Understanding and Improving Training-free Loss-based Diffusion Guidance

Yifei Shen, Xinyang Jiang, Yifan Yang, Yezhen Wang, Dongqi Han, Dongsheng Li

Pengxi Liu

October 3, 2025

Introduction: What is Guidance?

- **Unconditional diffusion:** generates any realistic-looking image from the overall learned distribution.

- **Guidance:** add a condition (e.g., "dog") so samples move toward that target.

Introduction: Methods of Incorporating Guidance

Training-based guidance

Classifier Guidance → Classifier-Free Guidance

training an additional classifier

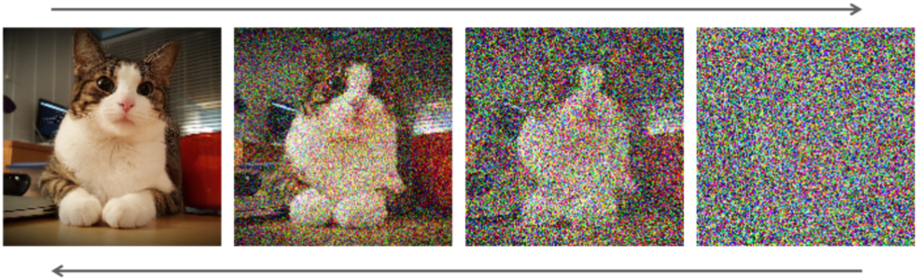
training an additional diffusion

Training-free guidance

→ **Training-free Guidance**

no extra training (zero-shot)

Background: Diffusion Models



Background: Diffusion Models

Forward Process: Transform an image into Gaussian noise.

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\,\boldsymbol{x}_0 + \sigma_t\,\boldsymbol{\epsilon}_t, \quad \alpha_t \in [0,1]\downarrow,\ \sigma_t = \sqrt{1-\alpha_t},\ \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$$

- Learn the noise (equiv. to score matching, up to a const. indep. of θ):

$$\min_\theta\ \mathbb{E}\left\|\epsilon_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}_t\right\|_2^2 \overset{\text{u.c.}}{=} \min_\theta\ \mathbb{E}\left\|\epsilon_\theta(\boldsymbol{x}_t, t) + \sigma_t \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right\|_2^2.$$

Backward Process: Convert a Gaussian noise into the image.

$$\frac{d\boldsymbol{x}_t}{dt} = f(t)\,\boldsymbol{x}_t - \frac{g(t)^2}{2}\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t) = f(t)\,\boldsymbol{x}_t + \frac{g(t)^2}{2\sigma_t}\epsilon_\theta(\boldsymbol{x}_t, t)\,.$$

$$f(t) = \frac{d}{dt}\log\sqrt{\alpha_t}, \qquad g(t)^2 = \frac{d\sigma_t^2}{dt} - 2\frac{d}{dt}\log\sqrt{\alpha_t}\,\sigma_t^2.$$

- If the learned score equals the true score, integrating this ODE from $t = T \to 0$ recovers $p_0(\boldsymbol{x}_0)$.

Background: Diffusion Guidance

Given condition (guidance) $\boldsymbol{y}$, the backward ODE becomes

$$\frac{d\boldsymbol{x}_t}{dt} = f(t)\,\boldsymbol{x}_t - \frac{g(t)^2}{2}\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t). \quad \Rightarrow \quad \frac{d\boldsymbol{x}_t}{dt} = f(t)\,\boldsymbol{x}_t - \frac{g(t)^2}{2}\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t \mid \boldsymbol{y}).$$

Now we focus on score $\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t \mid \boldsymbol{y})$.

$$\begin{aligned}
\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t \mid \boldsymbol{y}) &= \nabla_{\boldsymbol{x}_t}\log\frac{p_t(\boldsymbol{x}_t,\boldsymbol{y})}{p_t(\boldsymbol{y})} \\
&= \nabla_{\boldsymbol{x}_t}\left[\log p_t(\boldsymbol{x}_t,\boldsymbol{y}) - \log p_t(\boldsymbol{y})\right] \\
&= \nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t,\boldsymbol{y}) - \underbrace{\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{y})}_{=0} \\
&= \nabla_{\boldsymbol{x}_t}\log\left(p_t(\boldsymbol{x}_t)\,p_t(\boldsymbol{y} \mid \boldsymbol{x}_t)\right) \\
&= \nabla_{\boldsymbol{x}_t}\left[\log p_t(\boldsymbol{x}_t) + \log p_t(\boldsymbol{y} \mid \boldsymbol{x}_t)\right] \\
&= \nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{y} \mid \boldsymbol{x}_t).
\end{aligned}$$

Background: Diffusion Guidance

$$\frac{d\boldsymbol{x}_t}{dt} = f(t)\,\boldsymbol{x}_t - \frac{g(t)^2}{2}\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t \mid \boldsymbol{y}) = \underbrace{f(t)\,\boldsymbol{x}_t - \frac{g(t)^2}{2}\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t)}_{\text{unconditional}} - \boxed{\underbrace{\frac{g(t)^2}{2}\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{y} \mid \boldsymbol{x}_t)}_{\text{conditional}}}.$$

1) Classifier Guidance (training-based)

$$\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{y} \mid \boldsymbol{x}_t) := -\nabla_{\boldsymbol{x}_t}\ell\big(f_\psi(\boldsymbol{x}_t, t), \boldsymbol{y}\big)$$

$$\frac{d\boldsymbol{x}_t}{dt} = f(t)\boldsymbol{x}_t + \frac{g(t)^2}{2\sigma_t}\epsilon_\theta + \underbrace{w\,\frac{g(t)^2}{2}\nabla_{\boldsymbol{x}_t}\ell\big(f_\psi(\boldsymbol{x}_t, t), \boldsymbol{y}\big)}_{\text{guidance}}.$$

Background: Diffusion Guidance

$$\frac{d\boldsymbol{x}_t}{dt} = f(t)\,\boldsymbol{x}_t - \frac{g(t)^2}{2}\,\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t \mid \boldsymbol{y}) = \underbrace{f(t)\,\boldsymbol{x}_t + \frac{g(t)^2}{2\sigma_t}\,\epsilon_\theta(\boldsymbol{x}_t, t)}_{\text{unconditional}} + \boxed{\underbrace{\frac{g(t)^2}{2}\left(\epsilon_\theta(\boldsymbol{x}_t, t, \boldsymbol{y}) - \epsilon_\theta(\boldsymbol{x}_t, t)\right)}_{\text{conditional}}}.$$

2) Classifier-Free Guidance (training-based) Train a single noise predictor $\epsilon_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$ to handle both *conditional* and *unconditional* cases via condition dropout. The loss is the standard noise-prediction MSE:

$$\min_\theta\ \mathbb{E}\left\|\epsilon - \epsilon_\theta(\boldsymbol{x}_t, t, \tilde{\boldsymbol{c}})\right\|^2, \qquad \tilde{\boldsymbol{c}} \in \{\boldsymbol{y}, \varnothing\}.$$

$$\epsilon_{\text{cond}} = \epsilon_\theta(\boldsymbol{x}_t, t, \boldsymbol{y}), \quad \epsilon_{\text{uncond}} = \epsilon_\theta(\boldsymbol{x}_t, t, \varnothing),$$

$$\hat{\epsilon} = \epsilon_{\text{uncond}} + w\left(\epsilon_{\text{cond}} - \epsilon_{\text{uncond}}\right) \quad \Rightarrow \quad \frac{d\boldsymbol{x}_t}{dt} = f(t)\boldsymbol{x}_t + \frac{g(t)^2}{2\sigma_t}\,\hat{\epsilon}.$$

Background: Diffusion Guidance

$$\frac{d\boldsymbol{x}_t}{dt} = f(t)\,\boldsymbol{x}_t - \frac{g(t)^2}{2}\,\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t \mid \boldsymbol{y}) = \underbrace{f(t)\,\boldsymbol{x}_t - \frac{g(t)^2}{2}\,\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t)}_{\text{unconditional}} - \boxed{\underbrace{\frac{g(t)^2}{2}\,\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{y} \mid \boldsymbol{x}_t)}_{\text{conditional}}}.$$

3) Training-free (loss-based) Guidance

$$\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t] = \frac{\boldsymbol{x}_t - \sigma_t\,\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\alpha_t}}, \quad \nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{y} \mid \boldsymbol{x}_t) := -\nabla_{\boldsymbol{x}_t}\,\ell\big(f_\phi(\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]), \boldsymbol{y}\big),$$

$$\frac{d\boldsymbol{x}_t}{dt} = f(t)\boldsymbol{x}_t + \frac{g(t)^2}{2\sigma_t}\epsilon_\theta - \underbrace{w\,\frac{g(t)^2}{2}\,\nabla_{\boldsymbol{x}_t}\,\ell\big(f_\phi(\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]), \boldsymbol{y}\big)}_{\text{guidance}}.$$

Note: f_ϕ is a *discriminative/assessment* network trained on **clean images** x_0.

Training-Free Guidance: How Does It Work?

$$\text{True gradient of score: } \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y} \mid \boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log \mathbb{E}_{p(\boldsymbol{x}_0|\boldsymbol{x}_t)}\Big[e^{-\ell(f_\phi(\boldsymbol{x}_0),\boldsymbol{y})}\Big]$$

$$\neq$$

$$\text{Training-free replacement: } \nabla_{\boldsymbol{x}_t} \log \Big[e^{-\ell(f_\phi(\mathbb{E}_{p(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]),\boldsymbol{y})}\Big] = -\nabla_{\boldsymbol{x}_t} \ell\Big(f_\phi\big(\underbrace{\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]}_{=(\boldsymbol{x}_t - \sigma_t \boldsymbol{\epsilon}_\theta)/\sqrt{\alpha_t}}\big), \boldsymbol{y}\Big).$$

- One option: use Gaussian surrogate $q(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) = \mathcal{N}(\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t], r_t^2 I)$ and Monte-Carlo average

$$\nabla_{\boldsymbol{x}_t} \log \mathbb{E}_q\Big[e^{-\ell(f_\phi(\boldsymbol{x}_0),\boldsymbol{y})}\Big] \approx \nabla_{\boldsymbol{x}_t} \log \frac{1}{n}\sum_{i=1}^{n} e^{-\ell(f_\phi(\boldsymbol{x}_0^i),\boldsymbol{y})}.$$

Criticism: For sub-Gaussian q, mass concentrates on a thin spherical shell of radius r_t around $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$; the true posterior $p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)$ lies on a curved low-d manifold. Supports may not overlap $\Rightarrow$ biased/misaligned gradients.

Training-Free Guidance: How Does It Work?

- This paper: **optimization** view.

- Ideal but intractable: $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y} \mid \boldsymbol{x}_t)$.
- Practical controller: use the *surrogate potential* $v_{\mathrm{TF}}(\boldsymbol{x}_t) = -\ell(f_\phi(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)), \boldsymbol{y})$ and plug its gradient into the ODE.

$$\ell_t(\boldsymbol{x}_t) := \ell\Big(f_\phi\Big(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t, t)\Big), \boldsymbol{y}\Big), \qquad \hat{\boldsymbol{x}}_0(\boldsymbol{x}_t, t) = \frac{\boldsymbol{x}_t - \sigma_t\, \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\alpha_t}} \text{ (Tweedie mean).}$$

Exact gradient via Tweedie variance form: $\nabla_{\boldsymbol{x}_t} \ell_t(\boldsymbol{x}_t) = \underbrace{\frac{\partial \ell}{\partial \hat{\boldsymbol{x}}_0}}_{\text{clean-domain loss grad}} \cdot \underbrace{\frac{\partial \hat{\boldsymbol{x}}_0}{\partial \boldsymbol{x}_t}}_{= \frac{\mathrm{Cov}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]}{\sigma_t^2 \sqrt{\alpha_t}}}.$

One guidance step (control update): $\hat{\boldsymbol{x}}_t = \boldsymbol{x}_t - \eta_t \nabla_{\boldsymbol{x}_t} \ell_t(\boldsymbol{x}_t)$, with $\eta_t = \frac{\sqrt{\alpha_t}}{L_f(1+L_p)}$.

Training-Free Guidance: How Does It Work?

Proposition 3.1 (informal). Define $\ell_t(\boldsymbol{x}_t) = \ell(f_\phi(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)), \boldsymbol{y})$ with $\hat{\boldsymbol{x}}_0 = \frac{\boldsymbol{x}_t - \sigma_t \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\alpha_t}}$. Under PL $(\|\nabla f(\boldsymbol{x})\|^2 \geq \mu f(\boldsymbol{x}))$ + Lipschitz $(\|f(\boldsymbol{x}_2) - f(\boldsymbol{x}_1)\| \leq L \|\boldsymbol{x}_2 - \boldsymbol{x}_1\|)$ assumptions, the one-step update

$$\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t - \eta_t \nabla_{\boldsymbol{x}_t} \ell_t(\boldsymbol{x}_t), \qquad \eta_t \propto \sqrt{\alpha_t},$$

guarantees a *per-step decrease* of ℓ_t at late timesteps (small σ_t).

Why this helps.

- It certifies the guidance is *contractive*.
- It yields a principled schedule for the guidance weight/step size ($\eta_t \propto \sqrt{\alpha_t}$): small early, larger late.

Training-Free Guidance: Limitations

Training-free guidance injects the gradient of an off-the-shelf network into sampling:

$$\boldsymbol{g}_t^{\mathrm{TF}} = -\nabla_{\boldsymbol{x}_t} \ell\Big(f_\phi(\hat{\boldsymbol{x}}_0), \boldsymbol{y}\Big), \qquad \hat{\boldsymbol{x}}_0 = \frac{\boldsymbol{x}_t - \sigma_t\, \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\alpha_t}}.$$

Issue: misaligned (adversarial) gradients.

- Off-the-shelf f_ϕ (trained on clean images) can have **high Lipschitz gain** on $\hat{\boldsymbol{x}}_0 \Rightarrow$ small input changes cause large loss changes.
- The added gradient may *reduce the loss* yet be **not aligned with the intended semantic direction** (adversarial-style behavior) $\Rightarrow$ oscillations / off-target updates early in sampling.

Contrast vs. training-based guidance.

- Time-dependent classifiers are trained on *noisy* inputs x_t; the induced smoothing lowers sensitivity $\Rightarrow$ gradients align better with semantics.

Training-Free Guidance: Limitations

Prop. 3.2: noise-augmented (time-dependent) networks are smoother. If $\ell(\boldsymbol{x}) \leq C$, define the smoothed loss

$$\hat{\ell}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0,\boldsymbol{I})}[\ell(\boldsymbol{x} + \sigma_t \boldsymbol{\epsilon})] .$$

Then $\hat{\ell}$ is $C\sqrt{\frac{2}{\pi\sigma_t^2}}$-Lipschitz and its gradient is $\frac{2C}{\sigma_t}$-Lipschitz.

Intuition (Gaussian smoothing).

- Convolving with $\mathcal{N}(0, \sigma_t^2 I)$ *averages out* high-frequency/fragile directions.
- Larger $\sigma_t \Rightarrow$ smaller effective Lipschitz constants $\Rightarrow$ **more robust, better-aligned gradients**.

Training-Free Guidance: Limitations

Prop. 3.3: smaller Lipschitz $L \Rightarrow$ faster, stabler ODE solving.

$$\frac{d\boldsymbol{x}_t}{dt} = f(t)\boldsymbol{x}_t + \frac{g(t)^2}{2}u(\boldsymbol{x}_t, t), \quad u(\boldsymbol{x}_t, t) = \frac{1}{\sigma_t}\epsilon_\theta(\boldsymbol{x}_t, t) + \nabla_{\boldsymbol{x}_t} v(\boldsymbol{x}_t, t).$$

Let L be the Lipschitz constant of u, and $M = O(1/h_{\text{max}})$ with $h_{\text{max}} = \max_t \frac{1}{2}\left[\log \frac{\alpha_t}{1-\alpha_t} - \log \frac{\alpha_{t-1}}{1-\alpha_{t-1}}\right]$.
Then the global error obeys

$$\left\| \boldsymbol{x}_0 - \boldsymbol{x}_0^\star \right\| = O\left(\frac{1 + L^M}{M}\right).$$

Implications.

- **Smaller** L (smoother guidance) $\Rightarrow$ **lower discretization error**, **fewer NFEs**.
- Training-free (off-the-shelf) tends to have **larger** L than time-dependent guidance $\Rightarrow$ slower convergence; late-stage scheduling and light smoothing help.

Improving Training-free Guidance

Goal. Mitigate (i) misaligned gradients and (ii) slow ODE convergence in training-free guidance.

Two complementary techniques.

- **Random Augmentation (RA)**: replace pure Gaussian smoothing with a small set of differentiable data augmentations $T \in \mathcal{T}$ applied to $\hat{\boldsymbol{x}}_0$ before computing the loss.
- **Polyak Step Size (PSS)**: adapt the per-step guidance magnitude using a Polyak-style rule to balance diffusion vs. guidance and accelerate convergence.

Where they act in the pipeline.

$$\frac{dx_t}{dt} = f(t)\,x_t + \frac{g(t)^2}{2}\Big(\epsilon_\theta(x_t,t) + \nabla_{x_t} v(x_t,t)\Big), \quad \nabla_{x_t} v \approx -\,g_t,\; g_t := \nabla_{x_t}\ell\big(f_\phi(\hat{x}_0), y\big),\; \hat{x}_0 = \tfrac{x_t - \sigma_t \epsilon_\theta}{\sqrt{\alpha_t}}.$$

RA modifies the loss to $\ell(f_\phi(T(\hat{\boldsymbol{x}}_0)), \boldsymbol{y})$; PSS modifies the step length applied to g_t.

Random Augmentation (RA): smoother and better-aligned gradients

Algorithm.

Algorithm 1 Random Augmentation

for $t = T, \cdots, 0$ **do**
  $\boldsymbol{x}_{t-1} = \mathrm{DDIM}(\boldsymbol{x}_t)$
  $\hat{\boldsymbol{x}}_0 = \frac{\boldsymbol{x}_t - \sigma_t \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\alpha_t}}$ ▷ Tweedie's formula
  $\boldsymbol{g}_t = \frac{1}{|\mathcal{T}|} \sum_{T \in \mathcal{T}} \nabla_{\boldsymbol{x}_t} \ell(f_\phi(T(\hat{\boldsymbol{x}}_0)), \boldsymbol{y})$
  $\boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1} - \eta \cdot \boldsymbol{g}_t$
end for

- Like Gaussian smoothing, RA *averages out high-frequency/fragile directions*, but does so in a *lower-dimensional, task-relevant augmentation subspace* (resize, crop, color, cutout, ...), requiring fewer samples.
- Reduces effective Lipschitz constants of the loss; gradients align better with semantics (less misalignment).

Proposition 4.1 (informal). If ℓ is bounded, the augmented loss $\hat{\ell}(\boldsymbol{x}) = \mathbb{E}_{e \sim p(e)}[\ell(\boldsymbol{x} + e)]$ is $C \int \|\nabla p(t)\|_2 dt$-Lipschitz and its gradient is $C \int \|\nabla^2 p(t)\|_{\mathrm{op}} dt$-Lipschitz. $\Rightarrow$ RA improves smoothness & robustness of the guidance field.

Polyak Step Size (PSS): adaptive guidance magnitude

Fixed guidance may under/overshoot: small guidance under-steers when unconditional drift is strong; large guidance can oscillate when initialization deviates from the condition.

Algorithm.

Algorithm 2 Polyak Step Size

for $t = T, \cdots, 0$ **do**

$\boldsymbol{x}_{t-1} = \mathrm{DDIM}(\boldsymbol{x}_t)$

$\hat{\boldsymbol{x}}_0 = \frac{\boldsymbol{x}_t - \sigma_t \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\alpha_t}}$ ▷ Tweedie's formula

$\boldsymbol{g}_t = \nabla_{\boldsymbol{x}_t} \ell(f_\phi(\hat{\boldsymbol{x}}_0), \boldsymbol{y})$

$\boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1} - \eta \cdot \frac{\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\|}{\|\boldsymbol{g}_t\|_2^2} \cdot \boldsymbol{g}_t$

end for

- In short: take a DDIM step (unconditional), then apply a Polyak–step gradient correction (guidance).
- Polyak step size: $\eta_t \propto \|\epsilon_\theta\| \,/\, \|g_t\|^2$, i.e., "take larger steps when far, smaller steps when close, and smaller steps on steep slopes." This echoes Prop. 3.1's claim that later timesteps can use larger steps, and numerically it speeds convergence while reducing early-stage oscillations.

Experiments: Overview

- Training-free guidance enhanced by **Random Augmentation (RA)** and **Polyak Step Size (PSS)**. Our method = FreeDoM + PSS + RA. Time-travel trick (restart sampling) as in prior work.
- Compare to baselines: Universal Guidance (UG), Loss-Guided Diffusion with MC (LGD-MC), FreeDoM (training-free), MPGD-Z.

Tasks.

1. CelebA-HQ face diffusion: segmentation / sketch / text guidance.
2. ImageNet diffusion: zero-shot text guidance (CLIP-based).
3. Human motion diffusion (MDM): target reaching & obstacle avoidance.

Experiments: Results

Methods	Segmentation maps		Sketches		Texts	
	Distance↓	FID↓	Distance↓	FID↓	Distance↓	FID↓
UG [2]	2247.2	39.91	52.15	47.20	12.08	44.27
LGD-MC [37]	2088.5	38.99	49.46	54.47	11.84	41.74
FreeDoM [49]	1657.0	38.65	34.21	52.18	11.17	46.13
MPGD-Z [16]	1976.0	39.81	37.23	54.18	10.78	42.45
Ours	**1575.7**	**33.31**	**30.41**	**41.26**	**10.72**	**41.25**

Table 1: The performance comparison of various methods on CelebA-HQ with different types of zero-shot guidance. The experimental settings adhere to Table 1 of [49].

Experiments: Results

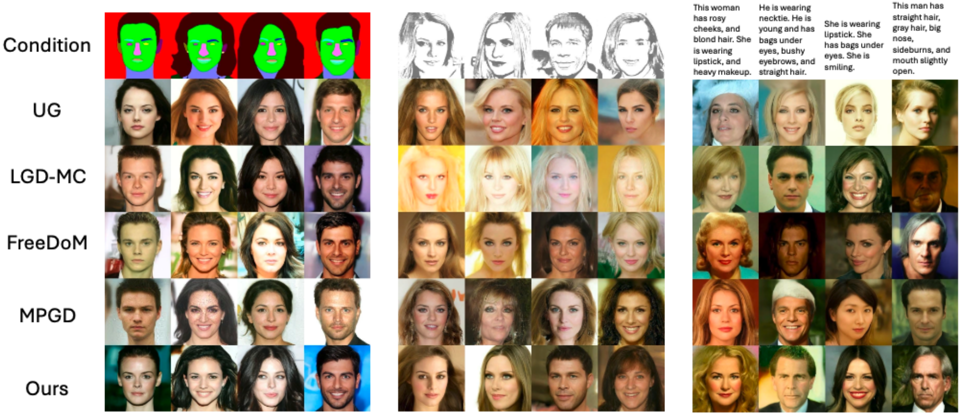


Figure 4: Qualitative results of CelebA-HQ with zero-shot segmentation, sketch, and text guidance. The images are randomly selected.

Experiments: Results

Methods	LGD-MC [37]	UG [2]	FreeDoM [49]	MPGD-Z [16]	Ours
CLIP Score↑	24.3	25.7	25.9	25.1	**27.7**

Table 2: The performance comparison of various methods on unconditional ImageNet with zero-shot text guidance. We compare various methods using ImageNet pretrained diffusion models with CLIP-B/16 guidance. For evaluating performance, the CLIP score is computed using CLIP-L/14.

Experiments: Results

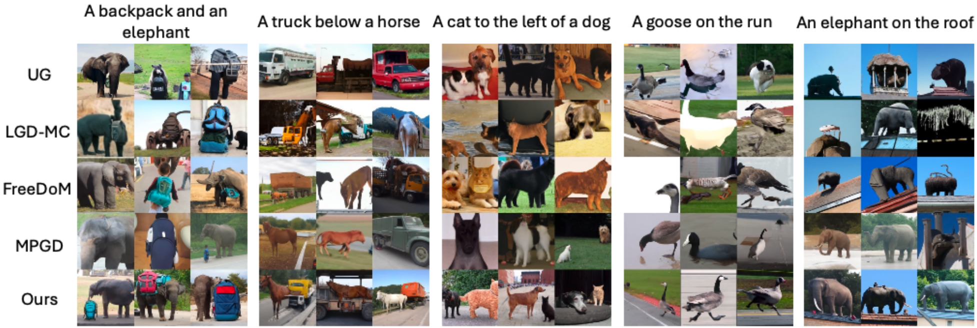


Figure 5: Qualitative results of ImageNet model with zero-shot text guidance. The images are randomly selected.

Experiments: Results

Methods	"Backwards"		"Balanced Beam"		"Walking"		"Jogging"	
	Loss↓	CLIP↑	Loss↓	CLIP↑	Loss↓	CLIP↑	Loss↓	CLIP↑
Unconditional [40]	$3.55 + 9.66$	65.6	$47.92 + 0$	**70.8**	$48.88 + 0$	37.6	$144.84 + 0$	**61.72**
FreeDoM [49]	$1.09 + 6.63$	67.23	$9.83 + 4.48$	62.65	$1.64 + 7.55$	40.12	$34.95 + 7.83$	58.74
LGD-MC [37]	$0.98 + 6.48$	67.31	$4.42 + 0.02$	63.13	$1.30 + 0.39$	38.82	$6.12 + 2.38$	57.89
Ours	**0.68+1.32**	**67.50**	**1.13+0.30**	63.02	**0.43+0.31**	**40.40**	**2.93+1.15**	60.03

Table 3: Comparison of various methods on MDM with zero-shot targeting and object avoidance guidance. Loss is reported as a two-component metric: the first part is the MSE between the target and the actual final position of the individual; the second part measures the object avoidance loss.

Experiments: Results

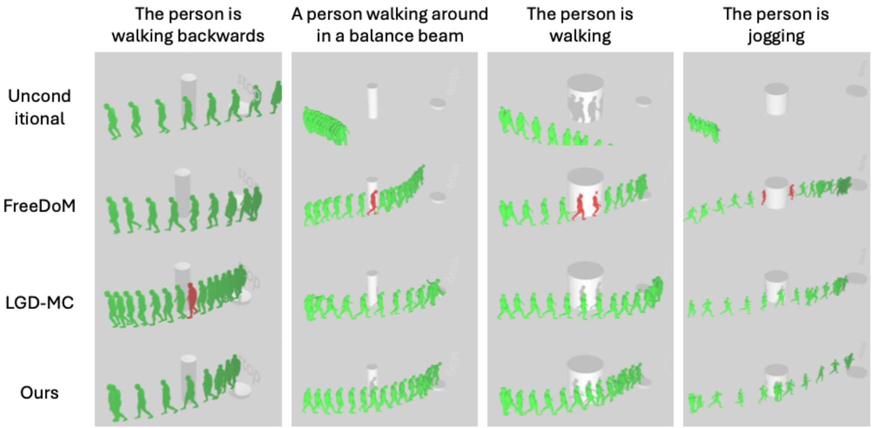


Figure 6: Qualitative results of human motion diffusion with zero-shot object avoidance and targeting guidance. Instances of intersection with obstacles are highlighted by marking the person in red. The trajectories are randomly selected.

Accommodation

Is it worth reading? **Yes!**

- Learn mechanisms of diffusion guidance: classifier guidance, classifier-free Guidance, training-free guidance.
- Learn how optimization view reshapes ODE behavior.
- Learn practical role of some mathematical concepts such as L-Lipschitz, PL condition, Gaussian smoothing / augment averaging in stabilizing sampling.

Is it worth implementing? **Maybe!**

- Where it fits: training-free guidance can steer structured generators (e.g., time-series EHR) and imaging (MRI/CT) using pretrained task nets.