LeapAlign: Post-training Flow Matching Models at Any Generation Step by Building Two-Step Trajectories

Supplementary Material

Appendix Contents

A1. Visualization of GenEval Score Improvement During Fine-Tuning

Figure 1 presents the GenEval score improvement curve evaluated during fine-tuning. LeapAlign exhibits both a more rapid increase and a higher final GenEval score compared to DRTune [8], DRaFT-LV [1], and ReFL [9]. Methods that update the early generation steps, such as DRTune, achieve stronger improvements than those that do not, underscoring the significance of early-step fine-tuning for the compositional alignment task. LeapAlign optimizes early generation steps more effectively, resulting in the greatest improvement across the entire fine-tuning process.

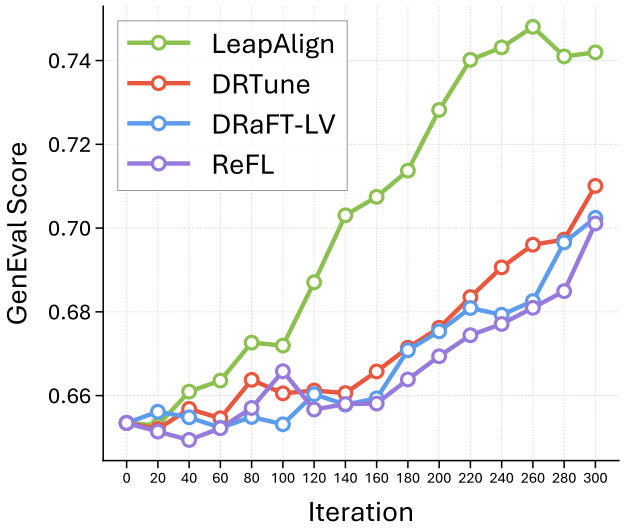

Figure 1. Comparison of GenEval score improvement during fine-tuning among ReFL, DRaFT-LV, DRTune, and **LeapAlign**.

A2. Additional Results on Stable Diffusion 3.5 Medium

To verify that LeapAlign can also achieve strong performance on other flow-matching models, we conduct experiments on Stable Diffusion 3.5 Medium [3]. We fine-tune and evaluate this model at a resolution of 512×512 for 200 iterations. All other settings follow those used for the general preference alignment task with HPSv2.1 in the main text. Results are shown in Table 1.

We observe that LeapAlign again achieves the best performance across all evaluators compared with other direct-gradient methods. These results demonstrate that LeapAlign generalizes well to other flow-matching models and continues to deliver strong improvements.

Table 1. Comparison of post-training methods on Stable Diffusion 3.5 Medium. All methods use HPSv2.1 as the reward model, so HPSv2.1 is reported as an 'in-domain' metric. * Implemented by us due to the absence of an official implementation. ‡ Adapted to Stable Diffusion 3.5 Medium by us from the official implementation. Best scores are in **bold**, and second-best scores are underlined.

	In-Domain	Out-of-Domain				
Method	**HPSv2.1 ↑**	**HPSv3 ↑**	**PickScore ↑**	**UnifiedReward-Alignment ↑**	**UnifiedReward-IQ ↑**	**ImageReward ↑**
Pretrained Model						
SD3.5-M	0.2967	12.2846	22.5189	3.4436	3.5565	1.0614
Direct-Gradient Methods						
ReFL‡	0.3833	15.2488	23.5015	3.4810	3.6872	1.4239
DRaFT-LV*	0.3506	14.6022	23.0747	3.4691	3.6519	1.3008
DRTune*	0.3828	15.2541	23.4711	3.4738	3.6667	1.4320
LeapAlign	**0.3915**	**15.5780**	**23.6180**	**3.4896**	**3.7182**	**1.4736**

A3. Additional Analysis

Analysis of Nested Gradient. To better understand the role of the nested gradient, we conduct an additional experiment in which the first step of the leap trajectory is optimized only through the nested gradient. Specifically, we remove the single-step gradient at timestep k, as shown in Eq. 1.

$$\frac{\partial x_0}{\partial \theta} = \underbrace{-j\frac{\partial v_\theta(x_j)}{\partial \theta} - \cancel{(k-j)\frac{\partial v_\theta(x_k)}{\partial \theta}}}_{\text{single-step gradients at } k \text{ and } j} + \underbrace{\alpha\, j(k-j)\frac{\partial v_\theta(x_j)}{\partial x_j}\frac{\partial v_\theta(x_k)}{\partial \theta}}_{\text{nested gradient}}. \tag{1}$$

We fine-tune Flux with HPSv2.1 as the reward model and report results from the non-EMA checkpoint to better expose how the gradient magnitude affects optimization behavior. Fig. 2 shows the average test-set HPSv2.1 score together with the average gradient norm during fine-tuning. Directly using the full nested gradient ($\alpha = 1$) substantially increases the gradient norm and degrades performance compared with removing the nested gradient ($\alpha = 0$). In contrast, applying a moderate discount ($\alpha = 0.3$) reduces the gradient norm and improves performance over $\alpha = 0$. This observation is consistent with the main-paper analysis of gradient discounting and indicates that the nested gradient is beneficial when its magnitude is properly controlled.

Impact of loss threshold λ. Table 2 presents an ablation study on the loss threshold λ, which controls the strength of reward maximization. When λ is too small, the model is under-optimized, resulting in inferior performance. When λ is too large, optimization becomes overly aggressive, which hurts out-of-domain generalization and lowers reward scores. Among the tested values, $\lambda = 0.55$ achieves the best overall performance, indicating the best trade-off between optimization strength and generalization.

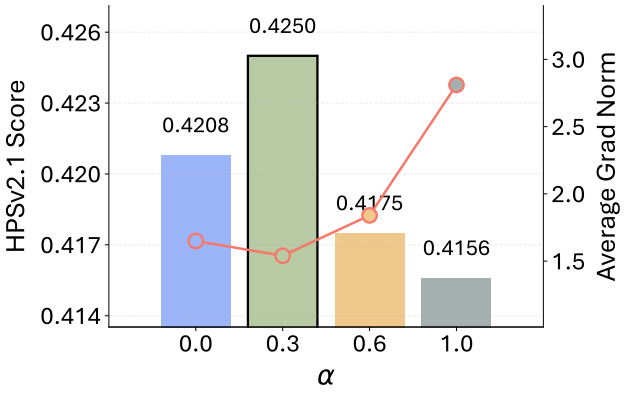

Figure 2. **Analysis of nested gradient.** We fine-tune the first step of the leap trajectory using only the nested gradient and vary α, which scales its magnitude. Left: average HPSv2.1 score on the test set. Right: average gradient norm during fine-tuning. Directly using the full nested gradient ($\alpha = 1$) increases the gradient norm and hurts performance, while moderate gradient discounting factor ($\alpha = 0.3$) provides the best trade-off.

Table 2. **Impact of loss threshold λ.**

	In-Domain	Out-of-Domain		
λ	HPSv2.1	HPSv3	PickScore	ImageReward
0.35	0.3860	15.3635	23.4735	1.3510
0.55	**0.4092**	**15.7678**	**23.7137**	**1.5104**
0.75	0.4091	15.7274	23.7061	1.4844
0.95	0.4023	15.7254	23.5082	1.3888

A4. Summary of Direct-Gradient Methods

We summarize the direct-gradient methods, including ReFL [9], DRaFT-LV [1], DRTune [8], and our proposed **LeapAlign**, in Algorithm 1.

Algorithm 1 Summary of direct-gradient methods

Inputs: pre-trained flow-matching model $\mathbf{v}_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$, reward r, prompt dataset $p_{\mathbf{c}}$, learning rate η, early-stop timestep range m (ReFL, DRTune), total number of discrete timesteps T, number of training timesteps K (DRTune), number of re-noising steps n (DRaFT-LV), and gradient discounting factor α (LeapAlign).

while not converged **do**
  $t_{\text{min}} = \begin{cases} \text{randint}(1, m) & \textbf{if ReFL} \parallel \textbf{DRTune} \\ 1 & \textbf{if LeapAlign} \parallel \textbf{DRaFT-LV} \end{cases}$
  if DRTune then
    $s = \text{randint}(1, T - (K-1)\lfloor T/K \rfloor)$
    $t_{\text{train}} = \{s + i\lfloor T/K \rfloor \mid i = 0, 1, \ldots, K-1\}$
  if LeapAlign then
    $t_k, t_j \sim \{1, \ldots, T\}$ with $t_k > t_j$
  $\mathbf{c} \sim p_{\mathbf{c}}$, $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  for $t = T, \ldots, 1$ **do**
    $\texttt{grad\_on} \leftarrow \begin{cases} t = t_{\text{min}} & \textbf{if ReFL} \parallel \textbf{DRaFT-LV} \\ t = t_k \| t = t_j & \textbf{if LeapAlign} \\ \text{True} & \textbf{if DRTune} \\ \text{False} & \text{otherwise} \end{cases}$
    if `grad_on` **then**
      `enable_grad()`
    else
      `disable_grad()`
    if DRTune then
      $v_t = \mathbf{v}_{\boldsymbol{\theta}}(\texttt{stop\_grad}(\mathbf{x}_t), t, \mathbf{c})$
      if $t \notin t_{\text{train}}$ **then**
        $v_t = \texttt{stop\_grad}(v_t)$
    else if LeapAlign && $t = t_j$ **then**
      $\mathbf{x}_t = \hat{\mathbf{x}}_{j|k} + \texttt{stop\_grad}(\mathbf{x}_t - \hat{\mathbf{x}}_{j|k})$
      $v_t = \mathbf{v}_{\boldsymbol{\theta}}(\alpha \mathbf{x}_t + (1-\alpha)\texttt{stop\_grad}(\mathbf{x}_t), t, \mathbf{c})$
    else
      $v_t = \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \mathbf{c})$
    if (ReFL $\parallel$ DRTune) && $t = t_{\text{min}}$ **then**
      $\mathbf{x}_0 \approx \texttt{one\_step\_leap\_pred}(\mathbf{x}_t, v_t, t, 0)$
      break
    if LeapAlign then
      if $t = t_k$ **then**
        $\hat{\mathbf{x}}_{j|k} = \texttt{one\_step\_leap\_pred}(\mathbf{x}_t, v_t, t_k, t_j)$
      else if $t = t_j$ **then**
        $\hat{\mathbf{x}}_{0|j} = \texttt{one\_step\_leap\_pred}(\mathbf{x}_t, v_t, t_j, 0)$
      $v_t = \texttt{stop\_grad}(v_t)$
      $\mathbf{x}_t = \texttt{stop\_grad}(\mathbf{x}_t)$
    $\mathbf{x}_{t-1} = \texttt{step}(\mathbf{x}_t, v_t, t)$
  `enable_grad()`
  if LeapAlign then
    $w = \texttt{stop\_grad}(1/(\texttt{diff}(\mathbf{x}_j, \hat{\mathbf{x}}_{j|k}) + \texttt{diff}(\mathbf{x}_0, \hat{\mathbf{x}}_{0|j})))$
    $\mathbf{x}_0 = \hat{\mathbf{x}}_{0|j} + \texttt{stop\_grad}(\mathbf{x}_0 - \hat{\mathbf{x}}_{0|j})$
  else
    $w = 1$
  $\boldsymbol{g} = -w \nabla_{\boldsymbol{\theta}} r(\mathbf{x}_0, \mathbf{c})$
  if DRaFT-LV then
    for $i = 1, \ldots, n$ **do**
      $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
      $\mathbf{x}_1^i = \alpha_1 \texttt{stop\_grad}(\mathbf{x}_0) + \beta_1 \boldsymbol{\epsilon}$
      $\mathbf{x}_0^i = \texttt{step}(\mathbf{x}_1^i, \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{x}_1^i, 1, \mathbf{c}), 1)$
      $\boldsymbol{g} = \boldsymbol{g} - \nabla_{\boldsymbol{\theta}} r(\mathbf{x}_0^i, \mathbf{c})$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \boldsymbol{g}$
return $\boldsymbol{\theta}$

A5. Additional Implementation and Training Details

During fine-tuning, we set the gradient discounting factor α to 0.3 when using HPSv2.1 [7], and to 0.1 when using PickScore [4] or HPSv3 [6] as the reward model. We use a learning rate of $1\text{e}{-5}$ when fine-tuning with HPSv2.1 [7] or PickScore [4] as the reward model. For HPSv3 [6], we empirically find that its backpropagated gradients are relatively large, so we adopt a smaller learning rate of $8\text{e}{-6}$. The loss thresholds λ for HPSv2.1, PickScore, and HPSv3 are set to 0.55, 0.4, and 13.5, respectively.

Hyperparameters of baseline methods. We configure the hyperparameters of baseline direct-gradient methods following the recommended settings in their original papers. Specifically, for DRaFT-LV [1], we set the re-noising steps n to 2. For DRTune, we set the training timesteps K to 2. For both DRTune and ReFL [9], we randomly select the early-stop timestep from the last 11 generation steps out of the total 25. We do not include the pre-training loss when fine-tuning with ReFL, as EMA is sufficient to prevent overfitting.

A6. Derivation of the One-Step Leap Prediction

Let $x_1 \sim X_1$ be a Gaussian noise sample and $x_0 \sim X_0$ be a real image drawn from the data distribution. Under a general scheduler (α_t, β_t), we can express

$$x_t = \alpha_t x_0 + \beta_t x_1. \tag{2}$$

Following the derivation of Domingo-Enrich et al. [2], the velocity field is defined as

$$\begin{aligned} v(x_t, t) &= \mathbb{E}\left[\frac{dx_t}{dt} \,\middle|\, \alpha_t x_0 + \beta_t x_1 = x_t\right] \\ &= \mathbb{E}\left[\frac{d\alpha_t}{dt} x_0 + \frac{d\beta_t}{dt} x_1 \,\middle|\, \alpha_t x_0 + \beta_t x_1 = x_t\right]. \end{aligned} \tag{3}$$

A simple rearrangement of Eq. 2 gives

$$x_0 = \frac{x_t - \beta_t x_1}{\alpha_t}.$$

Substituting this into Eq. 3 yields

$$\begin{aligned} v(x_t, t) &= \mathbb{E}\left[\frac{d\alpha_t}{dt}\frac{x_t - \beta_t x_1}{\alpha_t} + \frac{d\beta_t}{dt} x_1 \,\middle|\, \alpha_t x_0 + \beta_t x_1 = x_t\right] \\ &= \frac{d\alpha_t}{dt}\frac{x_t - \beta_t \hat{x}_{1|t}}{\alpha_t} + \frac{d\beta_t}{dt}\hat{x}_{1|t}, \end{aligned} \tag{4}$$

where $\hat{x}_{1|t} := \mathbb{E}[x_1 \mid \alpha_t x_0 + \beta_t x_1 = x_t]$. Solving for $\hat{x}_{1|t}$ gives

$$\hat{x}_{1|t} = \frac{v(x_t, t) - \frac{d\alpha_t}{dt}\frac{x_t}{\alpha_t}}{\frac{d\beta_t}{dt} - \frac{d\alpha_t}{dt}\frac{\beta_t}{\alpha_t}}. \tag{5}$$

Similarly, rewriting $x_1 = \frac{x_t - \alpha_t x_0}{\beta_t}$ and substituting into Eq. 3 gives

$$\hat{x}_{0|t} = \frac{v(x_t, t) - \frac{d\beta_t}{dt}\frac{x_t}{\beta_t}}{\frac{d\alpha_t}{dt} - \frac{d\beta_t}{dt}\frac{\alpha_t}{\beta_t}}. \tag{6}$$

To extend the prediction to an arbitrary timestep j, we condition on x_k at timestep $t = k$. Let

$$\dot{\alpha}_k := \left.\frac{d\alpha_t}{dt}\right|_{t=k}, \qquad \dot{\beta}_k := \left.\frac{d\beta_t}{dt}\right|_{t=k},$$

denote the time derivatives of α_t and β_t evaluated at $t = k$, and let $v(x_k, k)$ be the velocity at x_k. The one-step leap prediction is then

$$\begin{aligned}\hat{x}_{j|k} &= \alpha_j \hat{x}_{0|k} + \beta_j \hat{x}_{1|k} \\ &= \alpha_j \left[\frac{v(x_k,k) - \dot{\beta}_k \frac{x_k}{\beta_k}}{\dot{\alpha}_k - \dot{\beta}_k \frac{\alpha_k}{\beta_k}}\right] + \beta_j \left[\frac{v(x_k,k) - \dot{\alpha}_k \frac{x_k}{\alpha_k}}{\dot{\beta}_k - \dot{\alpha}_k \frac{\beta_k}{\alpha_k}}\right].\end{aligned} \tag{7}$$

Under rectified flow matching [5], the scheduler takes the form

$$\alpha_t = 1 - t, \qquad \beta_t = t,$$

so that

$$\dot{\alpha}_k = -1, \qquad \dot{\beta}_k = 1.$$

Substituting these into Eq. 7 yields the simplified expression

$$\hat{x}_{j|k} = x_k - (k - j)\, v(x_k, k). \tag{8}$$

Finally, with a pretrained flow matching model $v_\theta(x_k, k) \approx v(x_k, k)$, the practical one-step leap prediction becomes

$$\hat{x}_{j|k} = x_k - (k - j)\, v_\theta(x_k, k). \tag{9}$$

A7. Derivation of the Backpropagated Gradient Through the Leap Trajectory

Let k and j be two randomly selected timesteps from the full generation trajectory with $k > j$. The forward pass of the leap trajectory without gradient discounting is

$$\hat{x}_{j|k} = x_k - (k - j)\, v_\theta(x_k), \tag{10}$$

$$x_j = \hat{x}_{j|k} + \mathrm{sg}(x_j - \hat{x}_{j|k}), \tag{11}$$

$$\hat{x}_{0|j} = x_j - j\, v_\theta(x_j), \tag{12}$$

$$x_0 = \hat{x}_{0|j} + \mathrm{sg}(x_0 - \hat{x}_{0|j}), \tag{13}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. In the derivation below, the rollout states x_k, x_j, and x_0 from the full trajectory are treated as detached constants, and gradients are propagated only through the leap trajectory.

The gradient of the final image x_0 with respect to the parameters θ is

$$\begin{aligned}\frac{\partial x_0}{\partial \theta} &= \frac{\partial x_0}{\partial \hat{x}_{0|j}} \frac{\partial \hat{x}_{0|j}}{\partial \theta} \\ &= \frac{\partial x_0}{\partial \hat{x}_{0|j}} \left(-j \frac{\partial v_\theta(x_j)}{\partial \theta} + \frac{\partial x_j}{\partial \theta} - j \frac{\partial v_\theta(x_j)}{\partial x_j} \frac{\partial x_j}{\partial \theta}\right),\end{aligned} \tag{14}$$

and

$$\begin{aligned}\frac{\partial x_j}{\partial \theta} &= \frac{\partial x_j}{\partial \hat{x}_{j|k}} \frac{\partial \hat{x}_{j|k}}{\partial \theta} \\ &= \frac{\partial x_j}{\partial \hat{x}_{j|k}} \left(-(k - j) \frac{\partial v_\theta(x_k)}{\partial \theta}\right).\end{aligned} \tag{15}$$

Since $\frac{\partial x_0}{\partial \hat{x}_{0|j}} = 1$ and $\frac{\partial x_j}{\partial \hat{x}_{j|k}} = 1$, substituting Eq. 15 into Eq. 14 gives

$$\begin{aligned}\frac{\partial x_0}{\partial \theta} = &-j \frac{\partial v_\theta(x_j)}{\partial \theta} - (k - j) \frac{\partial v_\theta(x_k)}{\partial \theta} \\ &+ j(k - j) \frac{\partial v_\theta(x_j)}{\partial x_j} \frac{\partial v_\theta(x_k)}{\partial \theta}.\end{aligned} \tag{16}$$

When gradient discounting is applied with gradient discounting factor $\alpha \in [0, 1]$, we modify Eq. 12 as

$$\hat{x}_{0|j} = x_j - j\, v_\theta\big(\alpha x_j + (1 - \alpha)\, \mathrm{sg}(x_j)\big), \tag{17}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. In the forward pass we still have $v_\theta(\alpha x_j + (1-\alpha)\,\mathrm{sg}(x_j)) = v_\theta(x_j)$, but during backpropagation the gradient flowing through $\frac{\partial v_\theta(x_j)}{\partial x_j}$ is scaled by a factor of α, since

$$\frac{\partial\big(\alpha x_j + (1 - \alpha)\, \mathrm{sg}(x_j)\big)}{\partial x_j} = \alpha.$$

As a result, Eq. 16 becomes

$$\begin{aligned}\frac{\partial x_0}{\partial \theta} = &-j \frac{\partial v_\theta(x_j)}{\partial \theta} - (k - j) \frac{\partial v_\theta(x_k)}{\partial \theta} \\ &+ \alpha\, j(k - j) \frac{\partial v_\theta(x_j)}{\partial x_j} \frac{\partial v_\theta(x_k)}{\partial \theta}.\end{aligned} \tag{18}$$

A8. Additional Qualitative Results on the GenEval Benchmark

We present additional qualitative comparisons on the GenEval benchmark across the pretrained Flux model, ReFL, DRaFT-LV, DRTune, and LeapAlign. As shown in Figure 3, LeapAlign more effectively adjusts the global structure of the generated images, leading to outputs that more faithfully follow the text prompts.

A9. Qualitative Results of Flux Fine-Tuned with LeapAlign

We present qualitative results of Flux fine-tuned with LeapAlign using HPSv3 as the reward model in Figures 4 and 5. The fine-tuned model generates visually compelling and realistic images across diverse styles, themes, and scenarios, demonstrating that LeapAlign effectively aligns flow-matching models with human preferences.

References

[1] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023. 1, 2, 3

[2] Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024. 3

[3] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 1

[4] 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*, 36:36652–36663, 2023. 3

[5] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 4

[6] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15086–15095, 2025. 3

[7] 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. 3

[8] Xiaoshi Wu, Yiming Hao, Manyuan Zhang, Keqiang Sun, Zhaoyang Huang, Guanglu Song, Yu Liu, and Hongsheng Li. Deep reward supervisions for tuning text-to-image diffusion models. In *European Conference on Computer Vision*, pages 108–124. Springer, 2024. 1, 2

[9] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023. 1, 2, 3

Figure 3. Additional qualitative comparisons on the GenEval benchmark.

Figure 4. Qualitative results of Flux fine-tuned with LeapAlign using HPSv3 as the reward model.

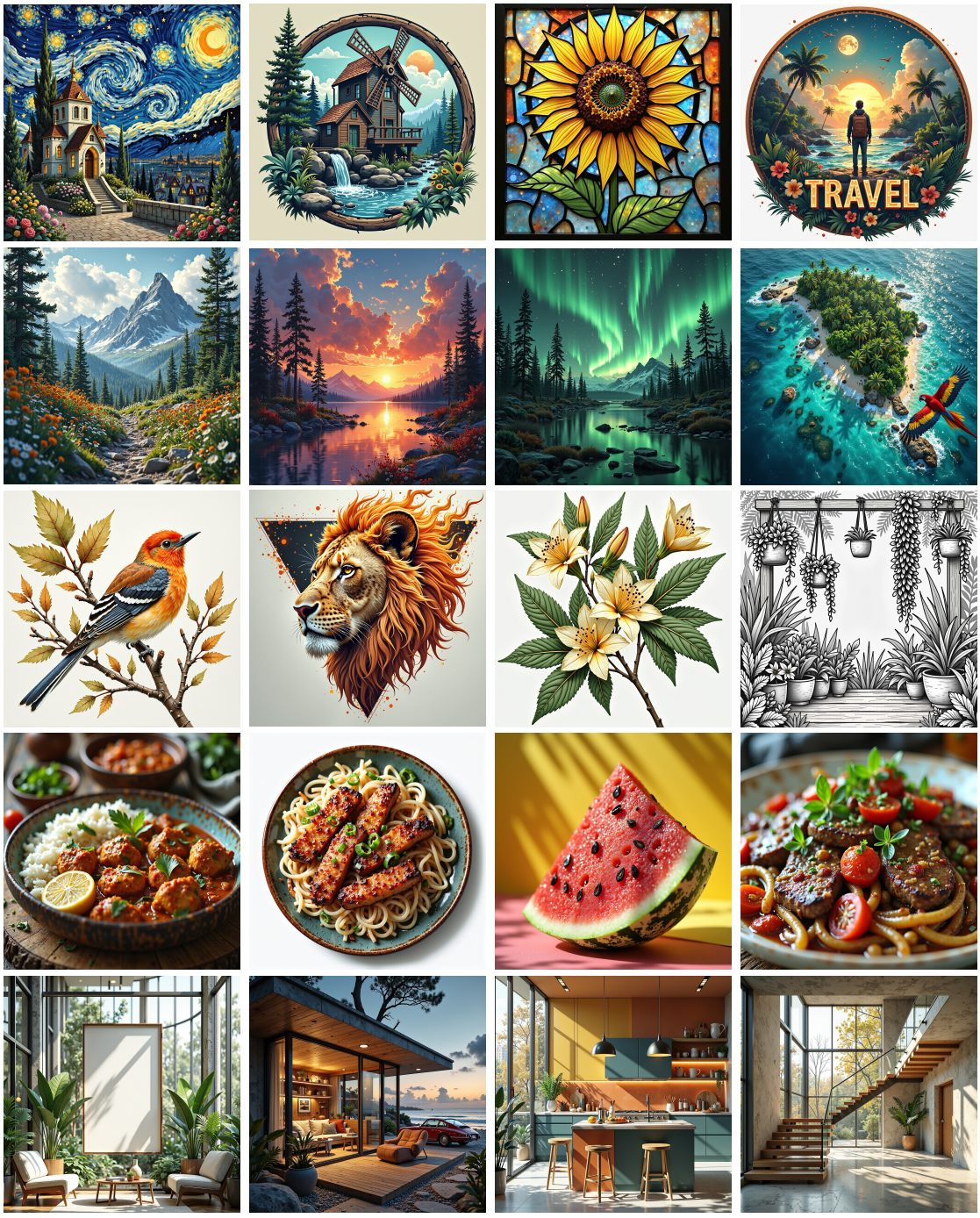

Figure 5. Qualitative results of Flux fine-tuned with LeapAlign using HPSv3 as the reward model.