# Test-time Scaling in Diffusion Models

Junoh Kang

Computer Vision Laboratory
ECE, Seoul National University
junoh.kang@snu.ac.kr

# Motivation

## Conditional sampling

Stages in generative tasks:

1. Imitate data distribution
2. Conditional generation
   - Simple conditions : class
   - Complicated conditions : text, image, style, human preference

# Motivation

## Conditional sampling

For conditional generation, two guidance approaches are dominant:

- Guidance

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \lambda \underbrace{\nabla_{\mathbf{x}_t} \log p(c|\mathbf{x}_t)}_{\text{guidance}} \quad (1)$$

- Fine-tune (Classifier-free guidance)

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\phi) + \lambda \underbrace{\left(\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|c) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\phi)\right)}_{\text{classifier-free guidance}} \quad (2)$$

# Motivation

## Conditional sampling

Requirements to perform **guidance**:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \lambda \underbrace{\nabla_{\mathbf{x}_t} \log p(c|\mathbf{x}_t)}_{\text{guidance}}$$

- **Well-defined** and **differentiable** $\log p(c|\mathbf{x}_t)$.
  - If $c$ is class, we need to train classifier.
  - If $c$ is reference image, this can be $\mathcal{L}_2$ distance between $c$ and $\mathbf{x}_t$.
- We need to train model that estimates $p(c|\mathbf{x}_t)$ for most cases.

# Motivation

## Conditional sampling

Requirements to perform **Fine-tune (classifier-free guidance)**

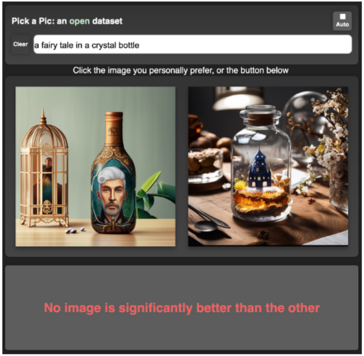
$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\phi) + \lambda \underbrace{\left(\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|c) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\phi)\right)}_{\text{classifier-free guidance}}$$

- **Training** for classifier-free guidance.
- Fine-tune may sacrifice image quality or diversity.
- Cannot fine-tune for every condition.

# Motivation

## Conditional sampling

What if the conditions are hard to quantify?




User preference

Number of objects

# Motivation

- Optimize with respect to diverse conditions $c$ without fine-tuning diffusion models nor training external models.
- Test-time scalining techniques like Chain of Thoughts (CoT) are very successful in LLMs.

Table 1: Conditional generation approaches and requirements.

| Approaches | No external model training | No diffusion model training |
|---|---|---|
| Guidacne | | ✓ |
| CFG | ✓ | |
| Test-time scaling | ✓ | ✓ |

# Test-time scaling

## Inference-time reward alignment

Our goal is to sample high-reward samples $\mathbf{x}_0 \in \mathbb{R}^d$ from data distribution $p_0$. Then, for a reward function $r : \mathbb{R}^d \to \mathbb{R}$, the target distribution is

$$p_0^* = \arg\max_q \mathbb{E}_{\mathbf{x}_0 \sim q}[r(\mathbf{x}_0)] - \beta D_{KL}(q || p_0), \tag{3}$$

and it is proved in [Rafailov et al., 2024] that

$$p_0^*(\mathbf{x}_0) = \frac{1}{Z} p_0(\mathbf{x}_0) \exp(r(\mathbf{x}_0)/\beta), \tag{4}$$

for normalization constant $Z$.

# Test-time scaling
## Particle sampling

For diffusion model, we have to sample from **conditional distribution with the optimal policy**:

$$p_\theta^*(\mathbf{x}_t|\mathbf{x}_{t+1}) = \frac{1}{Z} p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}) \exp(v(\mathbf{x}_t)/\beta), \tag{5}$$

where $v(\mathbf{x}_t) \approx \mathbb{E}_{\mathbf{x}_0|\mathbf{x}_t}[r(\mathbf{x}_0)]$.

One sampling option is **particle sampling**.

1. Make a set of candidate particles.
2. Propagate high-reward particles.

- SVDD [Li et al., 2024] uses importance sampling.
- DAS [Kim et al., 2025b] uses Sequential Monte Carlo (SMC).

# Test-time scaling in diffusion models

## SVDD [Li et al., 2024]

**Derivative-Free Guidance in Continuous and Discrete Diffusion Models with Soft Value-Based Decoding**

**Xiner Li** [1*] **Yulai Zhao** [2] **Chenyu Wang** [3] **Gabriele Scalia** [4]
**Gokcen Eraslan** [4] **Surag Nair** [4] **Tommaso Biancalani** [4] **Shuiwang Ji** [1]
**Aviv Regev** [4†] **Sergey Levine** [5†] **Masatoshi Uehara** [4†]
[1]Texas A&M University [2]Princeton University [3]MIT [4]Genentech [5]UC Berkeley

# Test-time scaling in diffusion models

## SVDD [Li et al., 2024]

SVDD [Li et al., 2024] uses **importance sampling** to sample from

$$p^*_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}) = \frac{1}{Z} p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}) \exp(v(\mathbf{x}_t)/\beta).$$

1. Sample particles from proposal distribution:

$$\{\mathbf{x}_t^{(i)}\}_1^K \sim p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}).$$

2. Calculate weight:

$$w_t^{(i)} = \exp(v(\mathbf{x}_t^{(i)})/\beta).$$

3. Select particle from categorical distribution:

$$\mathbf{x}_t \sim \text{Categorical}(\mathbf{x}_1, \dots, \mathbf{x}_K; w_t^{(1)}, \dots, w_t^K)$$

# Test-time scaling in diffusion models

## SVDD [Li et al., 2024]

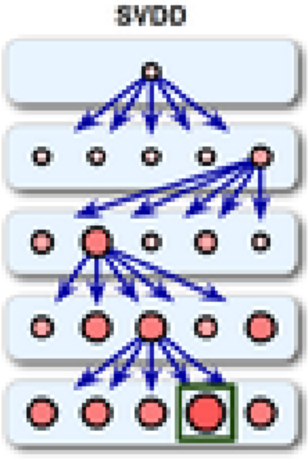



Figure 1: Illustration of SVDD

# Test-time scaling in diffusion models

## SVDD [Li et al., 2024]

**Algorithm 1 SVDD** (**S**oft **V**alue-Based **D**ecoding in **D**iffusion Models)

**Require**: Estimated soft value function $\{\hat{v}_t\}_{t=T}^{0}$ (refer to Algorithm 2 or Algorithm 3), pre-trained diffusion models $\{p_t^{\mathrm{pre}}\}_{t=T}^{0}$, hyperparameter $\alpha \in \mathbb{R}$

**for** $t \in [T+1, \cdots, 1]$ **do**

  Get $M$ samples from pre-trained polices $\{x_{t-1}^{\langle m \rangle}\}_{m=1}^{M} \sim p_{t-1}^{\mathrm{pre}}(\cdot|x_t)$, and for each $m$, calculate $w_{t-1}^{\langle m \rangle} := \exp(\hat{v}_{t-1}(x_{t-1}^{\langle m \rangle})/\alpha)$

  $x_{t-1} \leftarrow x_{t-1}^{\langle \zeta_{t-1} \rangle}$ after selecting an index: $\zeta_{t-1} \sim \mathrm{Categorical}\left(\left\{\frac{w_{t-1}^{\langle m \rangle}}{\sum_{j=1}^{M} w_{t-1}^{\langle j \rangle}}\right\}_{m=1}^{M}\right)$,

**end for**

**Output**: $x_0$

Figure 2: Algorithm of SVDD.

# Test-time scaling in diffusion models

SVDD [Li et al., 2024]

It still remains how to calculate reward at intermediate state, $v(\mathbf{x}_t) = \mathbb{E}_{\mathbf{x}_0|\mathbf{x}_t}[r(\mathbf{x}_0)]$. SVDD suggests two approaches:

- SVDD-MC: Monte Carlo regression.

**Algorithm 2** Value Function Estimation Using Monte Carlo Regression

1: **Require**: Pre-trained diffusion models, reward $r : \mathcal{X} \to \mathbb{R}$, function class $\Phi : \mathcal{X} \times [0, T] \to \mathbb{R}$.
2: Collect datasets $\{x_T^{(s)}, \cdots, x_0^{(s)}\}_{s=1}^S$ by rolling-out $\{p_t^{\mathrm{pre}}\}_{t=T}^0$ from $t = T$ to $t = 0$.
3: $\hat{v}' = \mathrm{argmin}_{f \in \Phi} \sum_{t=0}^T \sum_{s=1}^S \{r(x_0^{(s)}) - f(x_t^{(s)}, t)\}^2$.
4: **Output**: $\hat{v}'$

- SVDD-PM: Posterior mean approximation.

**Algorithm 3** Value Function Estimation using Posterior Mean Approximation

1: **Require**: Pre-trained diffusion models, reward $r : \mathcal{X} \to \mathbb{R}$
2: Set $\hat{v}^{\diamond}(\cdot, t) := r(\hat{x}_0(x_t = \cdot), t)$
3: **Output**: $\hat{v}^{\diamond}$

# Test-time scaling in diffusion models

## SVDD [Li et al., 2024]

Implementation details

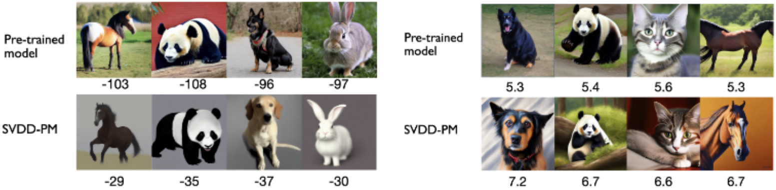
- $\beta = 0$ : select a particle with the largest reward.
- $5 \leq K \leq 20$

# Test-time scaling in diffusion models

## SVDD [Li et al., 2024]

Table 2: Top 10 and 50 quantiles of the generated samples for each algorithm (with 95% confidence intervals). Higher is better. **SVDD** consistently outperforms the baseline methods.

| Domain | Quantile | Pre-Train | Best-N | DPS | SMC | **SVDD-MC** | **SVDD-PM** |
|---|---|---|---|---|---|---|---|
| Image: Compress | 50% | -101.4 $\pm$ 0.22 | -71.2 $\pm$ 0.46 | -60.1 $\pm$ 0.44 | -59.7 $\pm$ 0.4 | -54.3 $\pm$ 0.33 | **-51.1** $\pm$ 0.38 |
| | 10% | -78.6 $\pm$ 0.13 | -57.3 $\pm$ 0.28 | -61.2 $\pm$ 0.28 | -49.9 $\pm$ 0.24 | -40.4 $\pm$ 0.2 | **-38.8** $\pm$ 0.23 |
| Image: Aesthetic | 50% | 5.62 $\pm$ 0.003 | 6.11 $\pm$ 0.007 | 5.61 $\pm$ 0.009 | 6.02 $\pm$ 0.004 | 5.70 $\pm$ 0.008 | **6.14** $\pm$ 0.007 |
| | 10% | 5.98 $\pm$ 0.002 | 6.34 $\pm$ 0.004 | 6.00 $\pm$ 0.005 | 6.28 $\pm$ 0.003 | 6.05 $\pm$ 0.005 | **6.47** $\pm$ 0.004 |




(a) Images: compressibility (b) Images: aesthetic scores

# Test-time scaling in diffusion models

SVDD [Li et al., 2024]

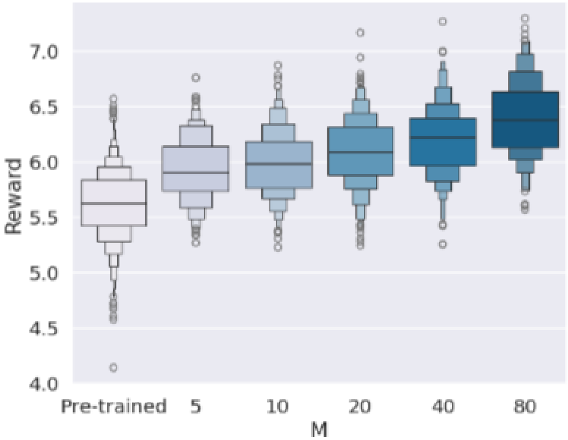



Figure 3: Number of particles and performance.

# Test-time scaling in diffusion models

## DAS [Kim et al., 2025b]

Published as a conference paper at ICLR 2025

TEST-TIME ALIGNMENT OF DIFFUSION MODELS WITHOUT REWARD OVER-OPTIMIZATION

**Sunwoo Kim**[1*] **Minkyu Kim**[2] **Dongmin Park**[2†]
[1] Seoul National University [2] KRAFTON
ksunw0209@snu.ac.kr {minkyu.kim, dongmin.park}@krafton.com

# Test-time scaling in diffusion models

DAS [Kim et al., 2025b]

DAS uses **SMC** (Sequential Monte Carlo) to sample from

$$p^*_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}) = \frac{1}{Z} p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}) \exp(v(\mathbf{x}_t)/\beta).$$

SMC extends the idea of importance sampling:

1. Sample each particle from each proposal distribution

$$\mathbf{x}_t^{(i)} \sim p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}^{(i)}).$$

2. Update weight

$$w_t^{(i)} = \frac{\exp(v(\mathbf{x}_t^{(i)})/\beta)}{\exp(v(\mathbf{x}_{t+1}^{(i)})/\beta)} w_{t+1}^{(i)}$$

3. If effective sample size, $\sum_{i=1}^K 1/(\tilde{w}_t^{(i)})^2$, is small, resample particles and initialize weights with 1.

# Test-time scaling in diffusion models

## DAS [Kim et al., 2025b]

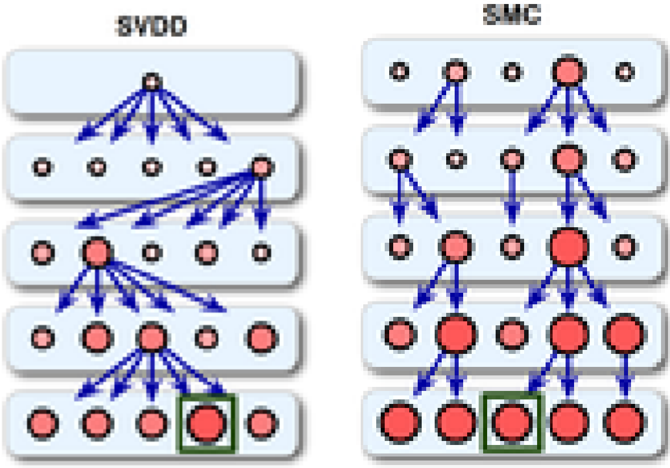



Figure 4: Illustration of SVDD and DAS

# Test-time scaling in diffusion models

## DAS [Kim et al., 2025b]

DAS does not use SMC naïvely. [Li et al., 2024] points out that SMC is suboptimal for small batch sizes, which is generally true for diffusion models.

# Test-time scaling in diffusion models

## DAS [Kim et al., 2025b]

DAS introduces **tempering** into SMC for diffusion model. They change target intermediate distribution:

$$\pi_t(\mathbf{x}_t) \propto p(\mathbf{x}_t) \exp(\frac{1}{\beta} v(\mathbf{x}_t)), \tag{6}$$

$$\Rightarrow \pi_t(\mathbf{x}_t) \propto p(\mathbf{x}_t) \exp(\frac{\lambda_t}{\beta} v(\mathbf{x}_t)), \tag{7}$$

where $\lambda_t = (1+\gamma)^t - 1$. Then, $\{\pi_t\}_T^0$ can interpolate more smoothly.

# Test-time scaling in diffusion models

## DAS [Kim et al., 2025b]

DAS also propose locally optimal proposal, which minimizes weight variance.

1. Sample each particle from each proposal distribution

$$
\begin{aligned}
\mathbf{x}_t &\sim p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}) = \mathcal{N}(\mu_\theta(\mathbf{x}_{t+1}), \sigma_{t+1}^2 \mathrm{I}) \\
\Rightarrow \mathbf{x}_t &\sim \mathcal{N}(\mu_\theta(\mathbf{x}_{t+1}) + \underbrace{\sigma_{t+1}^2 \frac{\lambda_t}{\beta} \nabla v(\mathbf{x}_{t+1})}_{\text{added}}, \sigma_{t+1}^2 \mathrm{I}) \qquad (8)
\end{aligned}
$$

- Reward function should be differentiable in this paper.

# Test-time scaling in diffusion models

## DAS [Kim et al., 2025b]

Can DAS optimize a single reward while avoiding over-optimization*?

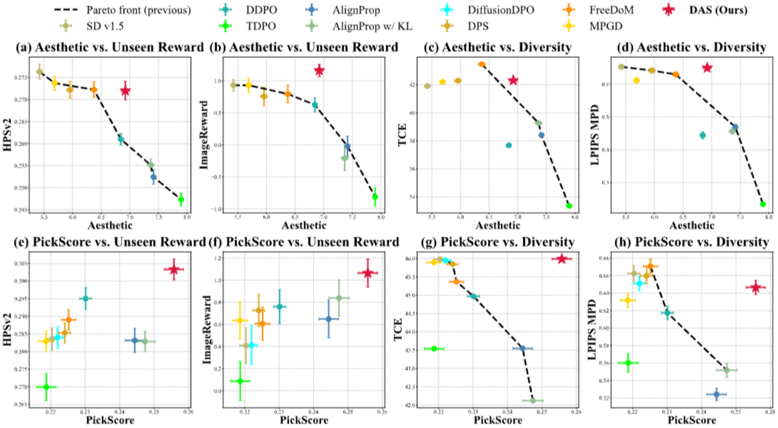



Figure 5: Target reward vs. evaluation metrics

---

* sacrifice quality and diversity

# Test-time scaling in diffusion models

## DAS [Kim et al., 2025b]

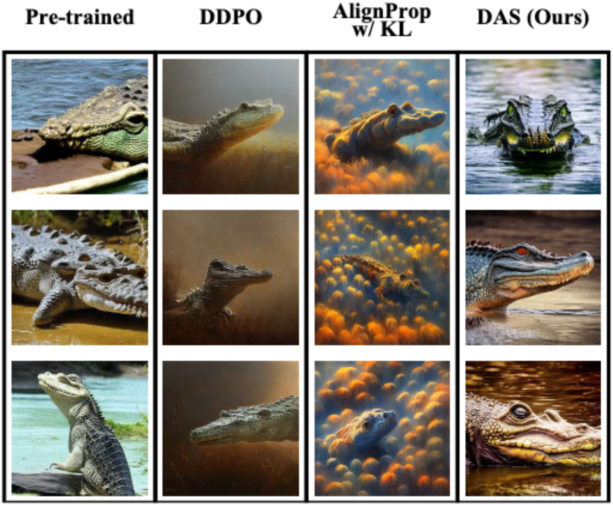



Figure 6: Qualitative comparison of over-optimization and diversity with fine-tuning methods.

# Test-time scaling in diffusion models

DAS [Kim et al., 2025b]

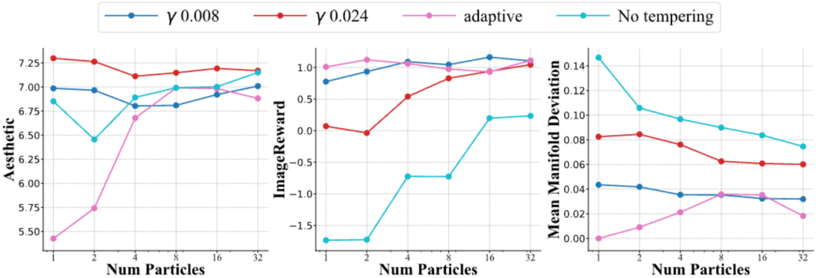



Figure 7: Ablation on tempering. We can infer that without tempering, samples suffer from over-optimization–high Aesthetic, low ImageReward.

# Test-time scaling in flow matching

[Kim et al., 2025a]

## Inference-Time Scaling for Flow Models via Stochastic Generation and Rollover Budget Forcing

**Jaihoon Kim*** **Taehoon Yoon*** **Jisung Hwang*** **Minhyuk Sung**
KAIST
`{jh27kim,taehoon,4011hjs,mhsung}@kaist.ac.kr`

# Test-time scaling in flow matching

[Kim et al., 2025a]

Extend test-time scaling in SVDD [Li et al., 2024] to flow matching models. They propose three key ideas

- **SDE-based generation for flow matching**. Particle sampling cannot be directly applied to ODE-based generation.
- **Broadening the search space** by interpolant conversion. Flow matching originally uses linear interpolant, but change it to VP-SDE.
- **Maximize compute utilization** through Rolloever Budget Forcing.

# Test-time scaling in flow matching

[Kim et al., 2025a]

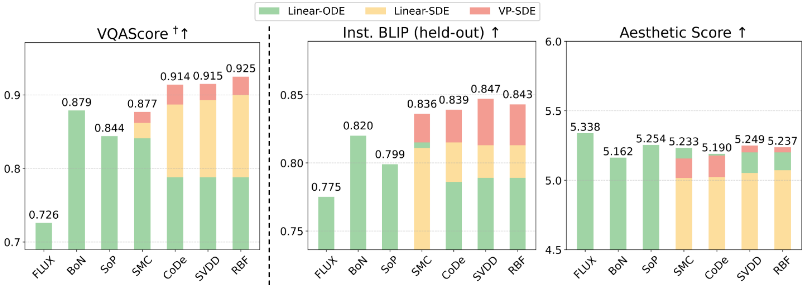



Figure 8: Performance improves from Linear-ODE to Linear-SDE, and VP-SDE.

# Test-time scaling in flow matching

[Kim et al., 2025a]

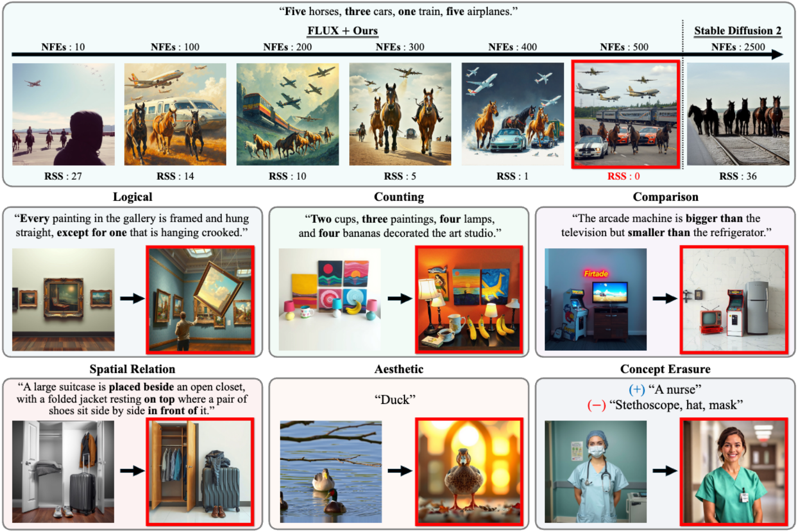



Figure 9: Diverse application of test-time scalining.

# Takeover

- Many recent works focus on enhancing inference. While we (may be only me) usually think 'efficiency', there can be various definition for 'enhancement'.
  - Parallel sampling reduce wall-clock time while maintaining computational cost.
  - Test-time scalining enables diverse conditioning and high quality image generation without training.
- I think test-time scaling is good research area that can be done in the lab.
  - It is even hard to fine-tune large models in the lab.
  - Frankly, it is hard for me to findout the reason why training fails. There are so many possibilities: algorithm or engineering (dataset, skills, architecture, ...)

# Reference I

Kim, J., Yoon, T., Hwang, J., and Sung, M. (2025a).
Inference-time scaling for flow models via stochastic generation and rollover budget forcing.
*arXiv prepring arXiv:2503.19385.*

Kim, S., Kim, M., and Park, D. (2025b).
Test-time alignment of diffusion models without reward over-optimization.
ICLR.

Li, X., Zhao, Y., Wang, C., Scalia, G., Eraslan, G., Nair, S., Biancalani, T., Ji, S., Regev, A., Levine, S., and Uehara, M. (2024).
Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding.
*arXiv preprint arXiv:2408.08252.*

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. (2024).
Direct preference optimization: Your language model is secretly a reward model.
*arXiv preprint arXiv:2305.18290.*