ERROR AS SIGNAL: STIFFNESS-AWARE DIFFUSION SAMPLING VIA EMBEDDED RUNGE-KUTTA GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Classifier-Free Guidance (CFG) has established the foundation for guidance mechanisms in diffusion models, showing that well-designed guidance proxies significantly improve conditional generation and sample quality. Autoguidance (AG) has extended this idea, but it relies on an auxiliary network and leave solver-induced errors unaddresed. In stiff regions, the ODE trajectory changes sharply, where local truncation error (LTE) becomes a critical factor to deteriorate sample quality. Our key observation is that these errors align with the dominant eigenvector, motivating us to target the solver-induced error as a guidance signal. We propose **E**mbedded **R**unge–**K**utta **Guid**ance (ERK-Guid), which exploits detected stiffness to reduce LTE and stabilize sampling. We theoretically and empirically analyze stiffness and eigenvector estimators with solver errors to motivate the design of ERK-Guid. Our experiments on both synthetic datasets and popular benchmark dataset, ImageNet, demonstrate that ERK-Guid consistently outperforms state-of-the-art methods.

1 INTRODUCTION

Generative models Kingma & Welling (2014); Rezende & Mohamed (2015); Heusel et al. (2017); Lipman et al. (2023); Ho et al. (2020); Song et al. (2021b); Goodfellow et al. (2020) aim to approximate complex data distributions and generate new samples, enabling a wide range of applications in image synthesis Brock et al. (2019); Rombach et al. (2022); Zhang et al. (2023); Kawar et al. (2023), editing Brooks et al. (2023), and video generation Gupta et al. (2024). Among them, diffusion models Ho et al. (2020); Song et al. (2021b); Karras et al. (2022) have emerged as a dominant paradigm, achieving strong performance across diverse generation tasks Gupta et al. (2024); Karras et al. (2024a). They define a forward process that gradually perturbs data into Gaussian noise, while a neural network is trained to predict the score function of each noisy distribution. This score estimate parameterizes the reverse-time dynamics, allowing data to be reconstructed through iterative denoising. Sampling is commonly formulated as solving an ordinary differential equation (ODE) or stochastic differential equation (SDE), where the drift is defined by the learned network Karras et al. (2022); Song et al. (2021a); Lu et al. (2022). As a result, the quality of generated samples depends not only on model accuracy but also on the numerical solver used to approximate the reverse dynamics, which can significantly influence fidelity and stability.

Guidance mechanisms emerged to improve both sampling fidelity and perceptual quality by introducing suitable proxies for steering the sampling trajectory. The de facto standard, Classifier-Free Guidance (CFG) Ho & Salimans (2022), combines unconditional and conditional predictions to strengthen alignment and enhance image quality. Predictor-Corrector Guidance (PCG) Bradley & Nakkiran (2024) further refines this view by interpreting CFG as a predictor–corrector update that extrapolates between these predictions. Autoguidance (AG) Karras et al. (2024a) follows a similar principle by contrasting outputs from models of different capacities, using their discrepancies to identify regions where model-induced errors are significant. However, subsequent methods Kynkäänniemi et al. (2024); Sadat et al. (2024); Zheng & Lan (2024); Zhao et al. (2025), rely solely on such model-based differences, overlooking the numerical errors arising from the solver itself as potential guidance signals. We observe that, in stiff regions of the diffusion ODE, the solver's local truncation error (LTE) aligns with the dominant eigenvector of the drift, revealing a numerically grounded proxy distinct from model-space signals and motivating our approach.

In this work, we propose **E**mbedded **R**unge–**K**utta based **Guid**ance (**ERK-Guid**), a novel approach that mitigates solver-induced local truncation error (LTE) during diffusion sampling by estimating the dominant eigenvector in stiff regions. In diffusion ODEs, stiffness arises when drift directions change rapidly, and we observe that the resulting LTE consistently aligns with the dominant eigenvector under such conditions. To exploit this property, we introduce two cost-free estimators derived from the Embedded Runge–Kutta (ERK) formulation: *ERK solution difference* (between Heun and Euler solutions) and *ERK drift difference* (between their corresponding drifts). The ratio of their norms serves as **a stiffness estimator** to identify regions of high stiffness, while the ERK drift difference further provides **an eigenvector estimator** to approximate the dominant eigenvector direction. ERK-Guid transforms theoretical insights on stiffness and the dominant eigenvector alignment into a stable, cost-free guidance mechanism that effectively reduces solver-induced errors. Ultimately, ERK-Guid applies guidance along this estimated direction, offering a stable and effective proxy that reduces solver-induced errors without additional computational cost.
Our contributions are summarized as follows:

- We introduce the Embedded Runge-Kutta Guidance (ERK-Guid), a stiffness-aware guidance method that leverages solver errors as informative signals for diffusion sampling.
- We propose cost-free estimators for stiffness detection and dominant eigenvector estimation, derived from ERK solution and drift differences to determine the guidance direction.
- We design a stabilized guidance scheme that bridges theoretical insights with practical robustness, ensuring robustness without additional network evaluations.
- We demonstrate through synthetic and ImageNet experiments that ERK-Guid delivers an orthogonal guidance signal and consistently improves over strong baselines.

2 Related Works

ODE solvers in diffusion models. A major research trend in diffusion models has been to accelerate sampling by improving ODE solvers that approximate the underlying probability flow dynamics more efficiently. Early work such as DDIM Song et al. (2021a) reinterprets the stochastic sampling process of DDPM Ho et al. (2020) as a deterministic ODE trajectory, enabling significantly fewer sampling steps without retraining. Building on this view, PNDM Liu et al. (2022) introduces a pseudo-numerical multistep method that generalizes DDIM beyond first-order updates. Subsequent studies further leverage the structure of diffusion dynamics. DEIS Zhang & Chen (2023) employs exponential integrators to reduce discretization error, DPM-Solver Lu et al. (2022) derives high-order solvers with coefficients designed to minimize local truncation error, and UniPC Zhao et al. (2023) unifies predictor–corrector schemes under a single framework. More recently, DPM-Solver-v3 Zheng et al. (2023) incorporates empirical model statistics into solver parameterization to jointly address discretization and model approximation errors. Bespoke solver approaches develop customized solver designs for a fixed pre-trained model, encompassing methods that optimize time-step schedules Xue et al. (2024) and solver parameters Wang et al. (2025); Shaul et al. (2024). In contrast to these approaches, which redesign or replace the ODE solver, ERK-Guid operates in a fundamentally different regime. We keep the solver fixed and instead leverage the solver's own error: the discrepancy between low- and high-order solver updates. By using this discrepancy as a directional correction, ERK-Guid provides solver-aware guidance without modifying the solver's numerical structure.

Adaptive guidance computation. In diffusion models, Classifier-Free Guidance (CFG) Ho & Salimans (2022) has become the de facto standard for improving fidelity and condition alignment by contrasting conditional and unconditional denoisers. Despite its success, CFG often suffers from overshoot, loss of diversity, and entangled fidelity–diversity trade-offs, limiting its flexibility across noise levels. Autoguidance (AG) Karras et al. (2024a) improves robustness by replacing the unconditional branch with a weaker model, correcting model-induced errors without sacrificing variation. Beyond these canonical approaches, several adaptive guidance strategies have been explored. Guidance Interval Kynkäänniemi et al. (2024) activates guidance only at mid-range noise levels. Other works Sadat et al. (2024); Zheng & Lan (2024) mitigate oversaturation under strong guidance, and

DyDiT Zhao et al. (2025) dynamically adjusts model capacity across timesteps. These advances demonstrate how CFG has shaped a broad family of model-based guidance mechanisms. In contrast, ERK-Guid employs a solver-driven proxy derived from ERK discrepancies, yielding an orthogonal guidance signal yet complementary to model-based guidance.

3 Preliminaries

Denoising Diffusion Models. Denoising diffusion models Ho et al. (2020); Song et al. (2021b); Karras et al. (2022) generate samples by simulating the reverse-time dynamics of a predefined stochastic differential equation (SDE). The SDE gradually transforms the data distribution p_{data} into a perturbed distribution $p(\mathbf{x};\sigma)$. Following EDM2 Karras et al. (2024b), the perturbed distribution is defined as the convolution of p_{data} with Gaussian noise Kynkäänniemi et al. (2024), i.e., $p(\mathbf{x};\sigma) = p_{\text{data}}(\mathbf{x}) * \mathcal{N}(\mathbf{x};\mathbf{0},\sigma^2\mathbf{I})$, where $\sigma \in [0,\sigma_{\text{max}}]$.

The reverse-time SDE can be equivalently reformulated as an ordinary differential equation (ODE) Song et al. (2021b), leading to a deterministic sampling process $\mathbf{x}_0 \sim p_{\text{data}}$, that solves the following initial value problem:

$$\frac{\mathrm{d}\mathbf{x}_\sigma}{\mathrm{d}\sigma} = \boldsymbol{f}(\mathbf{x}_\sigma;\sigma) = -\sigma\nabla_{\mathbf{x}_\sigma}\log p(\mathbf{x}_\sigma;\sigma), \quad \mathbf{x}_0 = \mathbf{x}_{\sigma_{\text{max}}} + \int_{\sigma_{\text{max}}}^{0}\left(\frac{\mathrm{d}\mathbf{x}_\sigma}{\mathrm{d}\sigma}\right)\mathrm{d}\sigma, \tag{1}$$

where $\boldsymbol{f}(\mathbf{x}_\sigma;\sigma)$ denotes the drift function of the ODE, and $\mathbf{x}_\sigma$ refers to the trajectory of the sample as a function of the noise level σ. The drift function is typically approximated by the learned model $\boldsymbol{f}_\theta(\mathbf{x}_\sigma;\sigma)$, which is trained via score-matching objectives Song et al. (2021b). Note that the initial state $\mathbf{x}_{\sigma_{\text{max}}}$ can be approximately sampled from $\mathcal{N}(\mathbf{0},\sigma_{\text{max}}^2\mathbf{I})$ when σ_{max} is sufficiently large.

In practice, the ODE cannot be solved analytically; numerical solvers are employed. To enable numerical integration, EDM2 Karras et al. (2024b) discretizes the interval into a sequence of noise levels $\{\sigma_0,\ldots,\sigma_N\}$, where N is the total number of integration steps. Note that $[\sigma_i,\sigma_{i+1}]$ denotes the i-th integration interval, with $\sigma_0 = \sigma_{\text{max}}$ and $\sigma_N = 0$. A numerical solver is then applied to each interval to approximate the following integration:

$$\mathbf{x}_{\sigma_{i+1}} = \mathbf{x}_{\sigma_i} + \int_{\sigma_i}^{\sigma_{i+1}} \boldsymbol{f}(\mathbf{x}_\sigma;\sigma)\mathrm{d}\sigma. \tag{2}$$

The Euler method Hairer et al. (1993), a widely used first-order solver, provides the following update and local truncation error (LTE):

$$\mathbf{x}^{\text{Euler}}_{\sigma_{i+1}} = \mathbf{x}_{\sigma_i} - h\boldsymbol{f}(\mathbf{x}_{\sigma_i};\sigma_i), \quad \text{LTE}^{\text{Euler}} = \mathbf{x}_{\sigma_{i+1}} - \mathbf{x}^{\text{Euler}}_{\sigma_{i+1}} = \mathcal{O}(h^2), lk \tag{3}$$

where $h := \sigma_i - \sigma_{i+1} > 0$ refers the step size. To reduce the local truncation error, higher-order solvers are commonly used. Heun's method Hairer et al. (1993), based on the trapezoidal rule Hairer et al. (1993), introduces a correction to the Euler estimate and effectively incorporates implicit integration. Its update and LTE are given as follows:

$$\mathbf{x}^{\text{Heun}}_{\sigma_{i+1}} = \mathbf{x}_{\sigma_i} - \frac{h}{2}\big(\boldsymbol{f}(\mathbf{x}_{\sigma_i};\sigma_i) + \boldsymbol{f}(\mathbf{x}^{\text{Euler}}_{\sigma_{i+1}};\sigma_{i+1})\big), \quad \text{LTE}^{\text{Heun}} = \mathbf{x}_{\sigma_{i+1}} - \mathbf{x}^{\text{Heun}}_{\sigma_{i+1}} = \mathcal{O}(h^3). \tag{4}$$

This can be interpreted as a second-order Runge–Kutta method and serves as the default solver throughout our work.

Embedded Runge–Kutta pair. In Heun's method, the Euler solution is computed first and then corrected. Thus, we obtain two solutions of different orders (Euler of order 1 and Heun of order 2) within a single step. This structure is referred to as an *embedded Runge–Kutta pair*, and their solution difference $\Delta^{\mathbf{x}} = \mathbf{x}^{\text{Heun}}_{\sigma_{i+1}} - \mathbf{x}^{\text{Euler}}_{\sigma_{i+1}}$ is commonly used as a proxy for the local truncation error Hairer et al. (1993). In this paper, we refer to this difference as the **ERK solution difference**. We also define the **ERK drift difference** as the difference of the drift evaluated at the two solutions, $\Delta^{\mathbf{f}} := \boldsymbol{f}(\mathbf{x}^{\text{Heun}}_{\sigma_{i+1}};\ \sigma_{i+1}) - \boldsymbol{f}(\mathbf{x}^{\text{Euler}}_{\sigma_{i+1}};\ \sigma_{i+1})$.

Stiffness. Stiffness refers to the presence of both fast and slow dynamics within an ODE system Hairer & Wanner (1996). It is commonly encountered in physical simulations such as fluid dynamics. To ensure stability, numerical solvers are forced to reduce their step sizes, leading to increased function evaluations and higher computational cost. To handle this, previous adaptive

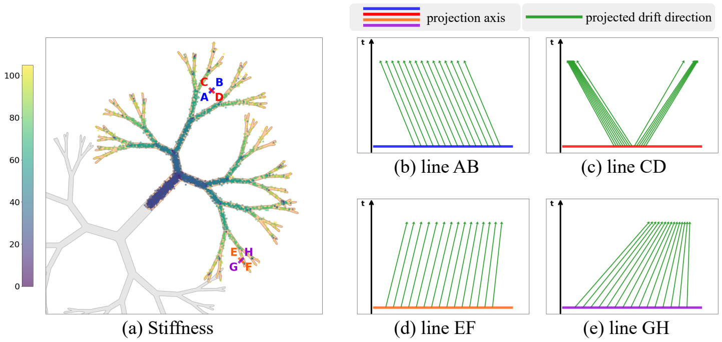


Table 1: **Quantitative comparison of local truncation error (LTE) and ERK solution difference.**

	LTE ↓	ERK solution diff ↓
AB	5.78e-06	6.18e-05
CD	7.22e-05	4.03e-04
Ratio	**12.5×**	**6.52×**

Figure 1: **Toy 2D example of stiffness and projected drift directions. (a)** We plot generated samples at the 28th step (out of 32) over the ground-truth distribution, colored by stiffness magnitude. **(b, d)** The projected drift directions remain nearly parallel, indicating low LTE and stable dynamics. **(c, e)** The drift directions spread into multiple orientations, reflecting high LTE and numerical instability. Table 1 compares (b) and (c), showing that parallel directions (AB) yield small errors, whereas divergent ones (CD) exhibit much larger LTE and ERK solution differences (12.5× and 6.52×, respectively).

solvers Petzold (1983); Shampine & Gear (1979) detect stiffness and dynamically adjust their integration behavior by refining step sizes or switching to implicit solvers.

Classically, stiffness is quantified by spectral properties of the Jacobian of the drift, such as the ratio between the largest and smallest eigenvalue magnitudes or, more simply, the maximum eigenvalue in magnitude Hairer & Wanner (1996).

$$J(\mathbf{x}_\sigma, \sigma) := \nabla_{\mathbf{x}} \boldsymbol{f}(\mathbf{x}_\sigma;\, \sigma)\,, \tag{5}$$

$$\rho_{\text{stiff}}(\mathbf{x}_\sigma, \sigma) := \max_k \left|\lambda_k\big(J(\mathbf{x}_\sigma, \sigma)\big)\right|, \tag{6}$$

where $\lambda_k(J)$ denotes the k-th eigenvalue of the matrix J. We denote by $\mathbf{v}_{\text{stiff}}(\mathbf{x}_\sigma, \sigma)$ a unit dominant-eigenvector associated with $\rho_{\text{stiff}}(\mathbf{x}_\sigma, \sigma)$.

4 METHOD

We first provide theoretical and experimental intuition that, in stiff ODEs, both the local truncation error (LTE) and the embedded Runge–Kutta (ERK) solution difference are aligned with the Jacobian's dominant eigenvector (Section 4.1). Based on this observation, we introduce cost-free estimators for stiffness and the dominant eigenvector (Section 4.2), and incorporate them into our guidance scheme, ERK-Guid (Section 4.3).

4.1 ALIGNMENT OF LTE AND ERK SOLUTION DIFFERENCES IN STIFF ODES

Theoretical Insight. We assume that the score-based vector field of the diffusion model $\mathbf{x}_{\sigma_i}$ is well approximated by its local linearization around the current state $\mathbf{x}_{\sigma_i}$ when the step size $h := \sigma_i - \sigma_{i+1} > 0$ is sufficiently small:

$$\frac{\mathrm{d}\mathbf{x}_\sigma}{\mathrm{d}\sigma} = \boldsymbol{f}(\mathbf{x}_\sigma;\sigma) \approx \boldsymbol{f}(\mathbf{x}_{\sigma_i};\sigma_i) + J(\mathbf{x}_{\sigma_i};\sigma_i)\,(\mathbf{x}_\sigma - \mathbf{x}_{\sigma_i}). \tag{7}$$

Let $J_{\mathbf{x}_{\sigma_i}}$ and $\mathbf{f}_{\mathbf{x}_{\sigma_i}}$ denote $J(\mathbf{x}_{\sigma_i};\sigma_i)$ and $\boldsymbol{f}(\mathbf{x}_{\sigma_i};\sigma_i)$, respectively. From Eq. 1, the Jacobian is given by $J_{\mathbf{x}_{\sigma_i}} = -\sigma_i \nabla^2_{\mathbf{x}} \log p(\mathbf{x}_{\sigma_i};\sigma_i)$. Since $J_{\mathbf{x}_{\sigma_i}}$ is the Hessian of $\log p(\mathbf{x}_{\sigma_i};\sigma_i)$ under C^2-smoothness, it is symmetric and therefore admits an eigendecomposition as follows:

$$J_{\mathbf{x}_{\sigma_i}} = V\Lambda V^\top, \text{ s.t. } J_{\mathbf{x}_{\sigma_i}}\mathbf{v}_k = \lambda_k \mathbf{v}_k,\ \|\mathbf{v}_k\|_2 = 1, \forall k \tag{8}$$

Therefore, the single-step Euler update can be decomposed along the eigenvector basis as:

$$\mathbf{x}^{\text{Euler}}_{\sigma_{i+1}} - \mathbf{x}_{\sigma_i} = -h\boldsymbol{f}_{\mathbf{x}_{\sigma_i}} = -h\sum\nolimits_k \left\langle \boldsymbol{f}_{\mathbf{x}_{\sigma_i}},\ \mathbf{v}_k \right\rangle \mathbf{v}_k, \tag{9}$$

where $\langle \cdot, \cdot \rangle$ refers inner product. Similarly, Heun update step is given by

$$\mathbf{x}^{\text{Heun}}_{\sigma_{i+1}} - \mathbf{x}_{\sigma_i} = -\frac{h}{2}\big(\boldsymbol{f}_{\mathbf{x}_{\sigma_i}} + \boldsymbol{f}_{\mathbf{x}^{\text{Euler}}_{\sigma_{i+1}}}\big) \tag{10}$$

$$\approx -\frac{h}{2}\big(\boldsymbol{f}_{\mathbf{x}_{\sigma_i}} + \boldsymbol{f}_{\mathbf{x}_{\sigma_i}} + J_{\mathbf{x}_{\sigma_i}}(\mathbf{x}^{\text{Euler}}_{\sigma_{i+1}} - \mathbf{x}_{\sigma_i})\big) \tag{11}$$

$$= -h \sum_k \Big(1 + \tfrac{1}{2} z_k\Big) \left\langle \boldsymbol{f}_{\mathbf{x}_{\sigma_i}},\, \mathbf{v}_k \right\rangle \mathbf{v}_k, \tag{12}$$

where $z_k := -h\lambda_k$. The exact single-step update can be approximated as follows (See Appendix A.1 for derivation):

$$\mathbf{x}_{\sigma_{i+1}} - \mathbf{x}_{\sigma_i} \approx -h \sum_k \frac{\mathrm{e}^{z_k} - 1}{z_k} \left\langle \boldsymbol{f}_{\mathbf{x}_{\sigma_i}},\, \mathbf{v}_k \right\rangle \mathbf{v}_k. \tag{13}$$

Thus, the local truncation error of Heun's method and the ERK solution difference can be written as

$$\text{LTE}^{\text{Heun}} := \mathbf{x}_{\sigma_{i+1}} - \mathbf{x}^{\text{Heun}}_{\sigma_{i+1}} \approx -h \sum_k \alpha(z_k) \big\langle \boldsymbol{f}_{\mathbf{x}_{\sigma_i}},\, \mathbf{v}_k \big\rangle \mathbf{v}_k \tag{14}$$

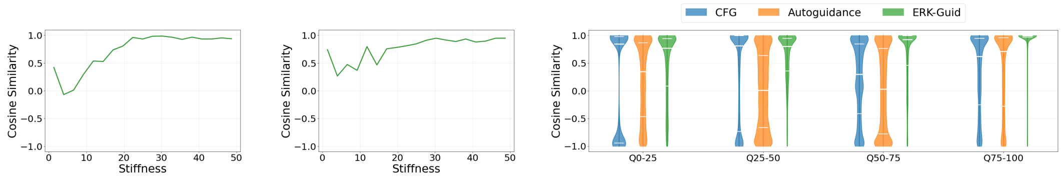
$$\Delta := \mathbf{x}^{\text{Heun}}_{\sigma_{i+1}} - \mathbf{x}^{\text{Euler}}_{\sigma_{i+1}} \approx -h \sum_k (\tfrac{1}{2} z_k) \big\langle \boldsymbol{f}_{\mathbf{x}_{\sigma_i}},\, \mathbf{v}_k \big\rangle \mathbf{v}_k \tag{15}$$

where $\alpha(z) := \frac{\mathrm{e}^z - 1}{z} - 1 - \frac{1}{2}z$. Figure 7 visualizes the behavior of $\alpha(z)$.

As $|z_k| = |h\lambda_k|$ increases, the weights $\frac{1}{2}z_k$ and $\alpha(z_k)$ associated with each eigenvector component also grow in magnitude, so that contributions from directions with large $|\lambda_k|$ come to dominate. Consequently, both the local truncation error and the ERK solution difference tend to align with the dominant eigenvector corresponding to the largest eigenvalue magnitude, specifically in stiff regions, i.e., regions with high stiffness. Motivated by this observation, we estimate both stiffness and the dominant eigenvector from the ERK solution difference during sampling (Section 4.2), and when stiffness is high, we apply guidance with Eq. 14 (Section 4.3).

Toy 2D Experiments. To better understand the connection between ODE dynamics and stiffness, we construct a two-dimensional toy system with an analytically defined ground-truth drift, followed by the Autoguidance Karras et al. (2024a). Please refer to Appendix B for experimental details, including the computation of eigenvectors, eigenvalues, and local truncation errors. The objective of this experiment is to reveal how stiff regions induce different behaviors along the dominant and subdominant eigenvector directions. Figure 1 (a) visualizes the degree of stiffness across the image, while panels (b)–(e) show the drift field projected onto eigenvector axes in two locally stiff regions. Along the *subdominant* eigenvector axes (panels (b) and (d)), the projected drift vectors (green arrows) remain nearly parallel with small variation in magnitude, indicating locally stable dynamics that can be well approximated by numerical solvers. In contrast, along the *dominant* eigenvector axes (panels (c) and (e)), the projected drift exhibits pronounced variations in both orientation and magnitude, making the dynamics harder to approximate and resulting in larger local truncation errors (LTE). Furthermore, in such stiff dynamics, higher-order solvers yield more significant improvements along the dominant eigenvector compared to the subdominant one, which naturally manifests as a larger ERK solution difference. In Table 1, we report the local truncation error (LTE) and the ERK solution difference measured in the upper region of Figure 1 (lines AB–CD), along the eigenvector axes. The LTE along the dominant eigenvector direction (line CD, 7.22e-05) is approximately 12.5× larger than that along the subdominant direction (line AB, 5.78e-06). Similarly, the ERK solution difference along line CD (4.03-04) is about 6.5× larger than that along AB (6.18e-05). These results confirm that both LTE and ERK solution difference are strongly amplified along the dominant eigenvector in stiff regions.

Inspired by these observations in Figure 1 and Table 1, where the LTE and the ERK solution difference exhibit substantially larger amplitudes along the dominant eigenvector axes, we further analyze their alignment across varying stiffness levels. Specifically, we compute the cosine similarity between the dominant eigenvector and both the LTE and ERK solution difference. As shown in Figure 2 (a), the alignment between LTE and dominant eigenvector steadily increases with stiffness, indicating that the eigenvector reliably serves as a proxy for the LTE direction in stiff regions. Additionally, Figure 2 (b) shows that the ERK solution difference is also strongly aligned with



(a) LTE (b) ERK solution difference (c) ERK drift difference across stiffness quantiles

Figure 2: **Eigenvector alignment across stiffness.** **(a)** Cosine similarity between the dominant eigenvector and the local truncation error (LTE) increases with stiffness. **(b)** The ERK solution difference exhibits a similar trend to LTE, suggesting it can serve as a reliable proxy for the LTE direction in high stiffness regions. **(c)** Our ERK-Guid consistently achieves higher cosine similarity with the eigenvector, highlighting its strong alignment in stiff regions. CFG and Autoguidance exhibit weaker or mixed alignment with the dominant eigenvector in stiff regions, supporting the complementarity of our method.

the dominant eigenvector in stiff regions. This motivates our novel guidance strategy, ERK-Guid, which leverages the dominant eigenvector estimated from the embedded Runge–Kutta (ERK) pair as its guiding signal.

We compare ERK-Guid against two widely used baselines: CFG Ho & Salimans (2022)(conditional–unconditional score difference) and Autoguidance Karras et al. (2024a)(main–weak model difference). Figure 2 (c) shows that ERK-Guid consistently exhibits strong alignment with the dominant eigenvector across stiffness quantiles. In particular, the gap widens in the high-stiffness bin (Q75–100), where CFG and Autoguidance exhibit weak or inconsistent alignment. These results suggest that ERK-Guid provides an orthogonal guidance signal that complements rather than overlaps with CFG and Autoguidance.

4.2 Stiffness and Dominant Eigenvector Estimator

The key intuition from the previous section is that, once stiffness is high, the dominant eigenvector provides a reliable proxy for reducing LTE. In practice, however, direct access to the Jacobian is infeasible, making stiffness estimation challenging. A common alternative is to use Jacobian–vector product (JVP) based power iterations, which are supported in frameworks such as PyTorch but remain prohibitively expensive for diffusion sampling, where each step already requires costly network evaluations. To overcome this, we propose cost-free estimators of stiffness and the dominant eigenvector, exploiting the ERK drift/solution difference without any additional evaluations.

Let $\mathbf{x}_{\sigma_i}$ denote the current state at σ_i (for $i > 0$), and $\mathbf{x}_{\sigma_i}^{\text{Euler}}$ the intermediate Euler prediction when advancing from σ_{i-1} to σ_i. We define the stiffness estimator as

$$\hat{\rho}_{\text{stiff}}(\mathbf{x}_{\sigma_i}, \sigma_i) := \frac{\left\| \boldsymbol{f}(\mathbf{x}_{\sigma_i}; \sigma_i) - \boldsymbol{f}\left(\mathbf{x}_{\sigma_i}^{\text{Euler}}; \sigma_i\right) \right\|_2}{\left\| \mathbf{x}_{\sigma_i} - \mathbf{x}_{\sigma_i}^{\text{Euler}} \right\|_2}. \tag{16}$$

Here, $\mathbf{x}_{\sigma_i} - \mathbf{x}_{\sigma_i}^{\text{Euler}}$ corresponds exactly to the ERK solution difference and $\boldsymbol{f}(\mathbf{x}_{\sigma_i}; \sigma_i) - \boldsymbol{f}\left(\mathbf{x}_{\sigma_i}^{\text{Euler}}; \sigma_i\right)$ corresponds to the ERK drift difference under the Heun sampler.

Proposition 1 *Let J be the Jacobian matrix of the drift function $\boldsymbol{f}(\mathbf{x}_{\sigma_i}; \sigma_i)$ at $\mathbf{x}_{\sigma_i}$. Assume that the ERK solution difference $\mathbf{x}_{\sigma_i} - \mathbf{x}_{\sigma_i}^{Euler}$ is sufficiently small and aligned with the eigenvector associated with the dominant eigenvalue λ of J in magnitude, in the sense that*

$$\left\| J(\mathbf{x}_{\sigma_i} - \mathbf{x}_{\sigma_i}^{Euler}) \right\| = |\lambda| \left\| \mathbf{x}_{\sigma_i} - \mathbf{x}_{\sigma_i}^{Euler} \right\| + \mathcal{O}(\left\| \mathbf{x}_{\sigma_i} - \mathbf{x}_{\sigma_i}^{Euler} \right\|^2).$$

Then the dominant eigenvalue λ admits the approximation

$$|\lambda| = \frac{\left\| \boldsymbol{f}(\mathbf{x}_{\sigma_i}; \sigma_i) - \boldsymbol{f}(\mathbf{x}_{\sigma_i}^{\text{Euler}}; \sigma_i) \right\|_2}{\left\| \mathbf{x}_{\sigma_i} - \mathbf{x}_{\sigma_i}^{\text{Euler}} \right\|_2} + \mathcal{O}(\left\| \mathbf{x}_{\sigma_i} - \mathbf{x}_{\sigma_i}^{Euler} \right\|).$$

Proposition 1 establishes that the proposed stiffness estimator accurately recovers the true stiffness under the assumption. We provide the proof of Proposition 1 in Appendix A.2. Since our estimator relies on the alignment between the ERK solution difference and the dominant eigenvector, it is reliable only when the estimated stiffness is sufficiently high. This requirement is explicitly incorporated into our guidance design (see Section 4.3). Notably, the stiffness estimator requires no additional network evaluations: $\mathbf{x}_{\sigma_i}$, $\mathbf{x}_{\sigma_i}^{\text{Euler}}$, and $\boldsymbol{f}(\mathbf{x}_{\sigma_i}^{\text{Euler}};\sigma_i)$ are already obtained during the Heun correction, while $\boldsymbol{f}(\mathbf{x}_{\sigma_i};\sigma_i)$ is reused in the subsequent Euler update step.

As shown in Section 4.1, the ERK solution difference provides a useful proxy for the dominant eigenvector in stiff regions. To improve robustness, we propose the *ERK drift difference*, which tracks the dominant eigenvector more accurately:

$$\hat{\mathbf{v}}_{\text{stiff}}(\mathbf{x}_{\sigma_i},\sigma_i) := \frac{\boldsymbol{f}(\mathbf{x}_{\sigma_i};\,\sigma_i) - \boldsymbol{f}\left(\mathbf{x}_{\sigma_i}^{\text{Euler}};\,\sigma_i\right)}{\left\|\boldsymbol{f}(\mathbf{x}_{\sigma_i};\,\sigma_i) - \boldsymbol{f}\left(\mathbf{x}_{\sigma_i}^{\text{Euler}};\,\sigma_i\right)\right\|_2}. \tag{17}$$

Under a local linearization, this difference approximates the Jacobian applied to the ERK solution difference:

$$\boldsymbol{f}(\mathbf{x}_{\sigma_i};\,\sigma_i) - \boldsymbol{f}\left(\mathbf{x}_{\sigma_i}^{\text{Euler}};\,\sigma_i\right) \;\approx\; J(\mathbf{x}_{\sigma_i};\,\sigma_i)\,\left(\mathbf{x}_{\sigma_i} - \mathbf{x}_{\sigma_i}^{\text{Euler}}\right),$$

which corresponds to a single-step JVP power iteration. In the eigenbasis of J, components associated with larger eigenvalues are amplified, whereas those with smaller eigenvalues are suppressed, thereby steering the estimated direction toward the dominant eigenvector. We demonstrate this effect in Section 5.1.

4.3 Embedded Runge–Kutta Guidance

Directly applying the cost-free estimators from Section 4.2 into Eq. 14 is unreliable: (i) LTE–eigenvector alignment weakens at low stiffness, (ii) the stiffness proxy suffers from a scale mismatch, and (iii) inaccurate eigenvalue estimates over-amplify the update. To address these issues, we introduce stabilizers that turn Eq. 14 into a practical and robust guidance scheme.

With the proposed cost-free estimators, let

$$\mathbf{f}_{\mathbf{x}_{\sigma_i}} := \boldsymbol{f}_\theta(\mathbf{x}_{\sigma_i};\sigma_i),\; \hat{\rho}_{\mathbf{x}_{\sigma_i}} := \hat{\rho}_{\text{stiff}}(\mathbf{x}_{\sigma_i},\sigma_i),\; \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}} := \hat{\mathbf{v}}_{\text{stiff}}(\mathbf{x}_{\sigma_i},\sigma_i). \tag{18}$$

We then define the ERK-Guid update as:

$$\beta := 1_{\{\hat{\rho}_{\mathbf{x}_{\sigma_i}} > w_{\text{con}}\}}, \tag{19}$$

$$z := w_{\text{stiff}}\, h\, \hat{\rho}_{\mathbf{x}_{\sigma_i}}, \tag{20}$$

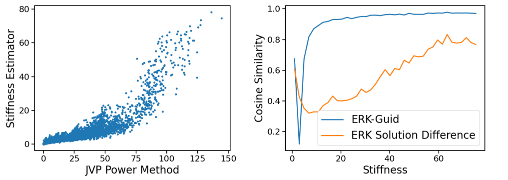
$$\hat{\mathbf{x}}_{\sigma_{i+1}}^{\text{Heun}} = \mathbf{x}_{\sigma_{i+1}}^{\text{Heun}} - h\,\beta\, z^2 \left\langle \mathbf{f}_{\mathbf{x}_{\sigma_i}},\, \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}} \right\rangle \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}}, \tag{21}$$

where $h := \sigma_i - \sigma_{i+1} > 0$. Here $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product, and w_{stiff}, w_{con} are hyperparameters. This construction is motivated by Eq. 14, which characterizes the exact local truncation error under the linearization assumption. A complete derivation is provided in Appendix A.3.

Compared to the exact LTE expression, we introduce three practical modifications: (i) a confidence gate β with hyperparameter w_{con}, which suppresses the update in low-stiffness regions and activates it only when $\hat{\rho} > w_{\text{con}}$, ensuring that guidance is applied where eigenvector alignment is reliable; (ii) a scale parameter w_{stiff} to correct for the consistent mismatch in absolute scale observed in our stiffness proxy and controls the overall guidance strength (no guidance when $w_{\text{stiff}} = 0$); and (iii) a quadratic form z^2 in place of $\alpha(z)$, which avoids exponential growth under inaccurate estimates while behaves similarly near zero (See Appendix B.3) and stabilizes the update by depending on eigenvalue magnitude rather than sign. The effects of β and z^2 are analyzed through ablations in Section 5.3.

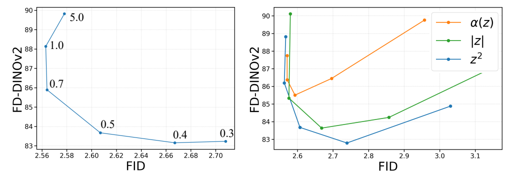
The first sampling step does not admit this construction because it requires an ERK pair from the previous iteration. We simply skip guidance at this step, noting that stiffness at initialization is typically very small in practice, so that $\beta \approx 0$. The complete procedure is summarized in Algorithm 1.

Computation cost. ERK-Guid incurs no additional network evaluations: all required quantities are already computed during the Heun update. Thus, unlike CFG or Autoguidance, our approach imposes no extra evaluation overhead and relies solely on the discrepancy between two solver orders.



(a) Accuracy of the Stiffness Estimator (b) Accuracy of the Eigenvector Estimator

Figure 3: **Accuracy of proposed estimators.** **(a)** Our estimated stiffness values highly correlate with JVP-based one. **(b)** ERK drift difference (blue) maintains higher alignment with the dominant eigenvector than the ERK solution difference (orange), especially at high stiffness.

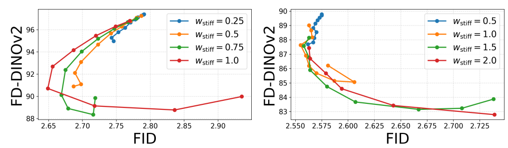


(a) Ablation on threshold w_{con} (b) Ablation on function of z

Figure 4: **Ablations on our ERK-Guid design.** **(a)** FD-DINOv2–FID trade-off curve according to confidence threshold w_{con}. **(b)** Comparison of scaling functions for FD-DINOv2–FID trade-off curve.

Table 2: **Quantitative results on ImageNet-512.**

#step	w_{stiff}	FD-DINOv2 ↓	FID ↓	Precision ↑	Recall ↑	IS ↑
32	0.0	90.1	2.58	0.630	0.673	244
32	0.5	88.8	2.57	0.632	0.673	245
32	1.0	86.2	**2.56**	**0.635**	0.674	247
32	1.5	83.7	2.58	**0.635**	0.674	**249**
32	2.0	**82.8**	2.74	0.633	**0.675**	247
32	2.5	84.9	3.03	0.625	0.668	241
16	0.0	97.5	2.79	0.628	0.653	238
16	0.5	**90.5**	**2.66**	**0.644**	**0.657**	**242**
8	0.0	161.2	7.06	0.445	**0.615**	183
8	0.5	**148.3**	**5.31**	**0.553**	0.590	**191**



(a) 16 Steps (b) 32 Steps

Figure 5: **Grid search of hyperparameters.** Quantitative trends of varying w_{con} at fixed w_{stiff} for (a) 16-step and (b) 32-step sampling.

5 EXPERIMENTS

We evaluate ERK-Guid across both toy and real-world datasets. On synthetic data, we validate our stiffness estimator against Jacobian Vector Product (JVP) references and compare ERK solution and drift differences for eigenvector estimation in Section 5.1. In Section 5.2, we present quantitative results on real-world datasets, comparing our method against unguided sampling. We then provide ablation studies on two stabilizers—the confidence gate β and the scaling function of z—to assess their impact in Section 5.3. In Section 5.4, we examine ERK-Guid's compatibility with existing guidance methods and its plug-and-play adaptability to solver methods to demonstrate its versatility. Please refer to the Appendix E for qualitative results.

Experimental setup. We conduct experiments on ImageNet (ILSVRC2012) Deng et al. (2009) at resolutions 512×512 and 64×64, as well as on the FFHQ Karras et al. (2019) at 64×64. We use the pre-trained EDM Karras et al. (2022) and EDM2 Karras et al. (2024a) models. Heun's method is used as the base solver, and other solvers are incorporated through our plug-and-play module.

Evaluation metrics follow prior work: *fidelity* via FD-DINOv2 Stein et al. (2023), FID Heusel et al. (2017), and Precision Kynkäänniemi et al. (2019); *diversity* via Recall Kynkäänniemi et al. (2019); and *condition alignment* using Inception Score (IS) Salimans et al. (2016). For additional implementation details, hyperparameters, and the reference estimator in Figure 3(a) are provided in Appendix B.

5.1 ACCURACY OF THE ESTIMATORS

In Figure 3, we validate the accuracy of our estimators. Figure 3(a) illustrates that the stiffness estimator shows strong correlation with the JVP reference, increasing consistently with the reference values. Also, Figure 3(b) demonstrates that the eigenvector estimator based on ERK drift differences exhibits higher alignment with the dominant eigenvector than the ERK solution difference, particularly in stiff regions. These results confirm the reliability of our estimators for identifying the dominant eigenvector direction as guidance.

Table 3: **Quantitative results of adaptation to guidance methods.**

#step	Method	FD-DINOv2 ↓	FID ↓	Precision ↑	Recall ↑	IS ↑
32	CFG	88.5	2.27	0.608	**0.708**	271
	+ERK-Guid	**83.8**	**2.27**	**0.610**	0.706	**275**
32	Autoguidance	50.5	1.36	0.698	**0.642**	262
	+ERK-Guid	47.7	**1.35**	0.692	0.630	267
	+ERK-Proj	**44.9**	1.36	**0.710**	0.605	**274**
16	CFG	133.89	3.60	0.593	**0.673**	210
	+ERK-Guid	**125.57**	**3.20**	**0.605**	**0.673**	**215**
16	Autoguidance	82.13	2.31	0.652	0.640	230
	+ERK-Guid	**75.16**	**1.92**	**0.669**	**0.643**	**236**

Table 4: **Quantitative results for plug-and-play solver adaptation on ImageNet-64 and FFHQ-64.**

Dataset	ImageNet 64×64			FFHQ 64×64		
NFEs	6	8	10	6	8	10
Heun	89.63	37.65	16.46	142.4	57.21	29.54
+ ERK-Guid	**85.19**	**35.92**	**13.85**	**132.8**	**54.73**	**23.38**
DPM-Solver	44.83	12.42	6.84	83.17	22.84	9.46
+ ERK-Guid	**31.59**	**10.58**	**6.54**	**49.0**	**10.44**	**4.64**
DEIS	12.57	6.84	5.34	12.25	7.59	5.56
+ ERK-Guid	**9.56**	**6.25**	**4.89**	**9.96**	**6.04**	**4.47**

5.2 Effectiveness of the Guidance

Table 2 summarizes quantitative results on ImageNet 512×512 with EDM2 and the Heun sampler. We take $w_{\text{stiff}} = 0$ as the baseline without guidance. As the guidance scale increases, FD-DINOv2 consistently decreases and reaches its best at $w_{\text{stiff}} = 2.0$, yielding 82.8 compared to the baseline 90.1. Importantly, this fidelity gain is achieved while keeping FID competitive and consistently improving Precision, Recall, and Inception Score, indicating that our update strengthens fidelity without sacrificing diversity or alignment. The advantage becomes more pronounced under fewer sampling steps, where truncation errors dominate. With 16 steps, FD-DINOv2 improves from 97.5 to 90.5 and FID from 2.79 to 2.66, accompanied by gains in Precision and Inception Score. At 8 steps, the effect is even stronger: FD-DINOv2 drops from 161.2 to 148.3, FID from 7.06 to 5.31, with substantial boosts in Precision, Recall, and Inception Score. Overall, these results demonstrate that ERK-Guid effectively mitigates error accumulation in stiff regions, delivering consistent improvements across settings and providing particular advantages in low-step regimes, all without any additional training or model evaluations.

5.3 Ablation on Guidance Design

We conduct an ablation study on two design choices from Eq. 21: the confidence gate β (controlled by threshold w_{con}) and the scaling form applied to z. When the estimated stiffness is below w_{con}, the confidence gate suppresses our guidance update, while the scaling function regulates the update magnitude for stability.

Confidence gate β. Figure 4(a) investigates the effect of varying w_{con}. A higher threshold activates guidance only in very stiff regions, leading to infrequent corrections and limited improvements in FD-DINOv2. As the threshold decreases, guidance is applied more often and FD-DINOv2 steadily improves, but excessively small thresholds eventually degrade FID. These results indicate that a moderate threshold provides the best trade-off between fidelity gains and stable FID. Moreover, Figure 5 provides a grid-search analysis over w_{stiff} and threshold w_{con}, illustrating robust trends across hyperparameter choices and confirming that ERK-Guid maintains stable improvements over a wide range of settings.

Scaling function for z. Figure 4(b) compares the exponential-like scaling $\alpha(z)$ with two alternatives, $|z|$ and z^2. Among them, z^2 consistently attains lower FD-DINOv2 at comparable or better FID, achieving the best trade-off by avoiding excessive growth under estimation errors. In contrast, $\alpha(z)$ introduces instability and degrades FD-DINOv2, while $|z|$ ensures stability but yields limited gains. These results demonstrate that the quadratic form z^2 provides a robust scaling that preserves stability while improving fidelity.

5.4 Guidance Compatibility and Solver plug-and-play solver adaptation

In this section, we highlight two key properties of ERK-Guid: (i) its compatibility with existing model-based guidance methods, and (ii) its plug-and-play adaptability to various solvers.

Guidance compatibility. We examine whether ERK-Guid can be combined with existing guidance schemes. Under the predictor–corrector view Bradley & Nakkiran (2024), guidance methods act as correctors. Diffusion sampling errors arise from two sources: solver error (LTE) and model error. ERK-Guid targets LTE, whereas CFG and Autoguidance target model error. Motivated by this complementarity, we combine CFG and Autoguidance with ERK-Guid and ERK-Proj. ERK-Proj is a light extension that simply interpolates between model-error–based and LTE-based corrections, aiming to reduce both simultaneously. Additional details are provided in Appendix B.2. In Table 3, we show that our correction consistently strengthens model-based guidances, indicating strong extensibility to other guidance methods.

Figure 6: **Qualitative comparison on PixArt-α Chen et al. (2023)** We perform text-to-image generation to compare DPM-Solver with our ERK-Guid. As shown in the blue zoomed-in regions, ERK-Guid captures fine semantic details more accurately.

Plug-and-play adaptation. Similar to other guidance methods, ERK-Guid can be applied to various solvers as a plug-and-play correction module. To demonstrate its effectiveness and broad applicability, we evaluate ERK-Guid on higher-order solvers, including Heun's method Hairer et al. (1993), DPM-Solver Lu et al. (2022), and DEIS Zhang & Chen (2023) on ImageNet Deng et al. (2009) and FFHQ Karras et al. (2019) at 64×64 resolution. Table 4 demonstrates that combining ERK-Guid with solver methods consistently improves performance across all NFEs on both datasets. These results highlight the robust plug-and-play capability of ERK-Guid and its effectiveness even when paired with higher-order ODE solvers. Additional details are provided in Appendix B.4 and Table 7. Moreover, Figure 6 demonstrates that ERK-Guid delivers strong qualitative improvements on PixArt-α Chen et al. (2023), built upon a Diffusion Transformer (DiT) Peebles & Xie (2023) backbone, further highlighting its architectural generalization capability.

6 Conclusion

In this work, we propose ERK-Guid, a stiffness-aware diffusion sampling scheme based on Embedded Runge–Kutta guidance. Motivated by the observation that local truncation error aligns with the dominant eigenvector in stiff regions, we introduce cost-free estimators for both stiffness and eigenvectors using ERK solution and drift discrepancies. Building on these estimators, we design a stabilized guidance framework that balances theoretical insight with practical robustness. Experiments on synthetic data and ImageNet demonstrate that ERK-Guid delivers an orthogonal guidance signal without additional neural evaluations, establishing an efficient paradigm for improving diffusion model sampling.

REPRODUCIBILITY STATEMENT

In Section 4.3, we describe our pipeline design, and the Appendix C provides algorithmic details and full pseudocode. Appendix B documents experimental settings and hyperparameters. Together, we offer sufficient information for reproducibility of ERK-Guid.

ETHICS STATEMENT

Our method relies on pretrained generative models, which may produce harmful or biased outputs depending on the conditioning input. This risk is inherent to the underlying pretrained models, and we emphasize the need for responsibility.

REFERENCES

Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector. In *NeurIPS Workshop*, 2024.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations, ICLR*, 2019.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2023.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations, ICLR*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2009.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Communications of the ACM*, 2020.

Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *European Conference on Computer Vision, ECCV*, 2024.

Ernst Hairer and Gerhard Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, volume 14 of *Springer Series in Computational Mathematics*. Springer, 2nd revised edition, 1996. ISBN 978-3-540-60452-5.

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer, 2nd edition, 1993. ISBN 978-3-540-56670-0.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems, NeurIPS*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *Advances in Neural Information Processing Systems, NeurIPS Workshop*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems, NeurIPS*, 2022.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems, NeurIPS*, 2024a.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2024b.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2023.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations, ICLR*, 2014.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems, NeurIPS*, 2019.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems, NeurIPS*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations, ICLR*, 2023.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations, ICLR*, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems, NeurIPS*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision, ICCV*, pp. 4195–4205, 2023.

Linda Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM journal on scientific and statistical computing*, 4(1):136–148, 1983.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Learning Representations, ICLR*, 2015.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2022.

Seyedmorteza Sadat, Otmar Hilliges, and Romann M Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *International Conference on Learning Representations, ICLR*, 2024.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems, NeurIPS*, 2016.

Lawrence F Shampine and Charles William Gear. A user's view of solving stiff ordinary differential equations. *SIAM review*, 21(1):1–17, 1979.

Neta Shaul, Juan Perez, Ricky TQ Chen, Ali Thabet, Albert Pumarola, and Yaron Lipman. Bespoke solvers for generative flow models. In *International Conference on Learning Representations, ICLR*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations, ICLR*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations, ICLR*, 2021b.

George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

Shuai Wang, Zexian Li, Tianhui Song, Xubin Li, Tiezheng Ge, Bo Zheng, Limin Wang, et al. Differentiable solver search for fast diffusion sampling. In *International Conference on Machine Learning, ICML*, 2025.

Shuchen Xue, Zhaoqiang Liu, Fei Chen, Shifeng Zhang, Tianyang Hu, Enze Xie, and Zhenguo Li. Accelerating diffusion sampling with optimized time steps. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2023.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *International Conference on Learning Representations, ICLR*, 2023.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer. In *International Conference on Learning Representations, ICLR*, 2025.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

Candi Zheng and Yuan Lan. Characteristic guidance: Non-linear correction for diffusion model at large guidance scale. In *International Conference on Machine Learning, ICML*, 2024.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. In *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

A Derivation

A.1 Derivation of the Exact One Step Increment

We provide a detailed derivation of Eq. 13, i.e., the exact one step increment under the local linearization. Recall that around $\mathbf{x}_{\sigma_i}$, the score-based vector field is approximated as

$$\frac{\mathrm{d}\mathbf{x}_\sigma}{\mathrm{d}\sigma} \approx \mathbf{f}_{\mathbf{x}_{\sigma_i}} + J_{\mathbf{x}_{\sigma_i}}(\mathbf{x}_\sigma - \mathbf{x}_{\sigma_i}), \tag{22}$$

where we denote $J_{\mathbf{x}_{\sigma_i}} = J(\mathbf{x}_{\sigma_i}; \sigma_i)$ and $\mathbf{f}_{\mathbf{x}_{\sigma_i}} = \mathbf{f}(\mathbf{x}_{\sigma_i}; \sigma_i)$. This yields a linear ODE of the form

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\sigma} = J_{\mathbf{x}_{\sigma_i}}\,\mathbf{y} + \mathbf{f}_{\mathbf{x}_{\sigma_i}}, \quad \mathbf{y}(\sigma_i) = \mathbf{0}, \tag{23}$$

where we introduced the shifted variable $\mathbf{y}(\sigma) := \mathbf{x}_\sigma - \mathbf{x}_{\sigma_i}$. Let $h := \sigma_i - \sigma_{i+1} > 0$ denotes the step size. Our goal is to compute $\mathbf{y}(\sigma_i - h) = \mathbf{x}_{\sigma_{i+1}} - \mathbf{x}_{\sigma_i}$.

Since $J_{\mathbf{x}_{\sigma_i}}$ is the Hessian of $\log p(\mathbf{x}_{\sigma_i}; \sigma_i)$ under C^2 smoothness, it is symmetric and hence admits the eigendecomposition

$$J_{\mathbf{x}_{\sigma_i}} = V\Lambda V^\top, \quad J_{\mathbf{x}_{\sigma_i}}\mathbf{v}_k = \lambda_k \mathbf{v}_k, \quad \|\mathbf{v}_k\|_2 = 1, \tag{24}$$

where $V = [\mathbf{v}_1, \ldots, \mathbf{v}_d]$ is orthogonal and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$.

Let define the state which projected to the eigenbasis space:

$$\boldsymbol{u}(\sigma) \;:=\; V^\top \boldsymbol{y}(\sigma), \qquad \boldsymbol{g} \;:=\; V^\top \boldsymbol{f}_{\boldsymbol{x}_{\sigma_i}}, \qquad g_k = \boldsymbol{g}^\top \boldsymbol{e}_k = \langle \mathbf{f}_{\mathbf{x}_{\sigma_i}}, \mathbf{v}_k \rangle. \tag{25}$$

where $\langle \cdot, \cdot \rangle$ refers inner product. Since V is constant on this interval (by the local linearization assumption):

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\sigma} = \frac{\mathrm{d}}{\mathrm{d}\sigma}\big(V^\top \boldsymbol{y}(\sigma)\big) &= V^\top \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\sigma} && (26)\\
&= V^\top \big(J_{\boldsymbol{x}_{\sigma_i}}\boldsymbol{y} + \boldsymbol{f}_{\boldsymbol{x}_{\sigma_i}}\big) && (27)\\
&= V^\top (V\Lambda V^\top)\,\boldsymbol{y} \;+\; V^\top \boldsymbol{f}_{\boldsymbol{x}_{\sigma_i}} && (28)\\
&= (V^\top V)\,\Lambda\,(V^\top \boldsymbol{y}) \;+\; \boldsymbol{g} && (29)\\
&= \Lambda \boldsymbol{u} + \boldsymbol{g}. && (30)
\end{aligned}$$

Therefore the system *decouples* into d independent scalar ODEs because Λ is diagonal.

Writing the k-th coordinate explicitly,

$$\frac{\mathrm{d}u_k}{\mathrm{d}\sigma} = \lambda_k u_k + g_k, \qquad u_k(\sigma_i) = 0. \tag{31}$$

It has the closed-form solution:

$$u_k(\sigma) = \frac{\mathrm{e}^{\lambda_k(\sigma - \sigma_i)} - 1}{\lambda_k}\, g_k, \quad \text{if } \lambda_k \neq 0, \tag{32}$$

and

$$u_k(\sigma) = (\sigma - \sigma_i)\, g_k, \quad \text{if } \lambda_k = 0. \tag{33}$$

Both cases can be compactly written as

$$u_k(\sigma) = \frac{\mathrm{e}^{\lambda_k(\sigma - \sigma_i)} - 1}{\lambda_k}\, g_k, \tag{34}$$

interpreting the fraction as its limit when $\lambda_k \to 0$.

Let $z_k := \lambda_k(\sigma_{i+1} - \sigma_i) = -h\lambda_k$. Since $\mathbf{y} = V\mathbf{u}$, we obtain

$$\mathbf{x}_{\sigma_{i+1}} - \mathbf{x}_{\sigma_i} = \mathbf{y}(\sigma_i - h) = V\mathbf{u}(\sigma_i - h) \tag{35}$$

$$= -hV \sum_{k=1}^{d} \frac{\mathrm{e}^{z_k} - 1}{z_k} g_k \mathbf{e}_k \tag{36}$$

$$= -h \sum_{k=1}^{d} \frac{\mathrm{e}^{z_k} - 1}{z_k} g_k V \mathbf{e}_k \tag{37}$$

$$= -h \sum_{k=1}^{d} \frac{\mathrm{e}^{z_k} - 1}{z_k} \langle \mathbf{f}_{\mathbf{x}_{\sigma_i}}, \mathbf{v}_k \rangle \mathbf{v}_k, \tag{38}$$

which is the desired expression in Eq. 13.

A.2 Derivation of Proposition 1

In Section 4.2 of the main paper, we introduce a cost-free stiffness estimator that exploit the ERK difference without additional evaluations. We provide the proof of Proposition 1, which shows that proposed estimator can approximate the magnitude of the dominant eigenvalue.

Proposition 1 *Let J be the Jacobian matrix of the drift function $\boldsymbol{f}(\mathbf{x}_{\sigma_i};\sigma_i)$ at $\mathbf{x}_{\sigma_i}$. Assume that the ERK solution difference $\mathbf{x}_{\sigma_i} - \mathbf{x}^{Euler}_{\sigma_i}$ is sufficiently small and aligned with the eigenvector associated with the dominant eigenvalue λ of J in magnitude, in the sense that*

$$\left\|J(\mathbf{x}_{\sigma_i} - \mathbf{x}^{Euler}_{\sigma_i})\right\| = |\lambda| \left\|\mathbf{x}_{\sigma_i} - \mathbf{x}^{Euler}_{\sigma_i}\right\| + \mathcal{O}(\left\|\mathbf{x}_{\sigma_i} - \mathbf{x}^{Euler}_{\sigma_i}\right\|^2).$$

Then the dominant eigenvalue λ admits the approximation

$$|\lambda| = \frac{\left\| \boldsymbol{f}(\mathbf{x}_{\sigma_i};\sigma_i) - \boldsymbol{f}\left(\mathbf{x}^{\mathrm{Euler}}_{\sigma_i};\sigma_i\right) \right\|_2}{\left\| \mathbf{x}_{\sigma_i} - \mathbf{x}^{\mathrm{Euler}}_{\sigma_i}\right\|_2} + \mathcal{O}(\left\|\mathbf{x}_{\sigma_i} - \mathbf{x}^{Euler}_{\sigma_i}\right\|).$$

Proof 1 *Let $\boldsymbol{\delta} := \mathbf{x}_{\sigma_i} - \mathbf{x}^{\mathrm{Euler}}_{\sigma_i} \neq \mathbf{0}$ and define the Jacobian*

$$J := \frac{\partial \boldsymbol{f}}{\partial \mathbf{x}}\Big(\mathbf{x}_{\sigma_i};\sigma_i\Big).$$

A first-order Taylor expansion of $\boldsymbol{f}(\mathbf{x};\sigma_i)$ about $\mathbf{x}_{\sigma_i}$ gives

$$\boldsymbol{f}\left(\mathbf{x}^{\mathrm{Euler}}_{\sigma_i};\sigma_i\right) = \boldsymbol{f}\left(\mathbf{x}_{\sigma_i};\sigma_i\right) - J\boldsymbol{\delta} + \mathcal{O}\big(\|\boldsymbol{\delta}\|^2\big), \tag{39}$$

where the remainder is understood in the chosen vector norm. Rearranging Eq. 39 and taking norms on both sides yields

$$\|J\boldsymbol{\delta}\| = \left\|\boldsymbol{f}(\mathbf{x}_{\sigma_i};\sigma_i) - \boldsymbol{f}(\mathbf{x}^{\mathrm{Euler}}_{\sigma_i};\sigma_i)\right\| + \mathcal{O}\big(\|\boldsymbol{\delta}\|^2\big). \tag{40}$$

By the alignment hypothesis,

$$\|J\boldsymbol{\delta}\| = |\lambda|\,\|\boldsymbol{\delta}\| + \mathcal{O}\big(\|\boldsymbol{\delta}\|^2\big), \tag{41}$$

where λ is the dominant eigenvalue of J. Substituting Eq. 41 into Eq. 40 and dividing by $\|\boldsymbol{\delta}\| \neq 0$ gives

$$|\lambda| = \frac{\left\|\boldsymbol{f}(\mathbf{x}_{\sigma_i};\sigma_i) - \boldsymbol{f}(\mathbf{x}^{\mathrm{Euler}}_{\sigma_i};\sigma_i)\right\|}{\left\|\mathbf{x}_{\sigma_i} - \mathbf{x}^{\mathrm{Euler}}_{\sigma_i}\right\|} + \mathcal{O}\big(\|\boldsymbol{\delta}\|\big), \tag{42}$$

establishing the claimed approximation of the dominant eigenvalue.

A.3 Derivation of the ERK-Guid from Eq. 14

We present a detailed derivation showing how the ERK-Guid (Eq. 21) can be obtained from the local truncation error of Heun's method (Eq. 14). Let assume v_1 refers ground truth dominant eigenvector in Eq. 8. As discussed in Section 4.1, when the dominant eigenvector v_1 governs local dynamics, the LTE of Heun's method is dominated along the direction of v_1:

$$\mathrm{LTE}^{\mathrm{Heun}} := \mathbf{x}_{\sigma_{i+1}} - \mathbf{x}^{\mathrm{Heun}}_{\sigma_{i+1}} \approx -h\,\alpha(z_1)\big\langle \boldsymbol{f}_{\mathbf{x}_{\sigma_i}},\, \mathbf{v}_1 \big\rangle\, \mathbf{v}_1. \tag{43}$$

Rearranging terms gives

$$\mathbf{x}_{\sigma_{i+1}} \approx \mathbf{x}^{\mathrm{Heun}}_{\sigma_{i+1}} - h\,\alpha(z_1)\big\langle \boldsymbol{f}_{\mathbf{x}_{\sigma_i}},\, \mathbf{v}_1 \big\rangle\, \mathbf{v}_1 \tag{44}$$

Because $\alpha(z) = \frac{e^z - 1}{z} - 1 - \frac{1}{2}z$, its Taylor expansion at $z = 0$ yields

$$\alpha(z_1) = \frac{1}{6} z_1^2 + O(z_1^3), \tag{45}$$

and substituting this Taylor approximation into Eq. 44 gives

$$\mathbf{x}_{\sigma_{i+1}} \approx \mathbf{x}^{\mathrm{Heun}}_{\sigma_{i+1}} - \frac{1}{6} h\, z_1^2 \left\langle \mathbf{f}_{\mathbf{x}_{\sigma_i}},\, \mathbf{v}_1 \right\rangle \mathbf{v}_1. \tag{46}$$

We interpret this additive term as a guidance correction and introduce a tunable scale w_{stiff} as follows

$$\hat{\mathbf{x}}^{\mathrm{Heun}}_{\sigma_{i+1}} = \mathbf{x}^{\mathrm{Heun}}_{\sigma_{i+1}} - h\, z^2 \left\langle \mathbf{f}_{\mathbf{x}_{\sigma_i}},\, \mathbf{v}_1 \right\rangle \mathbf{v}_1, \quad z := w_{\text{stiff}}\, z_1, \tag{47}$$

where constant $\frac{1}{6}$ is absorbed into w_{stiff}.

Since neither the dominant eigenvector v_1 nor the eigenvalue λ is available in practice, we replace them with our cost-free estimators $\hat{v}_1$ and stiffness $\hat{\rho} \approx |\lambda_1|$. This yields the practical ERK-Guid update:

$$\hat{\mathbf{x}}^{\mathrm{Heun}}_{\sigma_{i+1}} = \mathbf{x}^{\mathrm{Heun}}_{\sigma_{i+1}} - h\, z^2 \left\langle \mathbf{f}_{\mathbf{x}_{\sigma_i}},\, \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}} \right\rangle \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}}, \quad z \approx w_{\text{stiff}}\, h\hat{\rho}. \tag{48}$$

Since the derivation assumes operation within stiff regions, we introduce a step function β that activates the guidance only when the estimated stiffness exceeds a threshold, i.e., $\hat{\rho} > w_{\text{con}}$.

$$\hat{\mathbf{x}}^{\mathrm{Heun}}_{\sigma_{i+1}} = \mathbf{x}^{\mathrm{Heun}}_{\sigma_{i+1}} - h\,\beta\, z^2 \left\langle \mathbf{f}_{\mathbf{x}_{\sigma_i}},\, \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}} \right\rangle \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}}, \quad z \approx w_{\text{stiff}}\, h\hat{\rho}. \tag{49}$$

A.4 Resemblance Between our ERK-Guid and Other Guidances

We provide the alternative (but equivalent) formulation of our proposed method as a common form of guidance schemes. In Eq. 21, the ERK-Guid updates the Huen prediction as

$$\hat{\mathbf{x}}^{\mathrm{Heun}}_{\sigma_{i+1}} = \mathbf{x}^{\mathrm{Heun}}_{\sigma_{i+1}} - h\,\beta\, z^2 \left\langle \mathbf{f}_{\mathbf{x}_{\sigma_i}},\, \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}} \right\rangle \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}},$$

where $h = \sigma_i - \sigma_{i+1}$, $\beta := 1_{\{\hat{\rho}_{\mathbf{x}_{\sigma_i}} > w_{\text{con}}\}}$, and $z = w_{\text{stiff}}\, h\, \hat{\rho}_{\mathbf{x}_{\sigma_i}}$. We abbreviate the drift as $\mathbf{f}_{\mathbf{x}_{\sigma_i}}$ and eigenvector estimator $\hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}}$, where the latter is defined by Eq. 17 as following:

$$\hat{\mathbf{v}}_{\text{stiff}}(\mathbf{x}_{\sigma_i}, \sigma_i) := \frac{\boldsymbol{f}(\mathbf{x}_{\sigma_i}; \sigma_i) - \boldsymbol{f}\big(\mathbf{x}^{\mathrm{Euler}}_{\sigma_i}; \sigma_i\big)}{\big\|\boldsymbol{f}(\mathbf{x}_{\sigma_i}; \sigma_i) - \boldsymbol{f}\big(\mathbf{x}^{\mathrm{Euler}}_{\sigma_i}; \sigma_i\big)\big\|_2}.$$

Substituting this expression into ERK-Guid in Eq. 21 yields

$$\hat{\mathbf{x}}^{\mathrm{Heun}}_{\sigma_{i+1}} = \mathbf{x}^{\mathrm{Heun}}_{\sigma_{i+1}} - \frac{h\,\beta\, z^2 \left\langle \mathbf{f}_{\mathbf{x}_{\sigma_i}},\, \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}} \right\rangle}{\big\|\boldsymbol{f}(\mathbf{x}_{\sigma_i}; \sigma_i) - \boldsymbol{f}\big(\mathbf{x}^{\mathrm{Euler}}_{\sigma_i}; \sigma_i\big)\big\|_2} \Big(\boldsymbol{f}(\mathbf{x}_{\sigma_i}; \sigma_i) - \boldsymbol{f}\big(\mathbf{x}^{\mathrm{Euler}}_{\sigma_i}; \sigma_i\big)\Big). \tag{50}$$

For simplicity, we introduce an adaptive scaling function

$$\gamma(\mathbf{x}_{\sigma_i}, \mathbf{x}^{\mathrm{Euler}}_{\sigma_i}, \sigma_i) = \frac{-h\,\beta\, z^2 \left\langle \mathbf{f}_{\mathbf{x}_{\sigma_i}},\, \hat{\mathbf{v}}_{\mathbf{x}_{\sigma_i}} \right\rangle}{\big\|\boldsymbol{f}(\mathbf{x}_{\sigma_i}; \sigma_i) - \boldsymbol{f}\big(\mathbf{x}^{\mathrm{Euler}}_{\sigma_i}; \sigma_i\big)\big\|_2}. \tag{51}$$

The ERK-Guid update can then be rewritten as

$$\hat{\mathbf{x}}^{\mathrm{Heun}}_{\sigma_{i+1}} = \mathbf{x}^{\mathrm{Heun}}_{\sigma_{i+1}} + \gamma(\mathbf{x}_{\sigma_i}, \mathbf{x}^{\mathrm{Euler}}_{\sigma_i}, \sigma_i)\Big(\boldsymbol{f}(\mathbf{x}_{\sigma_i}; \sigma_i) - \boldsymbol{f}\big(\mathbf{x}^{\mathrm{Euler}}_{\sigma_i}; \sigma_i\big)\Big). \tag{52}$$

The above equation clearly shows that our method is a guidance with an adaptive guidance scaling γ.

B EXPERIMENTAL DETAILS

B.1 2D TOY EXPERIMENT

For the toy experiments 4.1, we explicitly construct the Jacobian by applying Jacobian–vector products (JVPs) to one-hot basis vectors, and compute its eigenvalues and eigenvectors directly. Since eigenvectors are sign-ambiguous, we orient them to point in the same half-space as the drift at each state, following Eq. 14. Sampling is performed without any additional guidance and always with the ground-truth score function. Because the exact ground-truth solution is not available, we approximate it by subdividing each original step into 30 smaller substeps with a much finer step size. The local truncation error (LTE) is defined relative to the Heun method under this reference. Figures 2(a,b) are obtained by partitioning into stiffness bins and plotting the median cosine similarity within each bin.

B.2 MAIN EXPERIMENTS

We estimate the dominant eigenvector and the stiffness using JVP-based power iteration with a random initialization and 300 iterations per timestep. Figure 8 visualizes the per-timestep convergence and indicates that 300 iterations are sufficient. As in the toy setup, we fix the eigenvector orientation via Eq. 14 so that it points toward the local drift; Figure 3(b) reports the median cosine similarity under this convention. For Table 2, we use the confidence threshold w_{con}=0.5. Figure 4(a) uses w_{stiff}=1.5, and Figure 4(b) uses w_{con}=0.5. In Figure 6, we conduct 15 sampling steps for text-to-image generation on PixArt-α Chen et al. (2023) which adopts Diffusion Transformer (DiT) Peebles & Xie (2023) architecture.

Guidance Integration with CFG and Autoguidance. In Section 5.4, we introduce guidance compatibility of ERK-Guid and ERK-Proj. We define the standard model-based guidance term, which applies to methods such as CFG and Autoguidance as follows

$$\boldsymbol{g} := \boldsymbol{f}_{\text{main}}(\mathbf{x}_{\sigma_i};\, \sigma_i) - \boldsymbol{f}_{\text{guiding}}(\mathbf{x}_{\sigma_i};\, \sigma_i)\,. \tag{53}$$

Model-based Guidance with ERK-Guid. Let the combined guidance method be expressed by Eq. 53. To incorporate ERK-Guid, we simply replace the original drift $\boldsymbol{f}$ with the guided drift as follows

$$\boldsymbol{f}^{w}(\mathbf{x}_{\sigma_i};\, \sigma_i) := \boldsymbol{f}_{\text{main}}(\mathbf{x}_{\sigma_i};\, \sigma_i) + (w-1)\boldsymbol{g}, \tag{54}$$

where w is scaling hyperparameters. Then, we compute the eigenvector estimator and stiffness estimator as following:

$$\hat{\mathbf{v}}^{w}_{\text{stiff}}(\mathbf{x}_{\sigma_i}, \sigma_i) := \frac{\boldsymbol{f}^{w}(\mathbf{x}_{\sigma_i};\, \sigma_i) - \boldsymbol{f}^{w}\left(\mathbf{x}^{\text{Euler}}_{\sigma_i};\, \sigma_i\right)}{\left\|\boldsymbol{f}^{w}(\mathbf{x}_{\sigma_i};\, \sigma_i) - \boldsymbol{f}^{w}\left(\mathbf{x}^{\text{Euler}}_{\sigma_i};\, \sigma_i\right)\right\|_2}, \tag{55}$$

$$\hat{\rho}^{w}_{\text{stiff}}(\mathbf{x}_{\sigma_i},\, \sigma_i) := \frac{\left\|\,\boldsymbol{f}^{w}(\mathbf{x}_{\sigma_i};\, \sigma_i) - \boldsymbol{f}^{w}\left(\mathbf{x}^{\text{Euler}}_{\sigma_i};\, \sigma_i\right)\right\|_2}{\left\|\,\mathbf{x}_{\sigma_i} - \mathbf{x}^{\text{Euler}}_{\sigma_i}\right\|_2}. \tag{56}$$

Finally, we constitute our ERK-Guid as follows

$$\hat{\mathbf{x}}^{\text{Heun}}_{\sigma_{i+1}} = \mathbf{x}^{\text{Heun}}_{\sigma_{i+1}} - h\,\beta\,z^2\left\langle \mathbf{f}^{w}_{\mathbf{x}_{\sigma_i}},\, \hat{\mathbf{v}}^{w}_{\mathbf{x}_{\sigma_i}}\right\rangle \hat{\mathbf{v}}^{w}_{\mathbf{x}_{\sigma_i}}. \tag{57}$$

Model-based Guidance with ERK-Proj. ERK-Proj interpolates between model-error–based and LTE-based corrections, aiming to reduce both simultaneously. ERK-Proj is a light extension that simply interpolates between these two correction signals, aiming to reduce both errors simultaneously. ERK-Proj is defined as follows

$$\eta := e^{-w_{\text{stiff}}\hat{\rho}_{\text{stiff}}}, \tag{58}$$

$$\hat{\mathbf{g}} := \eta\mathbf{g} + \left(1-\eta\right)\langle \mathbf{g}, \hat{\mathbf{v}}_{\text{stiff}}\rangle\, \hat{\mathbf{v}}_{\text{stiff}}, \tag{59}$$

$$\boldsymbol{f}^{w}(\mathbf{x}_{\sigma_i};\, \sigma_i) := \boldsymbol{f}_{\text{main}}(\mathbf{x}_{\sigma_i};\, \sigma_i) + (w-1)\hat{\mathbf{g}}, \tag{60}$$

where η adjusts the guidance scaling by stiffness estimator to interpolate the two guidance signals.

Table 5: **Wall-clock time (seconds per image) on a single RTX 3090 GPU.**

Method	**Avg**	**Min**	**Max**
Heun	2.777	2.775	2.782
ERK-Guid (Ours)	2.794	2.785	2.811

Table 6: **Memory consumption for generating a single image.**

Method	**Avg** (MB)
Heun	1906.82
ERK-Guid (Ours)	1906.82

Computational cost. All main experiments were run on 8× NVIDIA RTX 3090 GPUs. In Table 5 and Table 6, we evaluate the wall-clock time and memory overhead on the ImageNet 512×512 dataset using a single RTX 3090 GPU with batch size 1, comparing ERK-Guid against Heun's method. ERK-Guid incurs only a slight increase in wall-clock time, and its memory consumption remains identical to that of Heun's method.

B.3 Choice of ablation functions

We were concerned about excessive amplification from the original factor $\alpha(z)$, whose exponential growth makes it sensitive to estimation errors. Therefore, we replaced it with the lowest-order term of its Taylor expansion around $z{=}0$, i.e., a *quadratic* scaling z^2 (See Figure 7). As an ablation, we also tested an even lower-growth alternative, $|z|$, which is symmetric and linear in the magnitude of z. This pair (z^2 vs. $|z|$) lets us check whether stability comes simply from curbing amplification (linear) or from staying close to the local behavior of $\alpha(z)$ (quadratic).

Table 7: **Guidance configuration for Heun, DPM-Solver, and DEIS.**

Solver	Pair of states	h	β
Heun	$\mathbf{x}_{\sigma_i}$, $\mathbf{x}^{\text{Euler}}_{\sigma_i}$	$\sigma_i - \sigma_{i+1}$	$\{0, 1\}$
DPM-Solver (2S)	$\mathbf{x}_{\sigma_i+\delta}$, $\mathbf{x}_{\sigma_i}$	$\sigma_i - \sigma_{i+1}$	$\{0, 1\}$
DEIS	$\mathbf{x}_{\sigma_i}$, $\mathbf{x}_{\sigma_{i-1}}$	$\sigma_{i-1} - \sigma_i$	$\{0, -1\}$

B.4 Plug-and-Play Module for Advanced Solvers

In main paper Section 5.4, we present additional experimental results that confirm the effectiveness of our method combined with various solvers as a guidance/corrector.

$$\mathbf{x}^{\text{solver}}_{\sigma_{i+1}} = \text{solver}(\mathbf{x}_{\sigma_i}, \sigma_i, \mathbf{f}) \tag{61}$$

$$\hat{\mathbf{x}}^{\text{solver}}_{\sigma_{i+1}} = \mathbf{x}^{\text{solver}}_{\sigma_{i+1}} - h\beta z^2 \langle \mathbf{f}(\mathbf{x}_{\sigma_i}, \sigma_i),\, \hat{\mathbf{v}} \rangle\, \hat{\mathbf{v}} \tag{62}$$

DPM-Solver (2S) Lu et al. (2022) computes an intermediate state during its two-stage update. We denote this intermediate state as $\mathbf{x}_{\sigma_i+\delta}$, and construct the pair as $\{\mathbf{x}_{\sigma_i+\delta}, \mathbf{x}_{\sigma_i}\}$. DEIS Zhang & Chen (2023) computes its update using previous states in a multi-step formulation. We use the most recent previous state $\mathbf{x}_{\sigma_{i-1}}$ to construct the pair as $\{\mathbf{x}_{\sigma_i}, \mathbf{x}_{\sigma_{i-1}}\}$.

Table 8: **Evaluation of adaptive step-size control under different stiffness thresholds.** (A) ERK-Guid and (B) Heun are not adpot adaptive step-size. (C–G) adapt step-size using thresholds $\tau = 0.5, 1, 2, 5, 10$.

	Adaptive step-size	Threshold	NFE (Avg.) ↓	FD-DINOv2↓	FID ↓
(A) **ERK-Guid** (Ours)	✗	–	**63**	**86.2**	**2.56**
(B) Heun	✗	–	**63**	90.1	2.58
(C)	✓	$\tau = 0.5$	90.7	88.9	2.57
(D)	✓	$\tau = 1.0$	85.5	89.4	2.57
(E)	✓	$\tau = 2.0$	79.5	90.0	2.58
(F)	✓	$\tau = 5.0$	67.6	90.1	2.58
(G)	✓	$\tau = 10.0$	64.5	90.1	2.58

Table 9: **Quantitative results of predictor-corrector and ERK-Guid (Ours).**

Method	Stochasticity	r	NFE	FID	FD-DINOv2
Predictor-Corector Song et al. (2021b)	✓	0.05	64	2.65	91.5
	✓	0.10	64	2.66	93.0
	✓	0.15	64	2.90	98.9
	✗	0.01	64	2.61	90.6
	✗	0.02	64	2.73	87.4
	✗	0.03	64	3.33	88.6
ERK-Guid (**Ours**)	✗	-	**63**	**2.58**	**83.7**

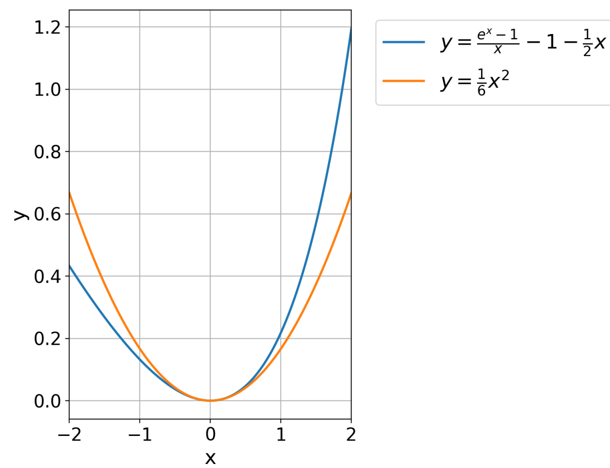


Figure 7: **The behavior of the funcition** $y = \alpha(x)$ **and** $y = \frac{1}{6}x^2$

B.5 Comparison with Adaptive step size

In Table 8, we analyze our proposed method with a classical adoption of stiffness during diffusion sampling. For experimental setting, we utilze a pre-trained EDM2 network on ImagNet-512. We use the same EDM discretization, and halved the step size whenever the stiffness estimator exceeded thresholds of 0.5, 1, 2, 5, or 10. Although a small threshold (e.g., $\tau = 0.5$) produced modest gains in FID and FD-DINOv2, it requires 1.44× more NFEs than ERK-Guid, making it substantially less efficient. Our method still achieved the best performance and efficiency compared to the baseline with various adaptive step-size.

B.6 Comparison with the predictor–corrector sampler

We evaluate the predictor–corrector (PC) sampler from Song et al. (2021b) by applying its corrector step after each Heun update. Since the corrector in Song et al. (2021b) is designed for the reverse SDE rather than an ODE, we also consider a deterministic variant obtained by removing its noise term. We vary the hyperparameter r and measure FID and FD-DINOv2 under a comparable NFE. In Table 9, ERK-Guid achieves strong performance compared to both variants of the PC sampler. The stochastic PC sampler consistently degrades performance across both metrics. In contrast, the deterministic PC sampler of Song et al. (2021b) exhibits a clear trade-off: as the correction scale r increases, FD-DINOv2 increases while FID decreases.

C Algorithm

In main paper Section 4.3, we introduce our ERK-Guid framework. For clarify our method, we provide Algorithm 1 in this part.

Algorithm 1 Sampling procedure with **ERK-Guid**

procedure OURS($\mathbf{f}_\theta(\mathbf{x};\sigma), \{\sigma_i\}_{i=0,\ldots,N}, w_{\text{stiff}}, w_{\text{con}}, \epsilon$)
  sample $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \sigma_0^2\mathbf{I}) \in \mathbb{R}^{H\times W\times C}$
  for $i \leftarrow 0$ to $N-1$ **do**
    $\mathbf{f}_i \leftarrow \mathbf{f}_\theta(\mathbf{x}_i;\sigma_i)$
    $h \leftarrow \sigma_i - \sigma_{i+1}$
    if $i \neq 0$ **then**
      $\Delta\mathbf{f}_i \leftarrow \mathbf{f}_i - \mathbf{f}_i^{\text{Euler}}$
      $\Delta\mathbf{x}_i \leftarrow \mathbf{x}_i - \mathbf{x}_i^{\text{Euler}}$
      $\mathbf{v}_i,\ \hat{\rho}\ \leftarrow \frac{\Delta\mathbf{f}_i}{||\Delta\mathbf{f}_i||},\ \frac{||\Delta\mathbf{f}_i||}{||\Delta\mathbf{x}_i||}$
      $\beta,\ z\ \leftarrow (\hat{\rho} > w_{\text{con}})\ ,\ w_{\text{stiff}}\, h\hat{\rho}$
      $\mathbf{g}_i \leftarrow \beta\, z^2\, (f_i^1 \cdot \mathbf{v}_i)\, \mathbf{v}_i$
    else
      $\mathbf{g}_i \leftarrow \mathbf{0}$
    end if
    $\mathbf{x}_{i+1}^{\text{Euler}} \leftarrow \mathbf{x}_i - h\mathbf{f}_i$
    if $i \neq N$ **then**
      $\mathbf{f}_{i+1}^{\text{Euler}} \leftarrow \mathbf{f}_\theta(\mathbf{x}_{i+1}^{\text{Euler}};\sigma_{i+1})$
      $\mathbf{x}_{i+1}^{\text{Heun}} \leftarrow \mathbf{x}_i - h\left(\frac{1}{2}\mathbf{f}_i + \frac{1}{2}\mathbf{f}_{i+1}^{\text{Euler}}\right)$
      $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_{i+1}^{\text{Heun}} - h\mathbf{g}_i$
      $\mathbf{Q} \xleftarrow{\text{buffer}} \mathbf{x}_{i+1}^{\text{Euler}}, \mathbf{f}_{i+1}^{\text{Euler}}$
    else
      $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_{i+1}^{\text{Euler}}$
    end if
  end for
  return $\mathbf{x}_N$
end procedure

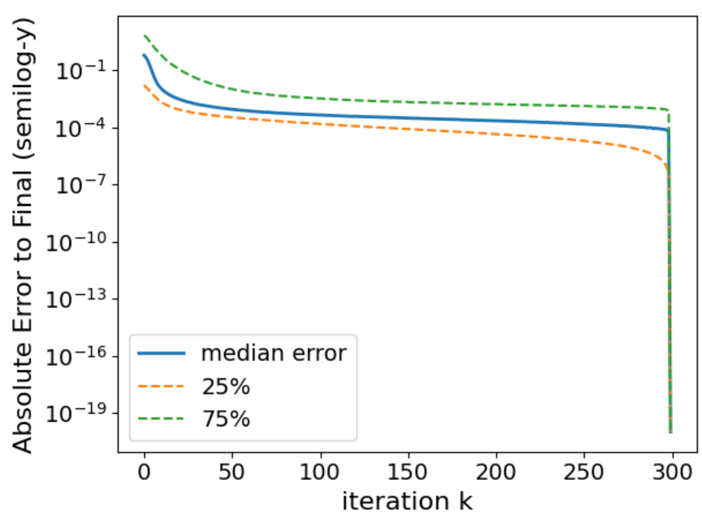


Figure 8: **Convergence of stiffness estimation under JVP power iteration.** The plot shows the absolute error between the estimated stiffness at iteration k and the final converged value. The solid line denotes the median across all seeds and timesteps, while the dashed lines indicate the 25th and 75th percentiles. Errors decrease rapidly and stabilize, demonstrating reliable convergence of the iteration procedure.

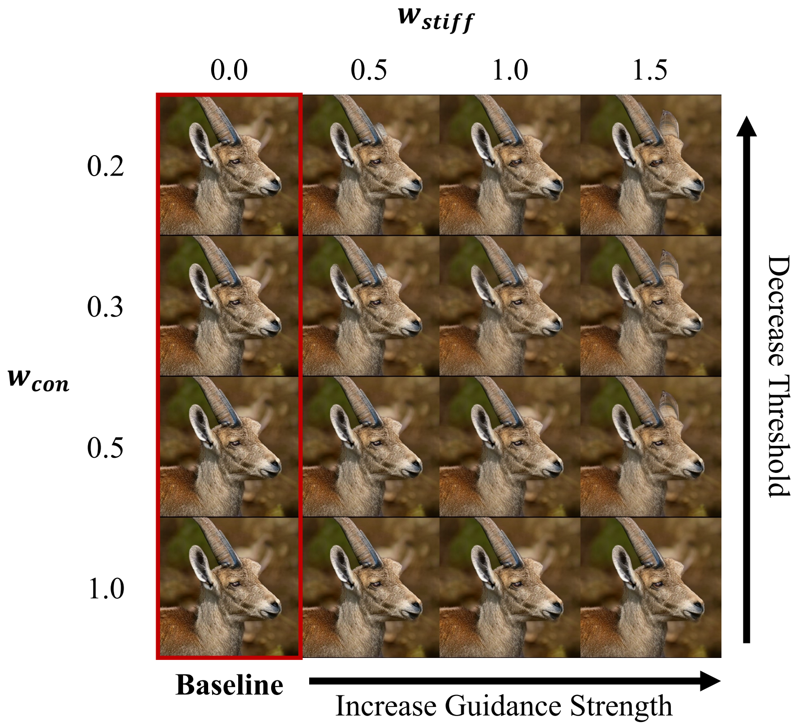


Figure 9: **Qualitative results of ERK-Guid across adjusting guidance scale.**

D LLM Usage

We only used a large language model as a writing assistant to refine phrasing, grammar, and clarity. It was not used for technical content, experiments, analyses, or results.

E Qualitative results

In this section, we present qualitative results of ERK-Guid on ImageNet 512×512. Figures 9 and 10 show that applying ERK-Guid enhances image fidelity when sufficient guidance is applied. These results demonstrate that our method effectively mitigates solver-induced errors during conditional updates along the sampling trajectory.

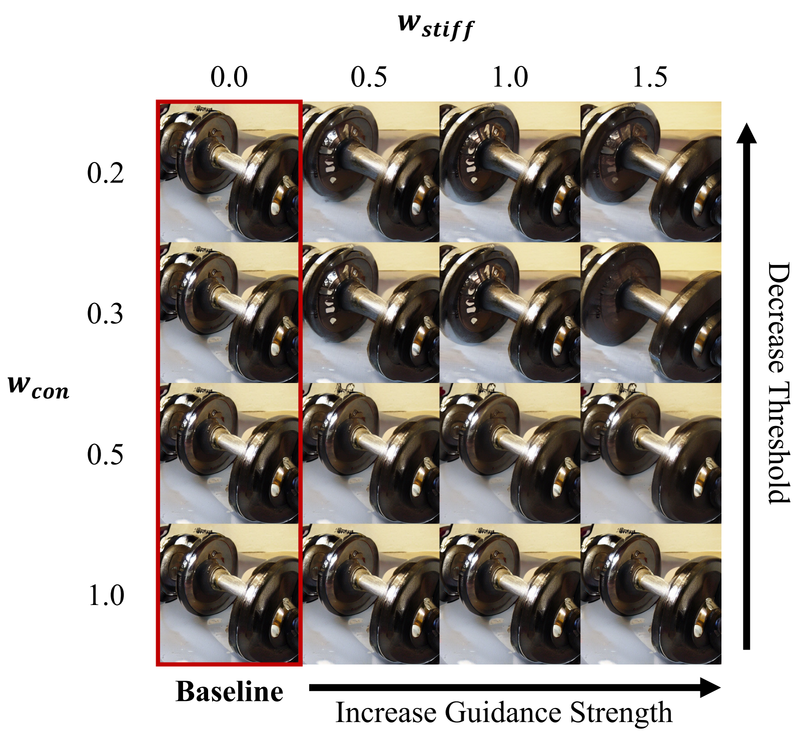


Figure 10: **Qualitative results of ERK-Guid across guidance scales.**