# Enhanced Diffusion Sampling via Extrapolation with Multiple ODE Solutions


Jinyoung Choi
Seoul National University
Seoul, Republic of Korea
jin0.choi@snu.ac.kr

Junoh kang
Seoul National University
Seoul, Republic of Korea
junoh.kang@snu.ac.kr

Bohyung Han
Seoul National University
Seoul, Republic of Korea
bhhan@snu.ac.kr



## Abstract

Diffusion probabilistic models (DPMs), while effective in generating high-quality samples, often suffer from high computational costs due to their iterative sampling process. To address this, we propose an enhanced ODE-based sampling method for DPMs inspired by Richardson extrapolation, which reduces numerical error and improves convergence rates. Our method, RX-DPM, leverages multiple ODE solutions at intermediate time steps to extrapolate the denoised prediction in DPMs. This significantly enhances the accuracy of estimations for the final sample while maintaining the number of function evaluations (NFEs). Unlike standard Richardson extrapolation, which assumes uniform discretization of the time grid, we develop a more general formulation tailored to arbitrary time step scheduling, guided by local truncation error derived from a baseline sampling method. The simplicity of our approach facilitates accurate estimation of numerical solutions without significant computational overhead, and allows for seamless and convenient integration into DDIM solver. Additionally, RX-DPM provides explicit error estimates, effectively demonstrating the faster convergence as the leading error term's order increases. Through a series of experiments, we show that the proposed method improves the quality of generated samples without requiring additional sampling iterations.



## CCS Concepts

• **Computing methodologies → Computer vision**.

## Keywords

Diffusion models, Richardson extrapolation,

**CIKM Reference Format:**
Jinyoung Choi, Junoh kang, and Bohyung Han. . Enhanced Diffusion Sampling via Extrapolation with Multiple ODE Solutions. Human-Centric AI: From Explainability and Trustworthiness to Actionable Ethics (co-located with CIKM '25), November 14, 2025, Seoul, Republic of Korea. 4 pages.


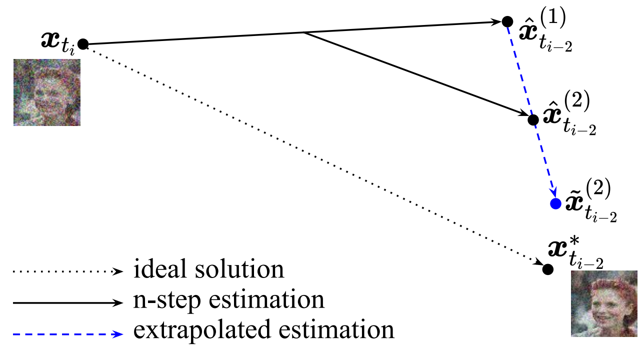


**Figure 1: Application of the proposed extrapolation on two denoising steps ($k = 2$) with time steps of $[t_i, t_{i-1}, t_{i-2}]$. $\hat{x}^{(n)}_{t_{i-2}}$ denotes that $n$ steps are used by the baseline sampler within the same interval. $\tilde{x}^{(2)}_{t_{i-2}}$ represents the extrapolated estimation using two ODE solutions at $t_{i-2}$, $\hat{x}^{(1)}_{t_{i-2}}$ and $\hat{x}^{(2)}_{t_{i-2}}$.**

## 1 Introduction

Diffusion probabilistic models (DPMs) have emerged as a powerful framework for generating high-quality samples in a wide range of applications and domains for images [2, 4, 10, 13], videos [5, 11, 14, 18], 3D shapes [15], *etc.* While DPMs demonstrate impressive performance in data fidelity and diversity, they also have limitations, particularly their computational inefficiency due to the sequential nature of sampling. Addressing this issue is crucial for enhancing the usability of DPMs in real-world scenarios, where time constraints are critical for practical deployment.

The generation process of DPMs can be formulated as a problem of finding solutions to SDEs or ODEs [13], where the truncation errors of the numerical solutions are highly correlated to the quality of the generated samples. To enhance the quality of these samples, it is essential to reduce truncation errors, which can be achieved by adopting advanced solvers or numerical techniques that improve the accuracy of numerical estimations. In this context, we aim to lower truncation errors by applying numerical extrapolation to existing sampling methods for DPMs. The key ingredient of the proposed method is Richardson extrapolation, a proven and widely used technique in the mathematical modeling of physical problems such as fluid dynamics and heat transfer, which demand high computational resources. While numerous variants and strategies have been studied [1, 8, 19], its application to DPMs remains unexplored. The method uses a simple linear combination of multiple numerical estimates from different resolutions of a grid to approximate the ideal solution, where the estimates are expected to converge as the resolution becomes finer, ultimately reaching the target value in the limit.

We propose an extrapolation algorithm that is applied repeatedly every $k$ denoising steps of an ODE-based sampling method to improve the accuracy of intermediate denoising steps. This is achieved by utilizing an additional ODE solution, which is estimated by a single step over an interval of $k$ time steps. Figure 1 illustrates this concept with $k = 2$ on time steps $[t_i, t_{i-1}, t_{i-2}]$,


*Human-Centric AI: From Explainability and Trustworthiness to Actionable Ethics, CIKM 2025,*

which forms a unit block of our extrapolation-based sampling. Two ODE solutions—single-step and two-step estimations at $t_{i-2}$ from $t_i$—can be leveraged to achieve an approximation closer to the ideal solution $\boldsymbol{x}^*_{t_{i-2}}$, which is unknown.

The standard Richardson extrapolation assumes a uniform discretization of the time grid. However, the uniform grid of denoising time steps might be suboptimal for DPMs; a smaller time interval near the clean sample is often more beneficial [6, 12] given the same number of steps. Considering such characteristics of DPMs and the advantages offered by specific time step discretizations, we propose an algorithm applicable to arbitrary schedules, with coefficients determined by the chosen configuration. We observe that our grid-aware approach yields better performance than conventional methods.

Although there exist other methods applying extrapolation techniques to diffusion models, their usages of extrapolation are somewhat different from ours. For example, [16, 17] utilize estimations from earlier steps to improve the estimation at the time step, $t_i$, whereas our approach adopts two denoised estimations at the same time step, $t_i$, to enhance accuracy at $t_i$. In addition, the main building block of our approach, Richardson extrapolation, is proven to enhance numerical accuracy and provides an explicit estimate of the error, which allows for a clear understanding of the convergence behavior. Furthermore, the implementation of our algorithm is simple and cost-effective because it requires no additional network evaluations and insignificant computational overhead to perform the extrapolation. We refer to the proposed sampling algorithm as RX-DPM.

Our main contributions are summarized below:

- We introduce an improved diffusion sampler, RX-DPM, inspired by Richardson extrapolation, which effectively increases the order of accuracy of existing ODE-based samplers without increasing NFEs.
- We systematically develop an algorithm for general DPM solvers with arbitrary time step scheduling starting from the derivation of a truncation error of the Euler method on a non-uniform grid.
- Our experiments demonstrate that RX-DPM exhibits strong generalization performance and high practicality, regardless of ODE designs and model architectures.

## 2 Preliminaries

### 2.1 Diffusion probabilistic models as solving an ODE

For $\boldsymbol{p}_0 = \boldsymbol{p}_{\text{data}}$ and $\boldsymbol{x} \in \mathbb{R}^d$, [6] defines a marginal distribution at $t$ as

$$\boldsymbol{p}_t(\boldsymbol{x}) = s(t)^{-d}\boldsymbol{p}(\boldsymbol{x}/s(t); \sigma(t)), \tag{1}$$

where $\boldsymbol{p}(\boldsymbol{x}; \sigma) = \boldsymbol{p}_{\text{data}} * \mathcal{N}(\boldsymbol{0}, \sigma(t)^2\boldsymbol{I})$, and $s(t)$ and $\sigma(t)$ are non-negative functions satisfying $s(0) = 1, \sigma(0) = 0$, and $\lim_{t\to\infty} \frac{\sigma(t)}{s(t)} = \infty$. The probability flow ODE with boundary condition $\boldsymbol{x}(T) \sim \boldsymbol{p}_T(\boldsymbol{x})$

$$d\boldsymbol{x} = [\dot{s}(t)/s(t) - s(t)^2\dot{\sigma}(t)\sigma(t)\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}/s(t); \sigma(t))]dt \tag{2}$$

matches the marginal distribution. By adopting the specific choices, $s(t) = 1$ and $\sigma(t) = t$ as in [6], Equation (2) is reduced as follows:

$$d\boldsymbol{x} = -t\nabla_{\boldsymbol{x}} \log \boldsymbol{p}(\boldsymbol{x}; t)dt, \quad \boldsymbol{x}(T) \sim \boldsymbol{p}_T(\boldsymbol{x}). \tag{3}$$

Diffusion models now learn the score function $\nabla_{\boldsymbol{x}} \log \boldsymbol{p}(\boldsymbol{x}; t)$, which is the only unknown component in the equation. For sufficiently large $T$, the marginal distribution $\boldsymbol{p}_T(\boldsymbol{x})$ can be approximated by $\mathcal{N}(\boldsymbol{x}; \boldsymbol{0}, T^2\boldsymbol{I})$ and the generation process is equivalent to solving for $\boldsymbol{x}(0)$ using Equation (3) with the boundary condition, $\boldsymbol{x}(T) \sim \mathcal{N}(\boldsymbol{0}, T^2\boldsymbol{I})$. Since the analytic solution of Equation (3) cannot be expressed in a closed form, numerical methods are used to solve the ODE. Given the time step scheduling, $0 = t_0 < t_1 < \ldots < t_N = T$, the solution is given by

$$\boldsymbol{x}(0) = \boldsymbol{x}(T) + \int_T^0 -t\nabla_{\boldsymbol{x}} \log \boldsymbol{p}(\boldsymbol{x}(t); t)dt \tag{4}$$

$$= \boldsymbol{x}(T) + \sum_{i=N}^{1} \int_{t_i}^{t_{i-1}} -t\nabla_{\boldsymbol{x}} \log \boldsymbol{p}(\boldsymbol{x}(t); t)dt, \tag{5}$$

where each integration from $t_i$ to $t_{i-1}$ can be approximated by ODE solvers such as the Euler method or Heun's method.

### 2.2 Richardson extrapolation

Let the exact and numerical solutions at $t = 0$ be $V^*$ and $V(h)$, respectively, where $h$ $(0 < h < 1)$ denotes the step size. If $V^* = \lim_{h\to 0} V(h)$ and the order of truncation error is known, Richardson extrapolation [9] identifies a faster converging sequence, $\tilde{V}(h)$. For instance, $V(h)$ with a truncation error in the order of $O(h^p)$ is expressed by

$$V^* = V(h) + ch^p + O(h^q) \tag{6}$$

for $0 < p < q$ and $c \neq 0$. Then, for a fixed constant $k > 1$,

$$V^* = V(h/k) + \frac{c}{k^p}h^p + O(h^q). \tag{7}$$

From Equations (6) and (7), eliminating the $h^p$ terms, we obtain the solution

$$\tilde{V}(h,k) = \frac{k^p V(h/k) - V(h)}{k^p - 1}, \tag{8}$$

which has a truncation error of $O(h^q)$, asymptotically smaller than $O(h^p)$.

## 3 RX-DPM

Before discussing the proposed method, we first outline the algorithmic development process for the most simplified problem and then explore an extension to a general DPM solver. The application of our method, RX-DPM, to a specific solver will be referred to as RX-[*SolverName*].

### 3.1 Truncation error of Euler method on non-uniform grid

We now derive the truncation error formula for the Euler method on a non-uniform grid, based on the local truncation error, which results from a single iteration. For intuitive clarity, we consider a one-dimensional ODE of the form

$$dx = f(x,t)dt,$$

where $f$ is a smooth function. Suppose that the numerical solution is obtained using the Euler method with the discretization points $[t_i, t_{i-1}, \dots, t_{i-k}]$ in a reverse time order, given the initial condition $x(t_i) = x_{t_i}$. From now on, we denote $\hat{x}^{(n)}_{t_j}$ as the numerical solution at $t_j$ obtained by $n$ iterations and $x^*_{t_j}$ as the exact solution at $t_j$. Given $h = t_i - t_{i-k}$ and $\lambda_j = \frac{1}{h}(t_{i-j+1} - t_{i-j})$ for $j = 1, \dots, k$, the local truncation error formula of the one-step Euler method, derived from the Taylor expansion, is expressed as

$$\hat{x}^{(1)}_{t_{i-1}} = x_{t_i} - \lambda_1 h f(x_{t_i}; t_i) = x^*_{t_{i-1}} - \frac{1}{2} x''_{t_i} \lambda_1^2 h^2 + O(h^3). \tag{9}$$

Then, the truncation error of the two-step numerical solution is derived as

$$\hat{x}^{(2)}_{t_{i-2}} = \hat{x}^{(1)}_{t_{i-1}} - \lambda_2 h f(\hat{x}^{(1)}_{t_{i-1}}) \tag{10}$$

$$= x^*_{t_{i-1}} - \frac{1}{2} x''_{t_i} \lambda_1^2 h^2 + O(h^3) - \lambda_2 h f(\hat{x}^{(1)}_{t_{i-1}}) \tag{11}$$

$$= x^*_{t_{i-1}} - \lambda_2 h f(x^*_{t_{i-1}}) - \frac{1}{2} x''_{t_i} \lambda_1^2 h^2 + O(h^3) \tag{12}$$

$$- \lambda_2 h f(\hat{x}^{(1)}_{t_{i-1}}) + \lambda_2 h f(x^*_{t_{i-1}}) \tag{13}$$

$$= x^*_{t_{i-2}} - \frac{1}{2} x^{*\prime\prime}_{t_{i-1}} \lambda_2^2 h^2 - \frac{1}{2} x''_{t_i} \lambda_1^2 h^2 + O(h^3) \tag{14}$$

$$(\because \text{Equation (9) and } f \text{ is smooth}) \tag{15}$$

$$= x^*_{t_{i-2}} - \frac{1}{2} x''_{t_i} (\lambda_1^2 + \lambda_2^2) h^2 + O(h^3) \quad (\because f \text{ is smooth}). \tag{16}$$

Inductively, we can obtain the truncation error for the $k$-step solution as

$$\hat{x}^{(k)}_{t_{i-k}} = x^*_{t_{i-k}} - \frac{1}{2} x''_{t_i} \sum_{j=1}^{k} \lambda_j^2 h^2 + O(h^3), \tag{17}$$

which approximates $x^*_{t_{i-k}}$ with a truncation error of $O(h^2)$.

## 3.2 RX-Euler

We now describe RX-Euler performing extrapolation every $k$ steps on the Euler method. Extrapolation is executed as a linear combination of two different numerical solutions $\hat{\boldsymbol{x}}^{(1)}_{t_{i-k}}$ and $\hat{\boldsymbol{x}}^{(k)}_{t_{i-k}}$ obtained by the Euler solver over a single step on the grid $[t_i, t_{i-k}]$ and $k$ steps on the grid $[t_i, t_{i-1}, \dots, t_{i-k}]$, respectively. To calculate coefficients for extrapolation, we use the truncation error derived in Section 3.1, which can be also applied to Equation (3) in Section 2.1, as the ideal score function is considered smooth; its derivative is Lipschitz continuous, referring to the equation in Appendix B.3 of Karras et al. [6]. From Equations (9) and (17), we derive the expressions for $\hat{\boldsymbol{x}}^{(1)}_{t_{i-k}}$ and $\hat{\boldsymbol{x}}^{(k)}_{t_{i-k}}$ for a constant $\boldsymbol{c}$ as follows:

$$\hat{\boldsymbol{x}}^{(1)}_{t_{i-k}} = \boldsymbol{x}^*_{t_{i-k}} + \boldsymbol{c} h^2 + O(h^3) \tag{18}$$

$$\hat{\boldsymbol{x}}^{(k)}_{t_{i-k}} = \boldsymbol{x}^*_{t_{i-k}} + \boldsymbol{c} \sum_{j=1}^{k} \lambda_j^2 h^2 + O(h^3). \tag{19}$$

Then, by solving the linear system of Equations (18) and (19), we approximate $\boldsymbol{x}^*_{t_{i-k}}$ through the following extrapolation:

$$\tilde{\boldsymbol{x}}^{(k)}_{t_{i-k}} = \frac{\hat{\boldsymbol{x}}^{(k)}_{t_{i-k}} - \sum_{j=1}^{k} \lambda_j^2 \hat{\boldsymbol{x}}^{(1)}_{t_{i-k}}}{1 - \sum_{j=1}^{k} \lambda_j^2}, \tag{20}$$

which involves a truncation error of $O(h^3)$, asymptotically smaller than $O(h^2)$.

In the sampling process, we set the initial condition at the next denoising step, $t_{i-k}$, as $\tilde{\boldsymbol{x}}^{(k)}_{t_{i-1}}$, and repeatedly perform the proposed extrapolation technique every $k$ steps. Because this approach provides provably more accurate solutions at every $k$ steps, we can reduce error propagation and expect better quality of generated examples.

The proposed method is applicable to first-order methods in general, including DDIM [12], which is arguably the most widely used DPM sampler. In this context, we bring the interpretation of DDIM as the Euler method applied to the following ODE:

$$d\boldsymbol{y} = \epsilon_\theta(\boldsymbol{x}(t), t) d\gamma, \tag{21}$$

where $\boldsymbol{y}(t) = \boldsymbol{x}(t)\sqrt{1 + \gamma(t)^2}$ and $\gamma(t) = \sqrt{\frac{1-\alpha_t^2}{\alpha_t^2}}$ in the variance-preserving diffusion process [13], *i.e.*, $\boldsymbol{p}_t(\boldsymbol{x}|\boldsymbol{x}_0) = \mathcal{N}(\alpha_t \boldsymbol{x}_0, (1 - \alpha_t^2)\boldsymbol{I})$. Thus, instead of using the time grid, we compute $\lambda_j$'s in Equation (20) in terms of $\gamma(t)$'s, replacing $t$ with the corresponding $\gamma(t)$ in the computation, while the other procedures remain unchanged.

RX-Euler (RX-DDIM) does not require additional NFEs beyond the number of time steps, as the first prediction of every $k$-step-interval can be stored during the computation of $\hat{\boldsymbol{x}}^{(k)}$ and reused to obtain $\hat{\boldsymbol{x}}^{(1)}$. The only extra computation involves a linear combination of two estimates, which is negligible compared to the forward evaluations of DPMs.

## 3.3 Analysis on global truncation errors

We perform an analysis on global truncation errors of the Euler method and RX-Euler under the same NFEs. Assume that we solve an ODE satisfying Lipschitz condition from $t = 1$ to $t = 0$ with NFEs $= N$.

*Euler.* Since the Euler method requires a single network evaluation for each time step, the number of time steps allowed is $N$. Since the local truncation error of the Euler method on step size of $h = 1/N$ is expressed as $ch^2 + O(h^3)$, the global truncation error is given by

$$(ch^2 + O(h^3)) \cdot N = \frac{c}{N} + O(N^{-2}) = O(N^{-1}). \tag{22}$$

*RX-Euler.* If RX-Euler performs extrapolation every $k$ steps, the extrapolation happens $N/k$ times, where $N$ is equal to the NFEs for the Euler method. The local truncation error for RX-Euler over each $k$ steps, which has the interval of $h = k/N$, is expressed as $c'h^3 + O(h^4)$ and therefore the global truncation error is given by

$$(c'h^3 + O(h^4)) \cdot \frac{N}{k} = \frac{k^2 c'}{N^2} + O(N^{-3}) = O(N^{-2}). \tag{23}$$

RX-Euler exhibits a higher convergence rate of the global truncation error compared to the Euler method by one order of magnitude. Using the same approach, we can also demonstrate that the proposed method achieves faster convergence of the global truncation error for higher-order solvers.

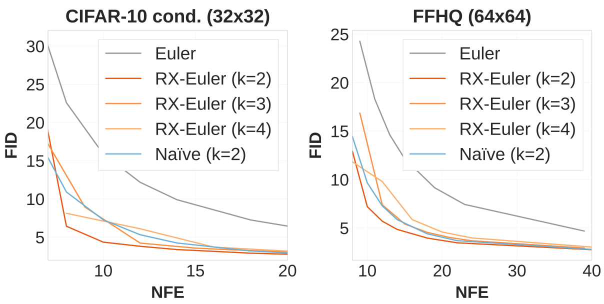


**Figure 2: Effect of extrapolation on the Euler method with different $k$'s.**

# 4 Experiment

## 4.1 Implementation details

We conduct the experiment with EDM [6], Stable Diffusion V2[1] [10], using their official implementations and provided pretrained models. Throughout all experiments, we retain the default settings from the official codebases, except for additional hyperparameters related to the proposed method. For experiments with EDM as a backbone, we generate 50K images and compute FID [3] using the evaluation code provided in their implementations. To evaluate Stable Diffusion V2 results, we use the PyTorch implementation for the computation of FID[2] and CLIP score[3] with the patch size of $32 \times 32$.

## 4.2 Validity test

We first evaluate the effectiveness of RX-Euler under the EDM backbone for $k \in \{2, 3, 4\}$, where smaller $k$ values correspond to more frequent extrapolation over the same number of time steps. As shown in Figure 2, RX-Euler consistently achieves significantly better FID scores than the Euler method across all values of $k$. In particular, when extrapolation is applied every two steps, *i.e.*, $k = 2$, our approach achieves the best performance across a wide range of NFEs. This indicates that more frequent extrapolation leads to more accurate intermediate predictions, effectively mitigating error accumulation in the final samples. Meanwhile, the curves for $k \geq 3$ remain closer to that of $k = 2$ than to the Euler method, empirically validating the reduced truncation error derived in Equation (20) for general $k$. In other words, even sparse extrapolation still has a significant impact on the output quality. For the rest of our results, we set $k = 2$.

We also compare the proposed method with conventional Richardson extrapolation, as formulated in Equation (8) with $k = 2$, which employs fixed coefficients. This baseline is labeled as Naïve ($k = 2$) in Figure 2. When comparing RX-Euler ($k = 2$) with Naïve ($k = 2$), we find that our method produces superior results. This implies that extrapolation coefficients adapted to arbitrary time step scheduling are more effective than fixed coefficients derived from uniformly discretized time steps. In other words, the proposed grid-aware coefficients enable more effective extrapolation than deterministic ones, better capturing the varying importance of precision in DPMs over time [6].

[1] https://github.com/Stability-AI/stablediffusion, v2-1_512-ema-pruned.ckpt
[2] https://github.com/mseitzer/pytorch-fid
[3] https://huggingface.co/openai/clip-vit-base-patch32

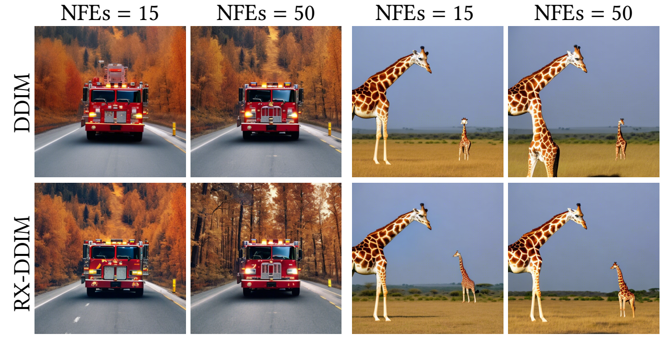


**Figure 3: Qualitative results of DDIM and RX-DDIM based on Stable Diffusion V2. RX-DDIM produces sharper and more detailed backgrounds, especially evident in the left example. On the right, RX-DDIM generates more realistic giraffes, whereas DDIM struggles at NFEs = 50, failing to properly render the giraffe.**

**Table 1: FID and CLIP scores of DDIM and RX-DDIM using Stable Diffusion V2.**

| NFEs | 15 | | 50 | |
|---|---|---|---|---|
| Method \ Metric | FID (↓) | CLIP (↑) | FID (↓) | CLIP (↑) |
| DDIM | 19.15 | **31.727** | 18.65 | 31.711 |
| RX-DDIM | **17.24** | 31.629 | **17.83** | **31.727** |

## 4.3 Comparisons on Stable Diffusion

We apply RX-DDIM to Stable Diffusion V2, which provides various conditional generations. For evaluation, we generate 10K $512 \times 512$ images from unique text prompts in the COCO2014 [7] validation set and compute FID and CLIP scores on resized $256 \times 256$ images. As shown in Table 1, our method also performs well on large models. However, we observe that RX-DDIM yields lower CLIP scores at NFEs = 15, which we attribute to classifier-guidance scales. According to [10], optimal classifier-free guidance scales differ across models. Since the default setting is tuned for DDIM, RX-DDIM may benefit from further optimization. Figure 3 presents qualitative comparisons between DDIM and RX-DDIM, highlighting the superior image quality of RX-DDIM. Notably, RX-DDIM generates images with more vivid colors, sharper textures, and more realistic object depictions, leading to an overall more natural appearance.

# 5 Conclusion

We introduced RX-DPM, an advanced ODE sampling method for DPMs that leverages extrapolation based on two ODE solutions derived from different discretizations of the same time interval. Our algorithm computes the optimal coefficients for arbitrary time step scheduling without additional training and incurs no extra NFEs by utilizing past predictions. This approach effectively reduces truncation errors, resulting in improved sample quality. Extensive experiments on well-established baseline models and datasets confirm that RX-DPM surpasses existing sampling methods, offering a more efficient and accurate solution for DPMs.

# References


[1] Mike A Botchev and Jan G Verwer. 2009. Numerical integration of damped Maxwell equations. *SIAM Journal on Scientific Computing* 31, 2 (2009), 1322–1346.

[2] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat GANs on image synthesis. In *NeurIPS*.

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.

[5] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. 2022. Video Diffusion Models. In *NeurIPS*.

[6] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the Design Space of Diffusion-Based Generative Models. In *NeurIPS*.

[7] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. *CoRR* abs/1405.0312 (2014). arXiv:1405.0312 http://arxiv.org/abs/1405.0312

[8] Shane A Richards. 1997. Completed Richardson extrapolation in space and time. *Communications in numerical methods in engineering* 13, 7 (1997), 573–582.

[9] Lewis Fry Richardson. 1911. IX. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London* 210, 459-470 (1911), 357–357.

[10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*.

[11] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. 2022. Make-A-Video: Text-to-Video Generation without Text-Video Data. In *ICLR*.

[12] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising diffusion implicit models. In *ICLR*.

[13] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *ICLR*.

[14] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. 2023. ModelScope Text-to-Video Technical Report. *arXiv preprint arXiv:2308.06571* (2023).

[15] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. 2022. LION: Latent Point Diffusion Models for 3D Shape Generation. In *NeurIPS*.

[16] Guoqiang Zhang, Niwa Kenta, and W. Bastiaan Kleijn. 2023. Lookahead Diffusion Probabilistic Models for Refining Mean Estimation. In *CVPR*.

[17] Qinsheng Zhang and Yongxin Chen. 2023. Fast Sampling of Diffusion Models with Exponential Integrator. In *ICLR*.

[18] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. 2022. MagicVideo: Efficient Video Generation With Latent Diffusion Models. *arXiv preprint arXiv:2211.11018* (2022).

[19] Zahari Zlatev, István Faragó, and Ágnes Havasi. 2010. Stability of the Richardson extrapolation applied together with the $\theta$-method. *J. Comput. Appl. Math.* 235, 2 (2010), 507–517.