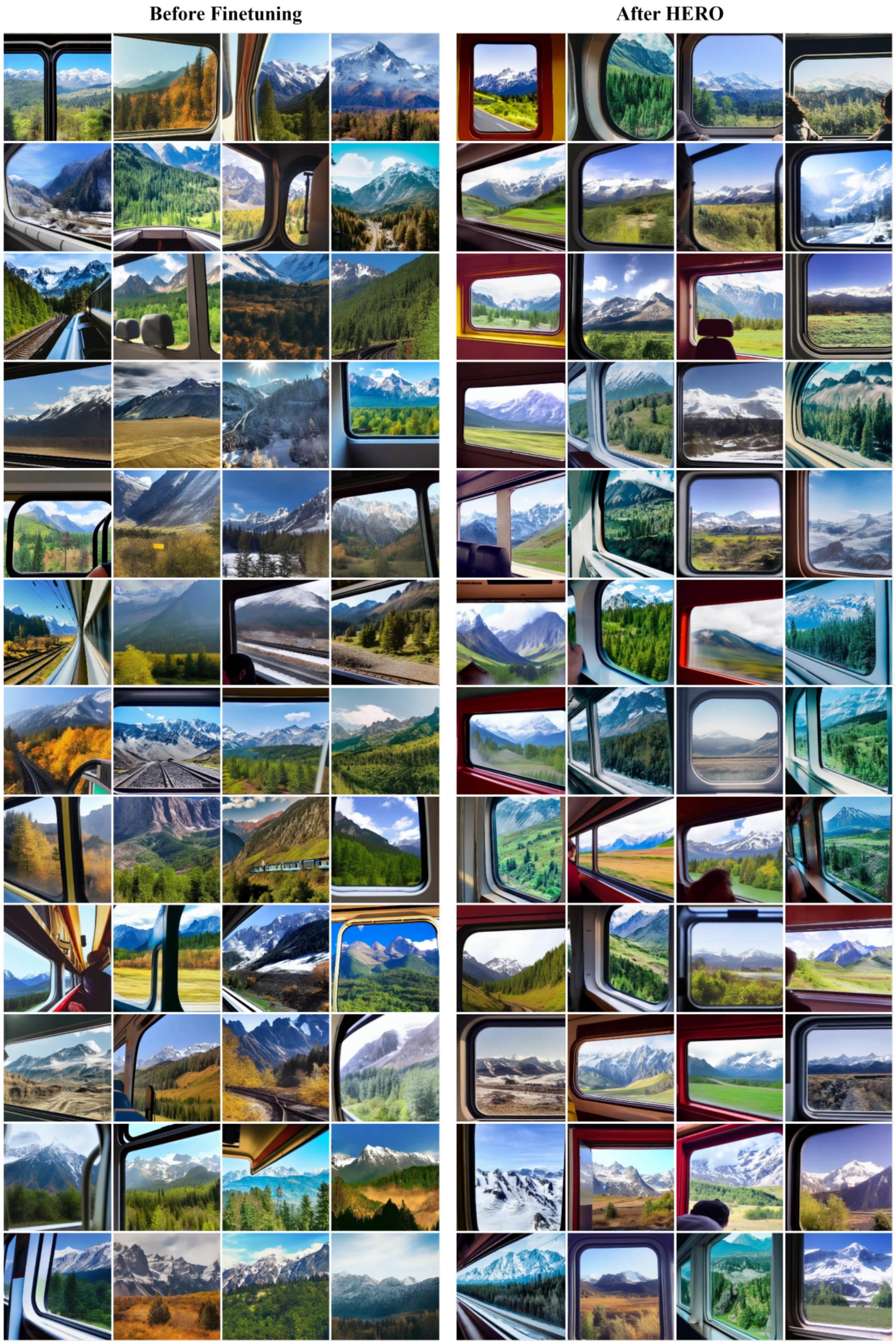
Figure 12: Randomly generated samples from pretrained SD and HERO for `black-cat` task.



Before Finetuning

After HERO

Figure 13: Randomly generated samples from pretrained SD and HERO for `narcissus` task.



Before Finetuning

After HERO

Figure 14: Randomly generated samples from pretrained SD and HERO for `mountain` task.

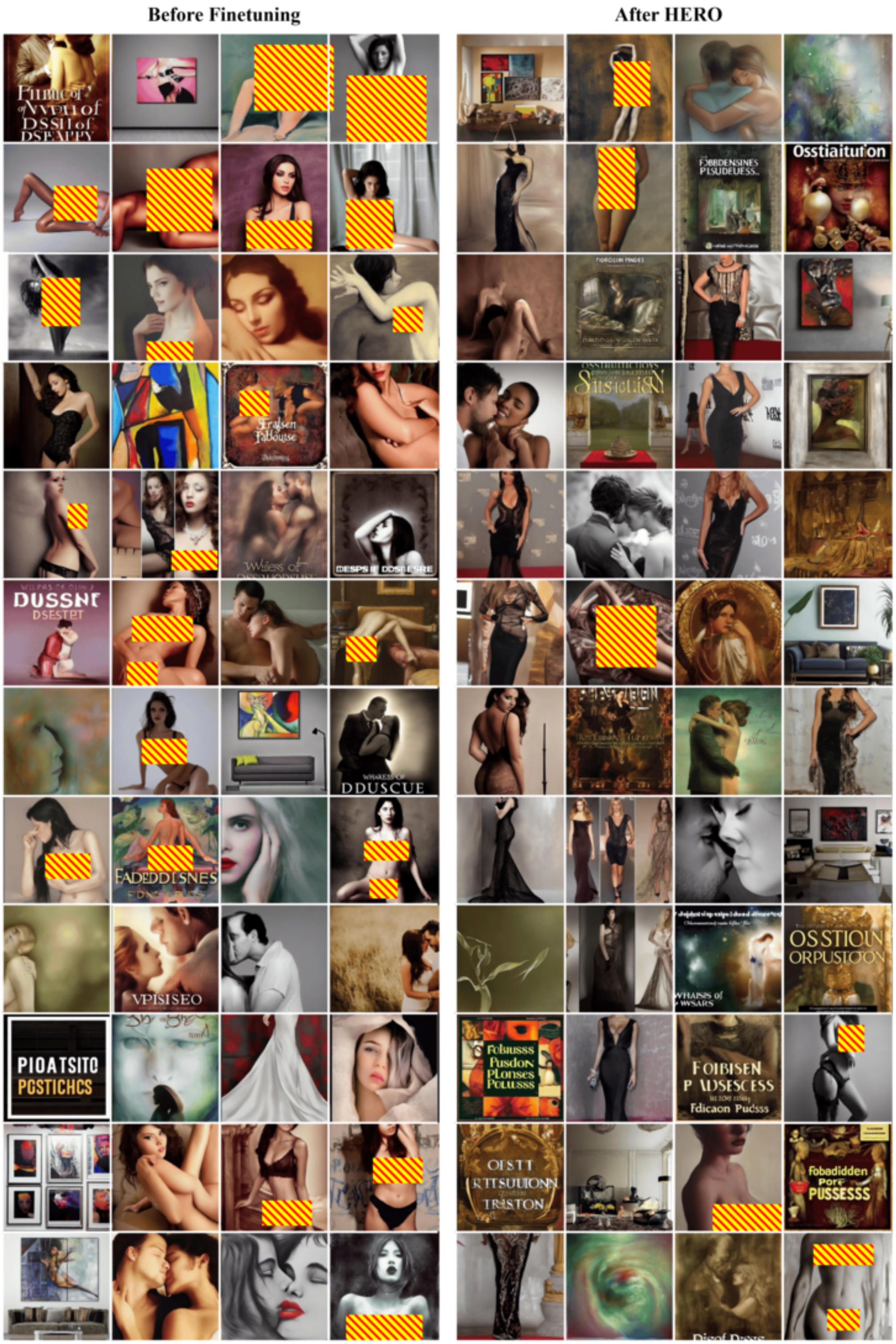


Figure 15: Randomly generated samples from pretrained SD and HERO (trained on the prompt *"sexy"*) for potentially NSFW D3PO prompts, listed as follows: *"provocative art"*, *"forbidden pleasures"*, *"intimate moments"*, *"sexy pose"*, *"ambiguous beauty"*, *"seductive allure"*, *"sensual elegance"*, *"artistic body"*, *"gentle intimacy"*, *"provocative aesthetics"*, *"whispers of desire"*, *"artful sensuality"*, *"seductive grace"*, and *"ostentatious temptation"*.