Unleashing High-Quality Image Generation in Diffusion Sampling Using Second-Order Levenberg-Marquardt-Langevin

Fangyikang Wang[1*‡] Hubery Yin[2*] Lei Qian[1] Yinan Li[1] Shaobin Zhuang[3‡] Huminhao Zhu[1]
Yilin Zhang[1] Yanlong Tang[4] Chao Zhang[1†] Hanbin Zhao[1] Hui Qian[1] Chen Li[2]
[1]Zhejiang University [2]WeChat Vision, Tencent Inc [3]Shanghai Jiao Tong University [4]Tencent Lightspeed Studio

Abstract

The emerging diffusion models (DMs) have demonstrated the remarkable capability of generating images via learning the noised score function of data distribution. Current DM sampling techniques typically rely on first-order Langevin dynamics at each noise level, with efforts concentrated on refining inter-level denoising strategies. While leveraging additional second-order Hessian geometry to enhance the sampling quality of Langevin is a common practice in Markov chain Monte Carlo (MCMC), the naive attempts to utilize Hessian geometry in high-dimensional DMs lead to quadratic-complexity computational costs, rendering them non-scalable. In this work, we introduce a novel Levenberg-Marquardt-Langevin (LML) method that approximates the diffusion Hessian geometry in a training-free manner, drawing inspiration from the celebrated Levenberg-Marquardt optimization algorithm. Our approach introduces two key innovations: (1) A low-rank approximation of the diffusion Hessian, leveraging the DMs' inherent structure and circumventing explicit quadratic-complexity computations; (2) A damping mechanism to stabilize the approximated Hessian. This LML approximated Hessian geometry enables the diffusion sampling to execute more accurate steps and improve the image generation quality. We further conduct a theoretical analysis to substantiate the approximation error bound of low-rank approximation and the convergence property of the damping mechanism. Extensive experiments across multiple pretrained DMs validate that the LML method significantly improves image generation quality, with negligible computational overhead.

1. Introduction

The emerging diffusion models (DMs) [30, 74, 76, 77], generating samples of data distribution from initial noise, have been proven to be an effective technique for modeling complex distribution, especially in generating high-quality images [13, 31, 58, 62, 67, 70]. Training DMs can be viewed as using a neural network to match the (Stein) score function of the target distribution corrupted by different levels of noise. The sampling of DMs can be seen as running Langevin dynamics [65] using the learned diffused score and simultaneously slowly decreasing the noise level, which is called annealed Langevin dynamics [76].

Recently, many sampling methods have been proposed to enhance the quality of generated samples of pretrained DMs. The majority of these efforts focus on refining the denoising scheme, as seen in works such as DDIM [75], DPM-Solvers [51, 52], PNDM [50], and others [7, 85, 91–94]. Some also propose to select more critical denoising timesteps to enhance sampling quality [23, 35, 87, 88]. These efforts are primarily based on the analysis along the denoising path, yet they still perform first-order Langevin using the learned score within a specific noise level.

In the Langevin sampling area, it is common to employ additional second-order Hessian geometry to enhance the sampling quality. The additional second-order information can guide Langevin dynamics to take more accurate steps [12, 54], which is called Newton-Langevin [73]. A natural idea is that we can also leverage the Hessian geometry of DMs within each noise level to enhance the quality of diffusion sampling results. However, it is extremely challenging to calculate the Hessian geometry within the context of high-dimensional DMs [3], cause it requires quadratic-complexity computations. Current methods on diffusion Hessian either require auxiliary networks [15, 84] or maintain memory-intensive states [64], making them inefficient for enhancing DM sampling.

In this paper, we introduce a novel method for approximating the diffusion Hessian within a specific noise level. Notably, our approach is the first to approximate the diffusion Hessian and apply it to enhance the sampling quality of pretrained commercial-level DMs. Our method, inspired by the celebrated Levenberg-Marquardt method in optimization [45, 53, 68], is referred to as the Levenberg-Marquardt-Langevin (LML) method. Our approach introduces two key

*Equal contribution † Corresponding author
‡Work done as interns at WeChat Vision

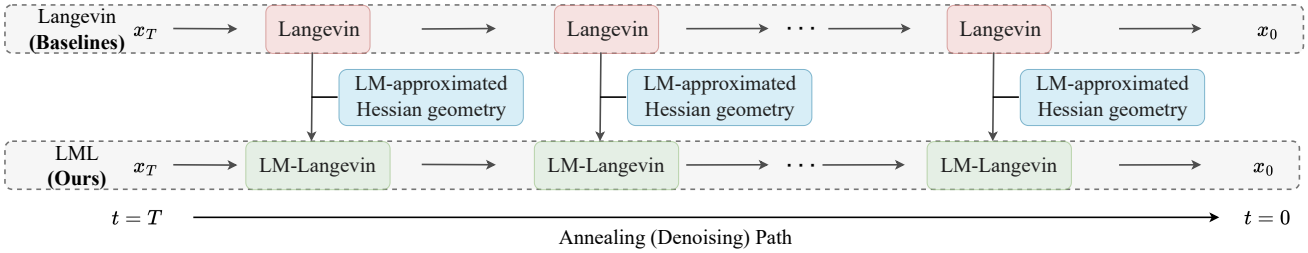


Figure 1. Schematic comparison between our LML method and baselines. While previous works mainly focus on intriguing designs along the annealing path to improve diffusion sampling, they leave operations at specific noise levels to be first-order Langevin. Our approach proposes to leverage the Levenberg-Marquardt approximated Hessian geometry to guide the Langevin update to be more accurate.

innovations: Initially, we derive a computationally tractable low-rank approximation of the diffusion Hessian by emulating the Gauss-Newton transformation, avoiding explicit quadratic-complexity computations. This approach capitalizes on the inherent structure of DMs to construct a low-rank approximation that captures critical geometric information. Following this, we stabilize the approximated Hessian using the damping mechanism, which addresses its ill-conditioning, and acquire its inverse for geometric guidance. Our method uses this LML-approximated Hessian geometry to enable the Langevin dynamics to take more accurate steps and improve image generation quality.

We also conduct comprehensive theoretical analyses for our low-rank approximation and damping mechanism. First, we establish the error bound for the low-rank approximation of the diffusion Hessian. Subsequently, we prove that the damping mechanism preserves an unbiased stationary measure and exhibit exponentially fast convergence in terms of χ^2-divergence.

Extensive experiments on multiple pretrained DMs, including CIFAR-10, CelebA-HQ, SD-15, SD2-base, SD-XL and PixArt-α, validate that the LML method significantly improves image generation quality with negligible computational overhead.

2. Preliminaries

Notation. The Euclidean norm over $\mathbb{R}^d$ is denoted by $\|\cdot\|$. Throughout, we simply write $\int g$ to denote the integral with respect to the Lebesgue measure: $\int g(x)\mathrm{d}x$. When the integral is with respect to a different measure μ, we explicitly write $\int g\mathrm{d}\mu$. The expectation and variance of $g(X)$when $X \sim p$ are respectively denoted $\mathbb{E}_\mu g = \int g\mathrm{d}\mu$ and $\mathrm{var}_\mu g := \int (g - \mathbb{E}_\mu g)^2 \mathrm{d}\mu$. When clear from context, we sometimes abuse notation by identifying a measure μ with its Lebesgue density. We use $\boldsymbol{I}_d$ to denote the d-dimensional identity matrix; when clear from context, we sometimes simply write $\boldsymbol{I}$. $\boldsymbol{H}_f$ and $\boldsymbol{J}_f$ denote the Hessian and Jacobian of f respectively.

2.1. Langevin Dynamics and Gradient Descent

In statistics, the *(Stein) score* of a distribution $p(\boldsymbol{x})$ is defined to be $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$. Given an initial value $\boldsymbol{x}_0 \sim \pi(\boldsymbol{x})$ with π being a prior distribution, Langevin dynamics (LD) can produce samples from $p(\boldsymbol{x})$ using only the score function $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$ following the SDE,

$$\mathrm{d}\boldsymbol{x}_t = \nabla_{\boldsymbol{x}} \log p\left(\boldsymbol{x}_t\right) \mathrm{d}t + \sqrt{2}\mathrm{d}B_t, \tag{1}$$

where B_t is the standard d-dimensional Brownian Motion (BM). The distribution of $\boldsymbol{x}_t$ equals $p(\boldsymbol{x})$ when $T \to \infty$, in which case $\boldsymbol{x}_t$ becomes an exact sample from $p(\boldsymbol{x})$ under some regularity conditions [86]. Strictly speaking, a Metropolis-Hastings adjustment is needed, but it can often be ignored in practice [9]. There is a deep connection involving the distribution of $\boldsymbol{x}_t$ in LD to the renowned Gradient Descent method (GD) [56] in optimization. The marginal distribution of a Langevin process $(\boldsymbol{x}_t)_{t\geq 0}$ evolves according to a GD, over the Wasserstein probability space, that minimizes the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(\cdot \| \pi)$[2, 36, 81].

2.2. Diffusion Models and Annealed Langevin

Suppose that we have a d-dimensional random variable $\boldsymbol{x}(0) \in \mathbb{R}^d$ following an unknown target distribution $p_0(\boldsymbol{x}_0)$. Diffusion Models (DMs) define a forward process $\{\boldsymbol{x}(t)\}_{t\in[0,T]}$ with $T > 0$ starting with $\boldsymbol{x}(0)$, such that the distribution of $\boldsymbol{x}(t)$ conditioned on $\boldsymbol{x}(0)$ satisfies

$$p_{t|0}(\boldsymbol{x}(t)|\boldsymbol{x}(0)) = \mathcal{N}(\boldsymbol{x}(t); \alpha(t)\boldsymbol{x}(0), \sigma^2(t)\mathbf{I}), \tag{2}$$

where $\alpha(\cdot), \sigma(\cdot) \in \mathcal{C}([0, T], \mathbb{R}^+)$ have bounded derivatives, and we denote them as α_t and σ_t for simplicity. The choice for α_t and σ_t is referred to as the noise schedule of a DM.

DMs, both SMLD [76] and DDPM [30], can be seen as learning a network to match the score of the diffused distribution $\log p_t(\boldsymbol{x}_t)$ at different noise levels. In practice, DMs usually use $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}(t), t)$ to estimate $-\sigma(t)\nabla_{\boldsymbol{x}(t)} \log p_t(\boldsymbol{x}(t), t)$ via optimizing the following denoising score matching objective:

$$\mathbb{E}_t \left\{ \lambda_t \mathbb{E}_{x_0, x_t} \left[\| s_\theta(x_t, t) - \nabla_{x_t} \log p_(x_t, t|x_0, 0) \|^2 \right] \right\}. \tag{3}$$

The sampling process of diffusion models can be seen as annealed Langevin dynamics [76], which executes Langevin updates at each noise level while progressively reducing the noise scale.

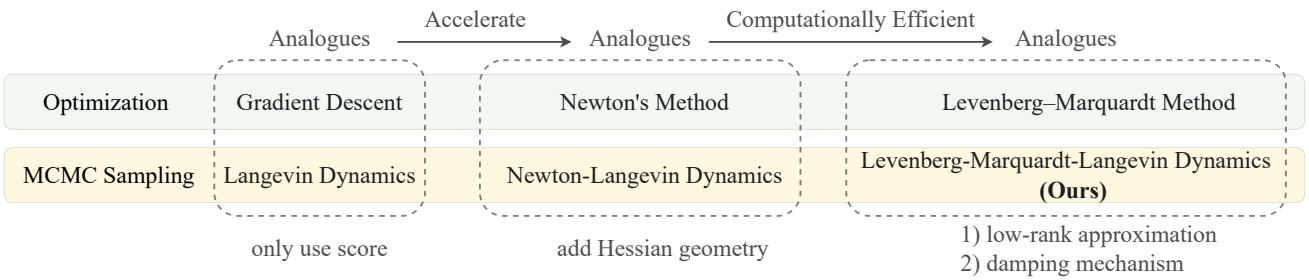


Figure 2. The relation between optimization algorithms and MCMC sampling algorithms. We initially wanted to develop a diffusion sampler utilizing Hessian geometry, following the path of Newton-Langevin dynamics [73]. However, this approach proved to be highly computationally expensive within the DM context. Drawing inspiration from the Levenberg-Marquardt method used in optimization, our method incorporates low-rank approximation and damping techniques. This enables us to obtain the Hessian geometry in a computationally affordable manner. Subsequently, we use this approximated Hessian geometry to guide the Langevin updates.

2.3. Langevin Guided by Hessian Geometry

Newton's method (also called Newton–Raphson) [60] is a famous optimization method that utilizes the second-order Hessian geometry to improve the GD. Developed as an analogy to Newton's method, the Newton-Langevin dynamics [54] utilize Hessian geometry to generate samples that adhere to the SDE

$$\mathrm{d}\boldsymbol{x}_t = \left[\nabla^2_{\boldsymbol{x}} \log p\left(\boldsymbol{x}_t\right)\right]^{-1} \nabla_{\boldsymbol{x}} \log p\left(\boldsymbol{x}_t\right) \mathrm{d}t + \sqrt{2}\mathrm{d}B'_t, \quad (4)$$

where the B'_t is the BM scaled by the square-rooted Hessian geometry. Calculating the Hessian geometry of high-dimensional distributions poses a significant challenge. Some studies have attempted to alleviate this issue through the Quasi-Newton technique [22, 73]. However, these approaches fail to address the problem in the context of DM-scale dimensional data. For instance, the Stable Diffusion-v1.5 model [67] features a latent dimension of 16384, resulting in a Hessian matrix of 16384×16384. Current methods on diffusion Hessian either require auxiliary networks training [15, 84] or maintain memory-intensive states [64]. To the best of our knowledge, there is currently no effective method available to access the diffusion Hessian of advanced commercial-level DMs like SD-XL.

2.4. Levenberg-Marquardt Method

In the field of optimization, the Levenberg-Marquardt (LM) method [68] is proposed as a computationally friendly and stabilized analogue to Newton's method. Specifically, it modifies the computation of Hessian geometry in Newton's method in two key aspects:

- **Low-rank approximated Hessian** In the context of least squares problems, the Levenberg-Marquardt method often constructs a low-rank estimation of the Hessian geometry. Specifically, when $f(\boldsymbol{x}) = \sum_{i=1}^{m} r_i(\boldsymbol{x})^2$, the Levenberg-Marquardt method approximates the Hessian into the following form, which is constructed from the Jacobians $\boldsymbol{J}_f(\boldsymbol{x})$.

$$\boldsymbol{H}_f(\boldsymbol{x}) \approx 2\boldsymbol{J}_f(\boldsymbol{x})^\top \boldsymbol{J}_f(\boldsymbol{x}). \quad (5)$$

  This low-rank approximation technique is also referred to as the Gauss-Newton method in some literature [16]. However, it is still unclear how to obtain a Levenberg-Marquardt-type low-rank estimation of the Hessian in the context of diffusion models.
- **Damping mechanism** The Levenberg-Marquardt Method introduces an additional damping identity matrix to the Hessian geometry. That is, it replaces the pure Hessian geometry $[\boldsymbol{H}_f(\boldsymbol{x}_k)]^{-1}$ in Newton's method with a combination expressed as $[\boldsymbol{H}_f(\boldsymbol{x}_k) + \lambda \boldsymbol{I}]^{-1}$. Consequently, the Levenberg-Marquardt Method with damping is expressed as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \left[\boldsymbol{H}_f(\boldsymbol{x}_k) + \lambda \boldsymbol{I}\right]^{-1} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k). \quad (6)$$

  Empirical evidence suggests that the damping mechanism contributes to numerical stabilization [41], since $\lambda \boldsymbol{I}$ can resolve the ill-conditioning of the Hessian. While the damping mechanism can be perceived as a trust region approach, it is more intuitive to view $\boldsymbol{H} + \lambda \boldsymbol{I}$ as a geometrical interpolation between $\boldsymbol{H}$ and $\boldsymbol{I}$.

3. Methodology

3.1. Low-rank Approximation of Diffusion Hessian

Inspired by the low-rank estimation technique of the Hessian in the Levenberg-Marquardt method, we derive a similar low-rank estimation for the diffusion Hessian. Specifically, we follow the intuition of the Levenberg-Marquardt method to approximate the diffusion Hessian by simplifying the second-order partial derivatives. The low-rank approximate Hessian of diffusion models that we obtained is shown below. It is important to note that it includes noise schedule related coefficients, which distinguishes it from the least squares problem case in Eq. 5.

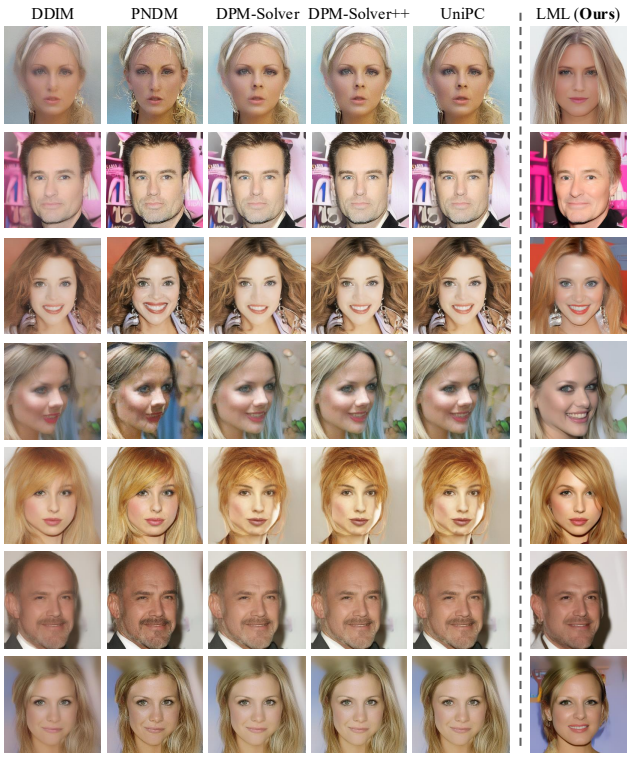


Figure 3. **Visual comparison** of images generated by our LML method and other methods, using the pre-trained LDM on CelebA-HQ 256×256 [39] with the same seeds. It shows that our LML method contributes to more vivid and detailed generated images.

Proposition 1. *Let p_t be the diffused marginal distribution at time t of the diffusion process, Eq. 2 and ε_θ is learned via Eq. 3, then the Hessian derivative of its log-density function $\nabla^2_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ has a low-rank approximation form of $\frac{1}{\sigma(t)^2\|\varepsilon_\theta\|^2}\varepsilon_\theta\varepsilon_\theta^\top$.*

Notice that this approximation form is a scaled outer-product of the score. The detailed derivation can be found in the Supplementary A.2. For here and for the remainder of the paper, we will denote $\varepsilon_\theta(\boldsymbol{x}(t), t)$ as ε_θ, provided there is no risk of ambiguity. In section 4.1, we will show that this low-rank approximation possesses an error bound, ensuring that our approximation does not generate excessive errors.

3.2. Damping Mechanism

Similar to the scenario in the Levenberg-Marquardt method, we have obtained an accessible approximation of the diffusion Hessian as presented in Proposition 1. However, this form is significantly ill-conditioned, making its inverse impossible to calculate. Specifically, the outer-product Hessian in Proposition 1 has an infinite condition number [32]; hence, its inverse strictly does not exist in mathematical terms. To address this issue, we propose adopting the damping mechanism used in the LML method. We introduce an additional damping identity matrix to the approximate Hessian as shown in Proposition 1. This damped Hessian geometry is used in place of the full Hessian in the Newton Langevin in Eq. 4, we consequently derive the following damping dynamics:

$$\mathrm{d}\boldsymbol{x}_t = \left[\nabla^2_{\boldsymbol{x}} \log p\left(\boldsymbol{x}_t\right) + \lambda \boldsymbol{I}\right]^{-1} \nabla_{\boldsymbol{x}} \log p\left(\boldsymbol{x}_t\right) \mathrm{d}t + \sqrt{2}\mathrm{d}B'_t, \tag{7}$$

where the B'_t is the BM scaled by the square-rooted LM approximated Hessian geometry. Drawing inspiration from the Levenberg-Marquardt method, we refer to this damping Hessian Langevin in Eq. 7 as the Levenberg-Marquardt-Langevin dynamics (LML dynamics). Similar to the Levenberg-Marquardt method, the damping coefficient λ in Eq. 7 can be interpreted as an interpolation coefficient between the Hessian geometry and Identity geometry. Specifically, as $\lambda \to \infty$, Eq. 7 would degenerate to the standard Langevin with normalization on the geometry. We observe that by adding the Levenberg-Marquardt damping identity matrix to the outer-product form of the Hessian in Equation Proposition 1, its inverse can be conveniently computed using the Sherman-Morrison formula [72].

$$\begin{aligned}
\boldsymbol{H}^{-1}_{LM}(\boldsymbol{x}_t, \lambda) &= \left[\nabla^2_{\boldsymbol{x}} \log p\left(\boldsymbol{x}_t\right) + \lambda \boldsymbol{I}\right]^{-1} \\
&\approx \left[\frac{1}{\sigma(t)^2\|\varepsilon_\theta\|^2}\varepsilon_\theta\varepsilon_\theta^\top + \lambda \boldsymbol{I}\right]^{-1} \\
&= \frac{1}{\sigma(t)^2\|\varepsilon_\theta\|^2}\left[\varepsilon_\theta\varepsilon_\theta^\top + \lambda' \boldsymbol{I}\right]^{-1} \\
&= \frac{1}{\lambda'\sigma(t)^2\|\varepsilon_\theta\|^2}\left[\boldsymbol{I} - \frac{\varepsilon_\theta\varepsilon_\theta^\top}{\lambda' + \|\varepsilon_\theta\|^2}\right],
\end{aligned} \tag{8}$$

where $\lambda' = \sigma(t)^2\|\varepsilon_\theta\|^2\lambda$. We also point out that the coefficients $\frac{1}{\lambda'\sigma(t)^2\|\varepsilon_\theta\|^2}$ will be absorbed in the discretizing stepsize of the LML dynamics and only affect the norm. The geometry direction essence is in the $\left[\boldsymbol{I} - \frac{\varepsilon_\theta\varepsilon_\theta^\top}{\lambda'+\|\varepsilon_\theta\|^2}\right]$ part. In the remainder of this paper, we will slightly abuse notation by not distinguishing between λ and λ'. This simplification should not lead to any confusion.

3.3. Annealed Levenberg-Marquardt-Langevin

To utilize the LML for diffusion sampling, we perform the discretization of Eq. 7 at each nose level, and gradually decrease the noise level. We called this annealed LML, and point out that its continuous-time form would converge to the following SDE:

$$\mathrm{d}x_t = \left[f_t\boldsymbol{x}_t - g_t^2\boldsymbol{H}^{-1}_{LM}\nabla_x \log p_t(\boldsymbol{x}_t)\right]\mathrm{d}t + g_t\mathrm{d}B'_t. \tag{9}$$

Similar to the connection of the reverse SDE and diffusion ODE [77], the LM reverse SDE in Eq. 9 also has an associated ODE, which is a deterministic process that shares the

Algorithm 1 Levenberg-Marquardt-Langevin (LML) diffusion sampler

1: **Input**: pretrained diffusion model noise predictor $\boldsymbol{\varepsilon}_\theta$, number of timesteps N, noise schedule $\{\alpha_t\}$ and $\{\sigma_t\}$, Levenberg-Marquardt damping coefficient $\lambda > 0$, EMA coefficient κ.
2: Initiate $\boldsymbol{x}_{\text{list}} = []$, $\widetilde{\boldsymbol{H}}_{N+1}^{-1} = \boldsymbol{I}$
3: Sample $\boldsymbol{x}_N \sim \mathcal{N}(0, \sigma_{t_N} I)$.
4: **for** $i = N, N-1, ..., 1$ **do**
5: $\quad \boldsymbol{\varepsilon}_i = \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_i, i)$
6: $\quad$ **if** $i \neq N$ **then**
7: $\quad\quad \widetilde{\boldsymbol{\varepsilon}}_i = \kappa * \boldsymbol{\varepsilon}_{i+1} + (1-\kappa) * \boldsymbol{\varepsilon}_i$
8: $\quad$ **else**
9: $\quad\quad \widetilde{\boldsymbol{\varepsilon}}_i = \boldsymbol{\varepsilon}_i$
10: $\quad$ **end if**
11: $\quad \widetilde{\boldsymbol{H}}_i^{-1} = \boldsymbol{I} - \frac{\widetilde{\boldsymbol{\varepsilon}}_i \widetilde{\boldsymbol{\varepsilon}}_i^\top}{\lambda + \|\widetilde{\boldsymbol{\varepsilon}}_i\|^2}$ {Levenberg-Marquardt approximate Hessian geometry}
12: $\quad \boldsymbol{\varepsilon}_i^{LM} = \widetilde{\boldsymbol{H}}_i^{-1} \boldsymbol{\varepsilon}_i$ {Apply the approximate Hessian geometry}
13: $\quad \boldsymbol{\varepsilon}_i^{LM} = \frac{\|\boldsymbol{\varepsilon}_i\|}{\|\boldsymbol{\varepsilon}_i^{LM}\|} \boldsymbol{\varepsilon}_i^{LM}$ {Geometrical normalization}
14: $\quad \boldsymbol{x}_{i-1} = \text{DPM-Solver}(\boldsymbol{x}_{\text{list}}, \boldsymbol{\varepsilon}_i^{LM}, i)$
15: $\quad \boldsymbol{x}_{\text{list}} = \boldsymbol{x}_{\text{list}}.\text{append}(\boldsymbol{x}_{i-1})$
16: **end for**
17: **Output**: $\boldsymbol{x}_0$

same single-time marginal distribution:

$$\mathrm{d}x_t = \left[f_t \boldsymbol{x}_t - \frac{1}{2} g_t^2 \boldsymbol{H}_{LM}^{-1} \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) \right] \mathrm{d}t. \tag{10}$$

The following section will develop a practical LML sampler based on this deterministic LML ODE.

Remark 1. *Our LM process can be viewed as the Legendre dual of the classical diffusion process with a transform map of* $\nabla \log p_t(\cdot) + \lambda \|\cdot\|$. *See discussions in Supp. D.3.*

3.4. Levenberg-Marquardt-Langevin Sampler

We implement a diffusion sampler based on the LML diffusion ODE in Eq. 10, because it is suggested that deterministic diffusion samplers are far more efficient than the stochastic ones [40]. Our approach is generally consistent with the practices outlined in [76]. The scheme for our LML diffusion sampler is illustrated in Algorithm 1. Primarily, our sampler substitutes the first-order Langevin dynamics with LML at each noise level. At each noise level, we initially compute the LM low-rank approximated and damping Hessian geometry, $\widetilde{\boldsymbol{H}}_i^{-1}$, as detailed in step 11 of Algorithm 1. We incorporate the network output from previous steps, as shown in step 7, following studies suggesting that this mixture is closer to the underlying true score due to the manifold's curvature [50]. We then apply this approximate Hessian geometry to our initial gradient, $\boldsymbol{\varepsilon}_i$, to obtain

Figure 4. **Qualitative comparison** of our LML method and other methods in text-guided image generation. The evaluation was performed on SD-15 [67], using 10 NFEs and the same seeds.

the Hessian-guided gradient, $\boldsymbol{\varepsilon}_i^{LM}$, as detailed in step 12. In terms of the stepsize for our LML sampler, in contrast to some sophisticated ways of choosing the stepsize of second-order methods [25, 43], we follow the approach of [20] to ensure that the Hessian geometry has a unit spectrum, which can be easily implemented through a normalization operation on the Hessian guided gradient $\boldsymbol{\varepsilon}_i^{LM}$ as illustrated in step 13. The rescaling of BM in Eq. 7 can also be absorbed into this normalization operation. Once the Hessian-guided gradient, $\boldsymbol{\varepsilon}_i^{LM}$, is obtained, we adopt the DPM-Solver as our denoising scheme to calculate the next state, $\boldsymbol{x}_{i-1}$, as shown in step 14.

It is worth noting that our LML does not require additional training, components, or network access. Instead, we only need several additional tensor arithmetic operations.

4. Theoretical Analysis

In this section, we present rigorous theoretical analyses to substantiate the correctness and efficacy of our LML.

4.1. Analysis on Low-rank Approximation

Here, we establish the error bound of the low-rank approximation of the diffusion Hessian in Proposition 1, thereby ensuring that our approximation does not introduce exces-

Figure 5. **Qualitative comparison** of our LML method and other methods in text-guided image generation. The evaluation was performed on SD2-base [67], using 10 NFEs and the same seeds.

sive errors. Given that we are evaluating the accuracy of a matrix-valued approximation, we choose to use the Hilbert-Schmidt norm [24] as a criterion.

Proposition 2. *Assume that the norm of $\boldsymbol{x}_t$ is bounded by δ_1, the approximation error on $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)$ is denoted as δ_2, δ_3 denote the bound on the second partial derivative of δ_1 w.r.t. $\boldsymbol{x}_t$, and $\mathcal{D}_y$ denote the dataset diameter. The approximation error of the LM low-rank Hessian, as referenced in Proposition 1, is at most $(\delta_1 + \alpha_t\delta_2 + \alpha_t\mathcal{D}_y)\left(2 + \delta_3 + 2\frac{\alpha_t^2}{\sigma_t^2}\mathcal{D}_y^2\right)$ when measured in terms of the Hilbert–Schmidt norm.*

Our analysis relies on the analytical form diffusion Fisher [84]. The detailed proof can be found in Supp. A.6.

4.2. Analysis on Damping Mechanism

Subsequently, we demonstrate the unbiased, exponentially fast convergence property of the damping mechanism.

4.2.1. Stationary Measure

We confirm that the stationary measure of the damping dynamics, as defined in Eq. 7, aligns with the target diffused distribution. This ensures that the damping updates still guide samples towards the correct data distribution.

Proposition 3. *Under mild regularity conditions, the stationary distribution of the damping dynamics in Eq. 7 exists and is unique, which also coincides with the marginal distribution $p_t(\boldsymbol{x}_t)$ at every noise level.*

We achieve this analysis by the Fokker-Planck equation [63]. The detailed proof can be found in Supp. A.7.

4.2.2. Ergodic Convergence Rate

We also want to determine the speed at which our damping dynamics converge. We demonstrate that our damping dynamics exhibit a satisfyingly fast, exponential convergence rate towards the target distribution.

Proposition 4. *Let μ_t be the evolving distribution of the damping dynamics in Eq. 7. We have that μ_t converges to the stationary distribution at an exponential ergodic convergence rate in terms of χ^2-distance at every noise level, under certain regularity conditions.*

We achieve this analysis by determining the relationship between Eq. 7 and the Mirror-Langevin [12]. The formal version and detailed proof can be found in Supp. A.8.

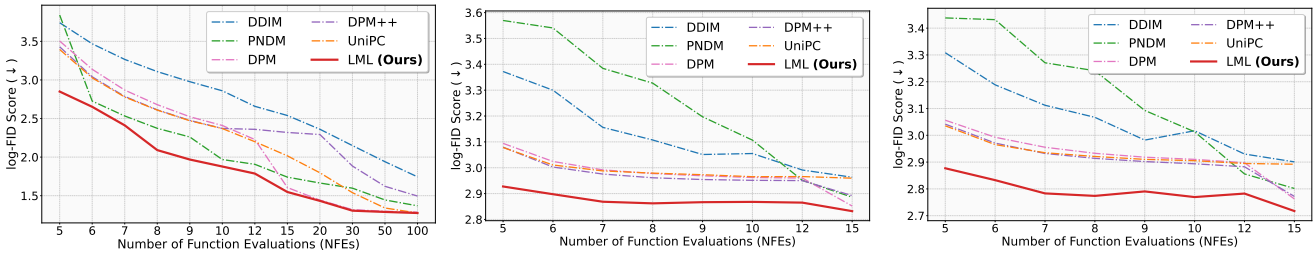
Remark 2. *Once we establish the exponential ergodic convergence rate in terms of χ^2-distance, The exponential convergence results for total variation distance [80], Hellinger distance [28], KL divergence [44] as illustrated in [4, §2.4]. And the exponential convergence results on Wasserstein distance [81] can also be established with a log-Sobolev assumption, as illustrated in [14].*

5. Experiments

In this section, we validate the enhanced sampling quality of our LML method across various pretrained DMs. We evaluate in both pixel-space, latent-space, and text-guided conditional generation scenarios. We select several most commonly-used and advanced sampling methods such as DDIM [77], PNDM [50], DPM-Solver [51], DPM-Solver++ [52] and UniPC [92] as baselines. We will also demonstrate the computational efficiency of our LML method. All experiments are executed using open-source, pretrained DMs, with the datatype set to float32 and the timestep scheme as uniform. More experimental details and results are deferred to Supp. B and C.

5.1. Pixel-Space Image Generation

We initially compare the unconditional sampling quality of our LML method with baselines on the CIFAR-10 dataset [1]. For each sampler, we generate 50,000 samples for FID evaluation. As illustrated in Table 1 and Figure 6a, our approach improves the sampling performance of the baseline Langevin methods in most NFE scenarios.



(a) CIFAR-10 (b) SD-15 on MS-COCO (c) SD2-base on MS-COCO

Figure 6. This line chart compares the log-FID scores (↓) on (a) CIFAR-10, (b) MS-COCO with SD-15 and (c) MS-COCO with SD2-base.

Table 1. Comparison of different samplers on FID score(↓) on CIFAR-10 unconditional generation. Best results are **bolded** and the second best results are underlined. The FID scores were obtained by generating 50,000 samples, and all samplers were tested using the same seeds on the same checkpoint. It is shown that our LML always achieves the best or second best FID across all different NFEs.

Methods	FID (↓) on CIFAR-10 generation											
	5 NFEs	6 NFEs	7 NFEs	8 NFEs	9 NFEs	10 NFEs	12 NFEs	15 NFEs	20 NFEs	30 NFEs	50 NFEs	100 NFEs
DDIM [75]	42.17	32.09	26.21	22.38	19.64	17.45	14.27	12.67	10.60	8.58	6.96	5.72
PNDM [50]	46.56	15.27	12.59	10.73	9.59	7.16	6.73	5.70	5.28	4.94	4.24	3.93
DPM-Solver [51]	33.26	23.06	17.61	14.58	12.47	11.13	9.28	4.98	4.23	3.74	3.66	3.62
DPM-Solver++ [52]	30.87	20.92	16.22	13.63	11.89	10.71	10.59	10.17	9.90	6.58	5.06	4.46
UniPC [92]	29.81	20.64	16.10	13.59	11.84	10.70	9.02	7.51	6.03	4.66	3.82	**3.58**
LML **(Ours)**	**17.28**	**14.18**	**11.15**	**8.09**	**7.16**	**6.54**	**5.97**	**4.70**	**4.19**	**3.69**	**3.63**	**3.58**

Table 2. Comparison of different samplers on CelebA-HQ unconditional generation. Best results are **bolded** and the second best results are underlined.

Methods	Colorful	Face Quality		Aesthetic		
	ColorS(↑)	FS(↑)	DFIQA(↑)	PicS(↑)	EAT(↑)	Laion(↑)
DDIM [75]	34.13	4.85	0.539	19.85	4.61	5.24
PNDM [50]	38.29	4.89	0.553	19.70	4.28	5.11
DPM [51]	34.85	5.06	0.566	20.00	4.51	5.29
DPM++ [52]	35.08	5.08	0.560	19.97	4.46	5.26
UniPC [92]	35.95	5.09	0.568	19.97	4.47	5.26
LML**(Ours)**	**40.53**	**5.24**	**0.607**	**20.98**	**4.75**	**5.37**

5.2. Latent-Space Image Generation

We evaluated our LML on the LDM [67] that was trained on CelebA-HQ [39] at a resolution of 256×256. In Table 2, we employed five metrics spanning three aspects to demonstrate the superiority of LML. These aspects include colorfulness [26], face quality (FS [46] and DFIQA [10]), and human-preference-aesthetic scores (PicS [42], EAT [27] and Laion-Aes [71]). Additionally, we present a visual comparison in Figure 3. All these experiments were carried out with a NFE of 10. The results were tested by averaging over 1000 samples and the same seeds.

5.3. Text-guided Image Generation

5.3.1. Qualitative Comparison

Qualitative comparison on SD-15 and SD2-base generation is provided in Figure 4 and 5, clearly indicating that our LML method enhances the quality of image details under the same seed and prompt with a NFE of 10.

Table 3. Comparison of FID score (↓) for the task of text-guided conditional generation of SD on randomly selected 30,000 MS-COCO prompts. Best results are **bolded** and the second best results are underlined.

Methods \NFEs	FID (↓) on MS-COCO-14 prompts							
	5	6	7	8	9	10	12	15
	SD-1.5							
DDIM [75]	29.14	27.11	23.48	22.36	21.14	21.22	19.91	19.36
PNDM [50]	35.50	34.49	29.50	27.86	24.46	22.35	19.13	17.92
DPM [51]	22.07	20.58	19.92	19.64	19.47	19.34	19.30	17.32
DPM++ [52]	21.75	20.14	19.60	19.32	19.19	19.13	19.11	18.03
UniPC [92]	21.72	20.30	19.84	19.67	19.55	19.40	19.41	19.29
LML **(Ours)**	**18.68**	**18.13**	**17.61**	**17.50**	**17.58**	**17.60**	**17.55**	**16.98**
	SD2-base							
DDIM [75]	27.34	24.25	22.47	21.48	19.73	20.43	18.73	18.19
PNDM [50]	31.13	30.93	26.33	25.56	22.04	20.35	17.39	16.47
DPM [51]	21.25	19.94	19.21	18.78	18.51	18.36	18.13	15.84
DPM++ [52]	20.94	19.51	18.77	18.43	18.21	18.06	17.86	15.99
UniPC [92]	20.81	19.40	18.82	18.56	18.36	18.27	18.08	18.03
LML **(Ours)**	**17.76**	**16.99**	**16.17**	**16.02**	**16.29**	**15.95**	**16.16**	**15.14**

5.3.2. FID on MS-COCO Benchmark

Following the approach in [69, 90], we randomly selected 30,000 prompts from the MS-COCO dataset [47] and generated images conditioned on these prompts. We tested our sampler in terms of FID on the SD-1.5 and SD2-base models with resolutions of 512×512 and a CFG scale 7.0. Table 3 shows that our LML sampler surpasses the baseline methods on SD models across different NFEs.

Table 4. Comparison of different samplers in text-guided image generation task on the T2I-BC benchmark [34]. The metrics are Color, Shape, and Texture. Best results are **bolded** and the second best results are underlined.

Metrics	T2I Benchmark on SD-1.5 [67]			T2I Benchmark on SD2-base [67]			T2I Benchmark on SD-XL [59]			T2I Benchmark on PixArt-α [8]		
	Color$_{(\uparrow)}$	Shape$_{(\uparrow)}$	Texture$_{(\uparrow)}$	Color$_{(\uparrow)}$	Shape$_{(\uparrow)}$	Texture$_{(\uparrow)}$	Color$_{(\uparrow)}$	Shape$_{(\uparrow)}$	Texture$_{(\uparrow)}$	Color$_{(\uparrow)}$	Shape$_{(\uparrow)}$	Texture$_{(\uparrow)}$
DDIM [75]	0.3864	0.3791	0.4231	0.5111	0.4182	0.4822	0.5670	0.4712	0.5076	0.2933	0.3746	0.4045
PNDM [50]	0.3810	0.3761	0.4331	0.5067	0.4190	0.4863	0.5682	0.4772	0.5085	**0.4035**	0.3996	0.4225
DPM [51]	0.3877	0.3945	0.4294	0.5162	0.4307	0.5002	0.5795	0.4841	0.5194	0.3263	0.3895	0.4349
DPM++ [52]	0.3881	0.3958	0.4307	0.5202	0.4326	0.5020	0.5798	0.4858	0.5201	0.3136	0.3882	0.4364
UniPC [92]	0.3892	0.3889	0.4306	0.5146	0.4337	0.4995	0.5828	0.4910	0.5234	0.3187	0.3880	0.4266
LML (**Ours**)	**0.4335**	**0.4252**	**0.4746**	**0.5640**	**0.4792**	**0.5330**	**0.5908**	**0.4938**	**0.5363**	0.3801	**0.4265**	**0.4684**

Figure 7. LML integrates seamlessly with ControlNet.

5.3.3. T2I-CB Benchmark

To further validate the enhanced sampling quality of LML in text-to-image generation, we carried out tests on diverse commercial-level DMs (SD-1.5 [67], SD2-base [67], SD-XL [59], and PixArt-α [8]) using the T2I-CB [34] benchmark. This benchmark is designed for open-world text-to-image generation evaluation. For each model, we evaluated three metrics (color, shape, and texture) to assess the visual quality of the samplers. Each assessment was based on 30,000 images with a NFE of 10. As shown in Table 4, our LML always achieves the best or second best scores on T2I-BC across these models.

5.3.4. Application on ControlNet

Our model can seamlessly integrate with existing diffusion model plugins, such as ControlNet [89]. The results in Fig 7 demonstrate that our method is fully compatible with ControlNet and generates high-quality samples.

5.4. Time-costs Comparison

Table 5 presents the average wall-clock time costs for LML across different image generation tasks. These time-cost experiments were conducted on a single NVIDIA RTX 4090 chip and averaged over 200 tests with a batchsize of 1. The results suggest that the additional computational cost incurred by our LML technique is virtually negligible, making LML as time-efficient as DDIM [75].

Table 5. Average wall-clock time costs for LML and DDIM.

Models	Methods \NFEs	Time-costs (s)						
		5	10	15	20	30	50	100
CIFAR-10	DDIM	0.13	0.20	0.27	0.34	0.52	0.81	1.49
	LML (**Ours**)	0.13	0.21	0.27	0.37	0.48	0.82	1.51
CelebA	DDIM	0.20	0.28	0.39	0.47	0.66	1.03	1.95
	LML (**Ours**)	0.20	0.31	0.38	0.48	0.67	1.10	2.08
SD-1.5	DDIM	0.45	0.73	1.03	1.31	1.87	3.01	5.85
	LML (**Ours**)	0.45	0.74	1.04	1.32	1.89	3.03	5.89
SD-2base	DDIM	0.45	0.71	1.00	1.25	1.80	2.87	5.53
	LML (**Ours**)	0.46	0.72	0.98	1.27	1.81	2.88	5.60
SD-XL	DDIM	2.32	4.25	6.23	8.12	12.06	19.82	39.48
	LML (**Ours**)	2.32	4.30	6.23	8.21	12.11	19.99	39.55
PixArt	DDIM	0.66	1.08	1.49	1.92	2.74	4.31	8.45
	LML (**Ours**)	0.67	1.08	1.51	1.92	2.75	4.28	8.49

5.5. Hyperparameters

Our LML introduces two robust hyperparameters with clear high-level interpretations: the damping coefficient λ and the mixture coefficient κ. The damping coefficient λ determines the degree to which our approximated Hessian is interpolated with the identity matrix. A larger λ value results in the dominance of the identity matrix, leading the LML to degenerate into Langevin. Conversely, a smaller λ value allows more guidance from the Hessian geometry, but a minimal value could lead to ill-conditioning issues. The κ controls the EMA rate of previous information. A small κ would make the LML close to the identity map, but LML contributes to obvious image quality enhancement in practice. More discussions on hyperparameter tuning scheme, setup, and ablation experiments are provided in Supp. B.2.

6. Conclusions

In this paper, we propose a training-free method termed Levenberg-Marquardt-Langevin (LML) for improving sampling quality, which uses a low-rank approximated damping Hessian geometry to guide the Langevin update. We provide a theoretical analysis of the approximation error for low-rank approximation, the stationary measure, and the convergence rate for damping dynamics. We conduct extensive experiments to demonstrate that our LML method can contribute to significant improvement in sampling quality, with negligible computational overhead. The code is available at https://github.com/zituitui/LML-diffusion-sampler.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62206248 and 62402430, and the Zhejiang Provincial Natural Science Foundation of China under Grant LQN25F020008. Fangyikang Wang would like to extend his gratitude to Pengze Zhang from ByteDance, Binxin Yang, and Xinhang Leng from WeChat Vision for their discussions regarding the experiments. Additionally, he is thankful to Zebang Shen from ETH Zürich and Zhichao Chen from Peking University for their insights on Langevin dynamics.

References

[1] Krizhevsky Alex. Learning multiple layers of features from tiny images. *https://www. cs. toronto. edu/kriz/learning-features-2009-TR. pdf*, 2009. 6

[2] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008. 2

[3] Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu, and Bo Zhang. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In *Proceedings of the 39th International Conference on Machine Learning*, pages 1555–1584. PMLR, 2022. 1

[4] P Bickel, P Diggle, S Fienberg, U Gather, I Olkin, and S Zeger. Springer series in statistics. *Principles and Theory for Data Mining and Machine Learning. Cham, Switzerland: Springer*, 2009. 6

[5] C Le Bris and P-L Lions. Existence and uniqueness of solutions to fokker–planck type equations with irregular coefficients. *Communications in Partial Differential Equations*, 33(7):1272–1317, 2008. 3

[6] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5): 1190–1208, 1995. 5

[7] Defang Chen, Zhenyu Zhou, Can Wang, Chunhua Shen, and Siwei Lyu. On the trajectory regularity of ode-based diffusion sampling. *arXiv preprint arXiv:2405.11326*, 2024. 1

[8] 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. *arXiv preprint arXiv:2310.00426*, 2023. 8

[9] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014. 2

[10] Wei-Ting Chen, Gurunandan Krishnan, Qiang Gao, Sy-Yen Kuo, Sizhou Ma, and Jian Wang. Dsl-fiqa: Assessing facial image quality via dual-set degradation learning and landmark-guided transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2931–2941, 2024. 7

[11] Zhichao Chen, Haoxuan Li, Fangyikang Wang, Odin Zhang, Hu Xu, Xiaoyu Jiang, Zhihuan Song, and Eric H Wang. Rethinking the diffusion models for numerical tabular data imputation from the perspective of wasserstein gradient flow. *arXiv preprint arXiv:2406.15762*, 2024. 5

[12] Sinho Chewi, Thibaut Le Gouic, Chen Lu, Tyler Maunu, Philippe Rigollet, and Austin Stromme. Exponential ergodicity of mirror-langevin diffusions. *Advances in Neural Information Processing Systems*, 33:19573–19585, 2020. 1, 6, 3

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, pages 8780–8794, 2021. 1

[14] Ying Ding. A note on quadratic transportation and divergence inequality. *Statistics & Probability Letters*, 100:115–123, 2015. 6

[15] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *Advances in Neural Information Processing Systems*, 35:30150–30166, 2022. 1, 3

[16] Ethan Eade. Gauss-newton/levenberg-marquardt optimization. *Tech. Rep.*, 2013. 3

[17] Jinyan Fan, Jianchao Huang, and Jianyu Pan. An adaptive multi-step levenberg–marquardt method. *Journal of Scientific Computing*, 78:531–548, 2019. 5

[18] Qian Feng, Hanbin Zhao, Chao Zhang, Jiahua Dong, Henghui Ding, Yu-Gang Jiang, and Hui Qian. Pectp: Parameter-efficient cross-task prompts for incremental vision transformer. *arXiv preprint arXiv:2407.03813*, 2024. 5

[19] Qian Feng, Dawei Zhou, Hanbin Zhao, Chao Zhang, and Hui Qian. Lw2g: Learning whether to grow for prompt-based continual learning. *arXiv preprint arXiv:2409.18860*, 2024. 5

[20] Andreas Fischer, Alexey F Izmailov, and Mikhail V Solodov. Unit stepsize for the newton method close to critical solutions. *Mathematical Programming*, 187(1):697–721, 2021. 5

[21] Hao Fu, Hanbin Zhao, Jiahua Dong, Chao Zhang, and Hui Qian. Iap: Improving continual learning of vision-language models via instance-aware prompting. *arXiv preprint arXiv:2503.20612*, 2025. 5

[22] Tianfan Fu, Luo Luo, and Zhihua Zhang. Quasi - newton hamiltonian monte carlo. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016. 3

[23] Yansong Gao, Zhihong Pan, Xin Zhou, Le Kang, and Pratik Chaudhari. Fast diffusion probabilistic model sampling through the lens of backward error analysis. *arXiv preprint arXiv:2304.11446*, 2023. 1

[24] Israel Gohberg, Seymour Goldberg, Marinus A Kaashoek, Israel Gohberg, Seymour Goldberg, and Marinus A Kaashoek. Hilbert-schmidt operators. *Classes of Linear Operators Vol. I*, pages 138–147, 1990. 6

[25] J Hackl, HJ Wacker, and W Zulehner. An efficient step size control for continuation methods. *BIT Numerical Mathematics*, 20:475–485, 1980. 5

[26] David Hasler and Sabine E Suesstrunk. Measuring colorfulness in natural images. In *Human vision and electronic imaging VIII*, pages 87–95. SPIE, 2003. 7

[27] Shuai He, Anlong Ming, Shuntian Zheng, Haobin Zhong, and Huadong Ma. Eat: An enhancer for aesthetics-oriented transformers. In *Proceedings of the 31st ACM international conference on multimedia*, pages 1023–1032, 2023. 7

[28] Ernst Hellinger. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271, 1909. 6

[29] 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. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 4

[30] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2

[31] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 1

[32] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012. 4

[33] Ya-Ping Hsieh, Ali Kavis, Paul Rolland, and Volkan Cevher. Mirrored langevin dynamics. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[34] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023. 8

[35] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021. 1

[36] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998. 2

[37] Mark Kac. *Enigmas of chance: an autobiography*. Univ of California Press, 1987. 2

[38] Ioannis Karatzas and Steven Shreve. *Brownian motion and stochastic calculus*. springer, 2014. 3

[39] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 4, 7

[40] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022. 5

[41] Atsushi Kawamoto. Stabilization of geometrically nonlinear topology optimization by the levenberg–marquardt method. *Structural and Multidisciplinary Optimization*, 37:429–433, 2009. 3

[42] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36: 36652–36663, 2023. 7

[43] Cheng-Yu Ku and Weichung Yeih. Dynamical newton-like methods with adaptive stepsize for solving nonlinear algebraic equations. *Computers, Materials, & Continua*, 31(3): 173–200, 2012. 5

[44] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86, 1951. 6

[45] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 1

[46] Zhenyi Liao, Qingsong Xie, Chen Chen, Hannan Lu, and Zhijie Deng. Facescore: Benchmarking and enhancing face quality in human generation. *arXiv preprint arXiv:2406.17100*, 2024. 7

[47] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 7

[48] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 5

[49] Guan-Horng Liu, Tianrong Chen, Evangelos Theodorou, and Molei Tao. Mirror diffusion models for constrained and watermarked generation. *Advances in Neural Information Processing Systems*, 36:42898–42917, 2023. 5

[50] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022. 1, 5, 6, 7, 8

[51] 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. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 1, 6, 7, 8

[52] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022. 1, 6, 7, 8

[53] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. 1

[54] James Martin, Lucas C Wilcox, Carsten Burstedde, and Omar Ghattas. A stochastic newton mcmc method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing*, 34(3): A1460–A1487, 2012. 1, 3

[55] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983. 5

[56] Yurii Nesterov et al. *Lectures on convex optimization*. Springer, 2018. 2, 1

[57] Lester SH Ngia and Jonas Sjoberg. Efficient training of neural nets for nonlinear adaptive filtering using a recursive

levenberg-marquardt algorithm. *IEEE Transactions on Signal Processing*, 48(7):1915–1927, 2000. 5

[58] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the 39th International Conference on Machine Learning*, pages 16784–16804, 2022. 1

[59] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 8

[60] Boris T Polyak. Newton's method and its use in optimization. *European Journal of Operational Research*, 181(3): 1086–1096, 2007. 3

[61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 4

[62] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[63] H Risken. The fokker-planck equation, 1996. 6

[64] Severi Rissanen, Markus Heinonen, and Arno Solin. Free hunch: Denoiser covariance estimation for diffusion models without extra costs. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 3

[65] CP Robert. Monte carlo statistical methods, 1999. 1

[66] Ralph Tyrell Rockafellar. Convex analysis:(pms-28). 2015. 3

[67] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 3, 5, 6, 7, 8

[68] Sam Roweis. Levenberg-marquardt optimization. *Notes, University Of Toronto*, 52, 1996. 1, 3

[69] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 36479–36494, 2022. 7, 4

[70] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, pages 36479–36494, 2022. 1

[71] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 7

[72] Jack Sherman and Winifred J Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1): 124–127, 1950. 4, 1

[73] Umut Simsekli, Roland Badeau, Taylan Cemgil, and Gaël Richard. Stochastic quasi-newton langevin monte carlo. In *International Conference on Machine Learning*, pages 642–651. PMLR, 2016. 1, 3

[74] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1

[75] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 1, 7, 8

[76] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 1, 2, 5

[77] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1, 4, 6

[78] Jiahang Tu, Hao Fu, Fengyu Yang, Hanbin Zhao, Chao Zhang, and Hui Qian. Texttoucher: Fine-grained text-to-touch generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7455–7463, 2025. 5

[79] Jiahang Tu, Wei Ji, Hanbin Zhao, Chao Zhang, Roger Zimmermann, and Hui Qian. Driveditfit: Fine-tuning diffusion transformers for autonomous driving data generation. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(3):1–29, 2025. 5

[80] Sergio Verdú. Total variation distance and the distribution of relative information. In *2014 information theory and applications workshop (ITA)*, pages 1–3. IEEE, 2014. 6

[81] Cédric Villani et al. *Optimal transport: old and new*. Springer, 2009. 2, 6

[82] Fangyikang Wang, Hubery Yin, Yuejiang Dong, Huminhao Zhu, Chao Zhang, Hanbin Zhao, Hui Qian, and Chen Li. Belm: Bidirectional explicit linear multi-step sampler for exact inversion in diffusion models. *arXiv preprint arXiv:2410.07273*, 2024. 5

[83] Fangyikang Wang, Huminhao Zhu, Chao Zhang, Hanbin Zhao, and Hui Qian. Gad-pvi: A general accelerated dynamic-weight particle-based variational inference framework. *Entropy*, 26(8):679, 2024. 5

[84] Fangyikang Wang, Hubery Yin, Shaobin Zhuang, Huminhao Zhu, Yinan Li, Lei Qian, Chao Zhang, Hanbin Zhao, Hui Qian, and Chen Li. Efficiently access diffusion fisher: Within the outer product span space, 2025. 1, 3, 6, 2

[85] Xiyu Wang, Anh-Dung Dinh, Daochang Liu, and Chang Xu. Boosting diffusion models with an adaptive momentum sampler. *arXiv preprint arXiv:2308.11941*, 2, 2023. 1

[86] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011. 2

[87] Mengfei Xia, Yujun Shen, Changsong Lei, Yu Zhou, Deli Zhao, Ran Yi, Wenping Wang, and Yong-Jin Liu. Towards more accurate diffusion model acceleration with a timestep tuner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5736–5745, 2024. 1

[88] Shuchen Xue, Zhaoqiang Liu, Fei Chen, Shifeng Zhang, Tianyang Hu, Enze Xie, and Zhenguo Li. Accelerating diffusion sampling with optimized time steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8292–8301, 2024. 1

[89] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 8

[90] Pengze Zhang, Hubery Yin, Chen Li, and Xiaohua Xie. Tackling the singularities at the endpoints of time intervals in diffusion models. *arXiv preprint arXiv:2403.08381*, 2024. 7

[91] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022. 1

[92] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 6, 7, 8

[93] Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. *Advances in Neural Information Processing Systems*, 36:55502–55542, 2023.

[94] Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ode-based sampling for diffusion models in around 5 steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7777–7786, 2024. 1

[95] Huminhao Zhu, Fangyikang Wang, Tianyu Ding, Qing Qu, and Zhihui Zhu. Analyzing and improving model collapse in rectified flow models. *arXiv preprint arXiv:2412.08175*, 2024. 5

[96] Huminhao Zhu, Fangyikang Wang, Chao Zhang, Hanbin Zhao, and Hui Qian. Neural sinkhorn gradient flow. *arXiv preprint arXiv:2401.14069*, 2024. 5

Unleashing High-Quality Image Generation in Diffusion Sampling Using Second-Order Levenberg-Marquardt-Langevin

Supplementary Material

A. Proofs and Detailed Formulations

A.1. Formulations of gradient descent and Newton's method

Gradient Descent (GD) [56] is a renowned optimization method for finding the minimum. Given a differentiable function f, its iterative scheme writes

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k). \tag{11}$$

Given a twice differentiable function f, the iterative scheme of Newton's method writes

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \left[\boldsymbol{H}_f(\boldsymbol{x}_k)\right]^{-1} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k). \tag{12}$$

In statistics, the *Hessian geometry* of a distribution $p(\boldsymbol{x})$ is defined to be $\nabla^2_{\boldsymbol{x}} \log p(\boldsymbol{x})$.

A.2. Proof of Proposition 1

Here, we give the proof of Proposition 1, which is an analogy of the Gauss-Newton technique in the diffusion model context.

Proof.

$$p_t\left(\boldsymbol{x}_t\right) \approx \frac{1}{\sqrt{2p}\sigma_t} \exp\left(-\frac{\left\|\boldsymbol{x}_t - \alpha_t \boldsymbol{y}_\theta(\boldsymbol{x}_t, t)\right\|^2}{2\sigma_t^2}\right), \tag{13}$$

denote $r(\boldsymbol{x}_t) = \left\|\boldsymbol{x}_t - \alpha_t \boldsymbol{y}_\theta(\boldsymbol{x}_t, t)\right\| \in \mathbb{R}$

$$\begin{aligned} -\frac{\boldsymbol{\varepsilon}_\theta}{\sigma_t} &= \nabla_{\boldsymbol{x}_t} \log p_t\left(\boldsymbol{x}_t\right) \\ &= -\frac{1}{2\sigma_t^2} \nabla_{\boldsymbol{x}_t} \left\|\boldsymbol{x}_t - \alpha_t \boldsymbol{y}_\theta(\boldsymbol{x}_t, t)\right\|^2 \\ &= -\frac{1}{\sigma_t^2} r(\boldsymbol{x}_t) \frac{\partial r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}. \end{aligned} \tag{14}$$

Rearranging Eq. 14, we can use $\boldsymbol{\varepsilon}_\theta$ to represent $\frac{\partial r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}$:

$$\frac{\partial r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t} = \frac{\sigma_t}{r(\boldsymbol{x}_t)} \boldsymbol{\varepsilon}_\theta. \tag{15}$$

We can then approximate the Jacobian matrix of $\boldsymbol{\varepsilon}_\theta$ using the Gauss-Newton technique, which essence in the omission of the second derivatives $\frac{\partial^2 r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t^2}$.

$$\begin{aligned} \frac{\partial \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{\partial \boldsymbol{x}_t} &= \frac{\partial \left(\frac{r(\boldsymbol{x}_t)}{\sigma_t} \frac{\partial r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}\right)}{\partial \boldsymbol{x}_t} \\ &= \frac{1}{\sigma_t}\left[\frac{\partial r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t} \frac{\partial r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}^T + r(\boldsymbol{x}_t) \frac{\partial^2 r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t^2}\right] \\ &\approx \frac{1}{\sigma_t} \frac{\partial r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t} \frac{\partial r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}^T \\ &= \frac{1}{\sigma_t}\left(\frac{\sigma_t}{r(\boldsymbol{x}_t)} \boldsymbol{\varepsilon}_\theta\right)\left(\frac{\sigma_t}{r(\boldsymbol{x}_t)} \boldsymbol{\varepsilon}_\theta\right)^T \\ &= \frac{\sigma_t}{\left\|\boldsymbol{x}_t - \alpha_t \boldsymbol{y}_\theta(\boldsymbol{x}_t, t)\right\|^2} \boldsymbol{\varepsilon}_\theta \boldsymbol{\varepsilon}_\theta^T \\ &= \frac{\sigma_t}{\left\|\sigma_t \boldsymbol{\varepsilon}_\theta\right\|^2} \boldsymbol{\varepsilon}_\theta \boldsymbol{\varepsilon}_\theta^T \\ &= \frac{1}{\sigma_t \left\|\boldsymbol{\varepsilon}_\theta\right\|^2} \boldsymbol{\varepsilon}_\theta \boldsymbol{\varepsilon}_\theta^T. \end{aligned} \tag{16}$$

□

A.3. Detailed Derivation of Eq. 8

The Eq. 8 is a direct result of the Sherman–Morrison formula. Here we first state this formula and then derive Eq. 8.

Theorem 1. *(Sherman–Morrison formula, [72]) Suppose $A \in \mathbb{R}^{n \times n}$ is an invertible square matrix and $u, v \in \mathbb{R}^n$ are column vectors. Then $A + uv^\top$ is invertible iff $1 + v^\top A^{-1} u \neq 0$. In this case,*

$$\left(A + uv^\top\right)^{-1} = A^{-1} - \frac{A^{-1} u v^\top A^{-1}}{1 + v^\top A^{-1} u}. \tag{17}$$

Here, $uv^\top$ is the outer product of two vectors u and v.

The Eq. 8 can be obtained by applying the Sherman–Morrison formula with $A = \lambda \boldsymbol{I}$ and $u = v = \boldsymbol{\varepsilon}_\theta$. In this case, the requirements of the Sherman–Morrison formula are satisfied, as $1 + \boldsymbol{\varepsilon}_\theta^\top \left(\lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\varepsilon}_\theta = 1 + \frac{\left\|\boldsymbol{\varepsilon}_\theta\right\|^2}{\lambda} \neq 0$.

A.4. Derivation of Eq. 9

Our way of transforming the annealing Langevin to the continuous SDE adopts a technique akin to that in [77]. Discretizing the Eq. 7 at each nose level, and gradually decreasing the noise level, we would get the following iterative schemes:

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \epsilon_i \boldsymbol{H}_{LM}^{-1}(\boldsymbol{x}_{i+1}, \lambda) \nabla_{\boldsymbol{x}} \log p\left(\boldsymbol{x}_{i+1}\right) + \sqrt{2\epsilon_i} \boldsymbol{H}_{LM}^{-1} \boldsymbol{z}_i, \tag{18}$$

Then, applying the reverse process of the ancestral sampling [76] to Eq. 18, we obtain the LM annealing SDE in Eq. 9.

A.5. Derivation of Eq. 10

Here, we will prove that the marginal distribution of Eq. 10 is the same as that of Eq. 9. We will first state the Feynman–Kac formula.

Theorem 2. *(Feynman–Kac formula, [37]) For an Ito SDE as follows*

$$\mathrm{d}\mathbf{x}_t = \boldsymbol{\mu}\left(\mathbf{x}_t, t\right)\mathrm{d}t + \boldsymbol{\sigma}\left(\mathbf{x}_t, t\right)\mathrm{d}\mathbf{B}_t, \tag{19}$$

Its underlying distribution evolves according to the following Fokker-Planck equation:

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = -\nabla \cdot [\boldsymbol{\mu} p] + \frac{1}{2}\nabla \cdot \left(\boldsymbol{\sigma}\boldsymbol{\sigma}^\top \nabla p\right) \tag{20}$$

Applying the theorem 2, the Fokker-Planck equation of Eq. 9 writes:

$$\begin{aligned}
&\frac{\partial p_t}{\partial t} \\
=& -\nabla \cdot [f_t \boldsymbol{x}_t p_t - g_t^2 \boldsymbol{H}_{LM}^{-1} \nabla_{\boldsymbol{x}_t} \log p_t \cdot p_t] + \frac{1}{2}\nabla \cdot \left(g_t^2 \boldsymbol{H}_{LM}^{-1} \nabla p_t\right) \\
=& -\nabla \cdot [f_t \boldsymbol{x}_t p_t - g_t^2 \boldsymbol{H}_{LM}^{-1} \nabla p_t] + \frac{1}{2}\nabla \cdot \left(g_t^2 \boldsymbol{H}_{LM}^{-1} \nabla p_t\right) \\
=& -\nabla \cdot [f_t \boldsymbol{x}_t p_t - \frac{1}{2} g_t^2 \boldsymbol{H}_{LM}^{-1} \nabla p_t]
\end{aligned} \tag{21}$$

Applying the theorem 2, the Fokker-Planck equation of Eq. 10 writes:

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot [f_t \boldsymbol{x}_t p_t - \frac{1}{2} g_t^2 \boldsymbol{H}_{LM}^{-1} \nabla p_t] \tag{22}$$

It is obvious to see that Eq. 9 and Eq. 10 result in the same form of Fokker-Planck equation. As they start from the same initial noise distribution, their marginal distribution will also be the same.

A.6. Proof of Proposition 2

Proof. In order to establish the boundary for the Gauss-Newton-type technique in diffusion scenarios, it is necessary to estimate the boundary of the simplified second derivatives. These second derivatives incorporate the Fisher information of the diffused distribution. We therefore utilize the analytical form as outlined in Proposition 3 of [84], leveraging their form to establish the boundary. The notation used in this proof is borrowed from their work.

$$\begin{aligned}
&\left\| r(\boldsymbol{x}_t) \frac{\partial^2 r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t^2} \right\|_{HS} \\
\leq& \|r(\boldsymbol{x}_t)\| \left\| \frac{\partial^2 r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t^2} \right\|_{HS} \\
\leq& \|\boldsymbol{x}_t - \alpha_t \boldsymbol{y}_\theta(\boldsymbol{x}_t, t)\| \left(\delta_3 + 2\sigma_t \|F_t(\boldsymbol{x}_t)\|_{HS}\right) \\
\leq& \left(\alpha_t \delta_2 + \|\boldsymbol{x}_t - \alpha_t \bar{\boldsymbol{y}}\|\right) \\
&\left(\delta_3 + 2\sigma_t^2 \left\| \frac{1}{\sigma_t^2} \boldsymbol{I} \right\|_{HS} + 2\sigma_t^2 \left\| \frac{\alpha_t^2}{\sigma_t^4} \sum_i w_i \boldsymbol{y}_i \boldsymbol{y}_i^\top \right\|_{HS} \right. \\
&\left. + 2\sigma_t^2 \left\| \frac{\alpha_t^2}{\sigma_t^4} \left(\sum_i w_i \boldsymbol{y}_i\right) \left(\sum_i w_i \boldsymbol{y}_i\right)^\top \right\|_{HS} \right) \\
\leq& \left(\alpha_t \delta_2 + \delta_1 + \alpha_t \mathcal{D}_y\right) \left(\delta_3 + 2 + 2\frac{\alpha_t^2}{\sigma_t^2} \mathcal{D}_y^2\right)
\end{aligned} \tag{23}$$

□

A.7. Proof of Proposition 3

To obtain the stationary analysis result of our LM-Langevin, we first borrow the stationary analysis of mirror Langevin from [33].

Theorem 3. *[33, Eq 3.2] If we are able to draw a sample* $\mathbf{Y}$ *from* $e^{-W(\mathbf{y})}\mathrm{d}\mathbf{y}$*, then* $\nabla h^\star(\mathbf{Y})$ *immediately gives a sample for the desired distribution* $e^{-V(\mathbf{x})}\mathrm{d}\mathbf{x}$*. Furthermore, suppose for the moment that* $\mathrm{dom}\,(h^\star) = \mathbb{R}^d$*, so that* $e^{-W(\mathbf{y})}\mathrm{d}\mathbf{y}$ *is unconstrained. Then we can simply exploit the classical Langevin Dynamics (1.1) to efficiently take samples from* $e^{-W(\mathbf{y})}\mathrm{d}\mathbf{y}$*. The above reasoning leads us to set up the Mirrored Langevin Dynamics (MLD):*

$$\begin{cases} \mathrm{d}\mathbf{Y}_t = -(\nabla W \circ \nabla h)\left(\mathbf{X}_t\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t \\ \mathbf{X}_t = \nabla h^\star\left(\mathbf{Y}_t\right) \end{cases} \tag{24}$$

Notice that the stationary distribution of $\mathbf{Y}_t$ *in MLD is* $e^{-W(\mathbf{y})}\mathrm{d}\mathbf{y}$*.*

Then the behavior of our stationary in Proposition 3 is a direct result of Theorem 2 by setting the mirror duality map ∇h as $\nabla \log p_t(\cdot) + \lambda\|\cdot\|$.

A.8. Proof of Proposition 4

Definition 1. *The* χ^2*-distance between* μ *and* π *is defined as :*

$$\chi^2(\mu\|\pi) := \mathrm{var}_\pi \frac{\mathrm{d}\mu}{\mathrm{d}\pi} = \int \left(\frac{\mathrm{d}\mu}{\mathrm{d}\pi}\right)^2 \mathrm{d}\pi - 1, \quad \textit{if} \quad \mu \ll \pi \tag{25}$$

and $\chi^2(\mu\|\pi) = \infty$ *otherwise, where* $\mu \ll \pi$ *means* μ *is absolutely continuous with respect to* π*.*

Definition 2. *(Mirror Poincaré condition, [12]). Given a mirror map ϕ, that is a strictly convex twice continuously differentiable function of Legendre type [66], we say that the distribution π satisfies a mirror Poincaré condition with constant C_{MP} if*

$$(MP) \qquad \mathrm{var}_\pi\, g \le C_{\mathrm{MP}} \mathbb{E}_\pi \left\langle \nabla g, \left(\nabla^2 \phi\right)^{-1} \nabla g \right\rangle, \quad \text{for all locally Lipschitz } g \in L^2(\pi) \tag{26}$$

When $\phi = \|\cdot\|^2/2$, (MP) is simply called a Poincaré condition and the smallest C_{MP} for which the inequality holds is the Poincaré constant of π, denoted C_{P}.

Assumption 1. *The target distribution $p(\boldsymbol{x}_t)$ satisfy the mirror Poincaré condition with $\phi = \log p(\cdot) + \lambda\|\cdot\|^2/2$ with constant C_{LMP}, we name this Levenberg-Marquardt Poincaré condition.*

Proof. The law $(\mu_t)_{t\ge 0}$ of LML in Eq. 7 satisfies the following Fokker-Planck equation in the weak sense [38, §5.7]

$$\partial_t \mu_t = \mathrm{div}\left(\mu_t \left(\nabla^2 \log p(\cdot) + \lambda \boldsymbol{I}\right)^{-1} \nabla \ln \frac{\mu_t}{p}\right) \tag{27}$$

which is well-posed with enough regularity [5, Proposition 6]. Using this, we can compute the derivative of the chi-squared divergence:

$$\begin{aligned}
&\partial_t \chi^2\left(\mu_t \| p\right) \\
&= \partial_t \int \frac{\mu_t^2}{p} \\
&= 2\int \frac{\mu_t}{p} \partial_t \mu_t \\
&= 2\int \frac{\mu_t}{p} \mathrm{div}\left(\mu_t \left(\nabla^2 \log p(\cdot) + \lambda \boldsymbol{I}\right)^{-1} \nabla \ln \frac{\mu_t}{p}\right) \\
&= -2\int \left\langle \nabla \frac{\mu_t}{p}, \left(\nabla^2 \log p(\cdot) + \lambda \boldsymbol{I}\right)^{-1} \nabla \ln \frac{\mu_t}{p} \right\rangle \mu_t \\
&= -2\int \left\langle \nabla \frac{\mu_t}{p}, \left(\nabla^2 \log p(\cdot) + \lambda \boldsymbol{I}\right)^{-1} \nabla \frac{\mu_t}{p} \right\rangle p
\end{aligned} \tag{28}$$

The Assumption 1 implies

$$\begin{aligned}
\partial_t \chi^2\left(\mu_t \| p\right) &= -2\int \left\langle \nabla \frac{\mu_t}{p}, \left(\nabla^2 \log p(\cdot) + \lambda \boldsymbol{I}\right)^{-1} \nabla \frac{\mu_t}{p} \right\rangle p \\
&\ge -2\frac{1}{C_{\mathrm{LMP}}} \chi^2\left(\mu_t \| p\right)
\end{aligned} \tag{29}$$

Applying Grönwall's inequality to Eq. 29, we can get the exponential ergodic convergence rate:

$$\chi^2\left(\mu_t \| p\right) \le e^{-\frac{2t}{C_{\mathrm{LMP}}}} \chi^2\left(\mu_0 \| p\right) \tag{30}$$

□

B. Experimental Details

B.1. Detail settings

Across all experiments, for our LML, DPM-Solver, and DPM-Solver++, we maintain the solver order at 3, and the log-SNR trajectory follows the original setup. The text prompts of Figure 4 and 5 are taken from https://medium.com/phygital/top-40-useful-prompts-for-stable-diffusion-xl-008c03dd0557.

B.2. Hyperparameters

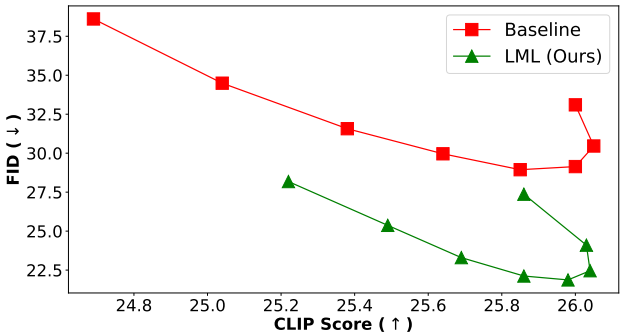
Ablations and settings on κ As illustrated in Figure 10, there is a noticeable improvement in performance as κ gradually increases. However, excessively large values of κ may be detrimental. Due to computational constraints, it is not feasible to optimize κ for all of our experiments. Consequently, we have chosen to fix $\kappa = 1 \times 10^{-8}$ for all tests in Tables 1 and 3. This is a very small value that contributes minimally to performance enhancement. There remains considerable potential for performance improvement by fine-tuning κ further in our LML sampler. For CelebA-HQ, we set λ as 0.004. And we set λ as 0.001 for SD-XL, 0.006 for PixArt-α.

Settings on λ Due to the behavior of λ as in Figure 9, we employed a binary search strategy to determine the value for the hyperparameter λ. In each iteration, we computed the performance of our model using the current λ value and then updated the range based on the results. If the performance improved, we would continue the search in the direction of the current λ; otherwise, we would search in the opposite direction. This process was repeated until we reached 5 iterations. Thus, the total tuning computation budget is controlled. We present the detailed λ setting of our experiments in Table 6 and 7.

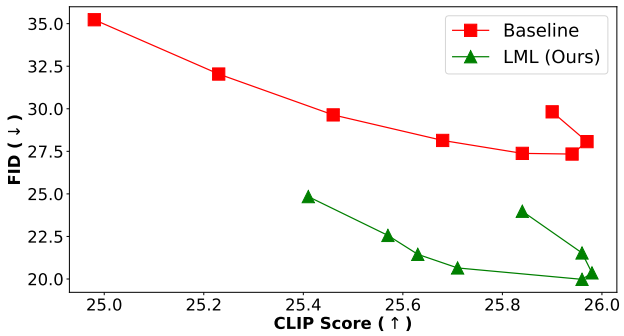
B.3. Pretrained models

All of the pretrained models used in our research are open-sourced and available online as follows:

- ddpm-ema-cifar10
  https://github.com/VainF/Diff-Pruning/releases/download/v0.0.1/ddpm_ema_cifar10.zip
- CompVis/ldm-celebahq-256
  https://huggingface.co/CompVis/ldm-celebahq-256
- stable-diffusion-v1.5
  https://huggingface.co/runwayml/stable-diffusion-v1-5
- stable-diffusion-v2-base
  https://huggingface.co/stabilityai/stable-diffusion-2-base



(a) SD-v1.5



(b) SD-v2-base

Figure 8. Comparison of Pareto curves between LML and baseline on SD-v1.5 and SD-v2-base on 30k COCO images, across various guidance scales in [5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0], using 5 NFEs.

Table 6. The λ setting in Table 1.

Methods	λ settings on CIFAR-10 generation											
	5 NFEs	6 NFEs	7 NFEs	8 NFEs	9 NFEs	10 NFEs	12 NFEs	15 NFEs	20 NFEs	30 NFEs	50 NFEs	100 NFEs
LML	0.0008	0.0008	0.001	0.001	0.001	0.0008	0.001	0.001	0.0005	0.0003	0.0001	0.00005

Table 7. The λ setting in Table 3.

Methods \NFEs	λ settings on SD models							
	5	6	7	8	9	10	12	15
	SD-1.5							
LML	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	SD2-base							
LML	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.0008

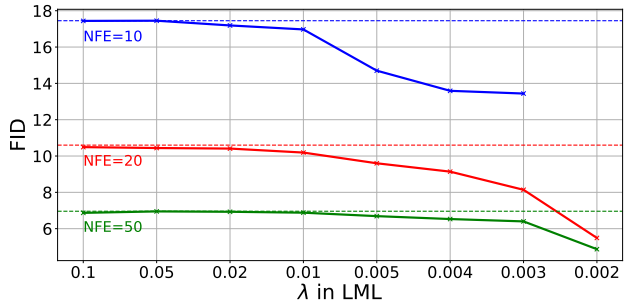


Figure 9. The performance of the LML method on CIFAR-10 generation with various choices of the damping coefficients λ. The dashed lines signify the performance of the DDIM method. For simplicity, we adopt DDIM denoise scheme here.

- stable-diffusion-XL
  https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0
- PixArt-α
  https://huggingface.co/PixArt-alpha/PixArt-XL-2-512x512

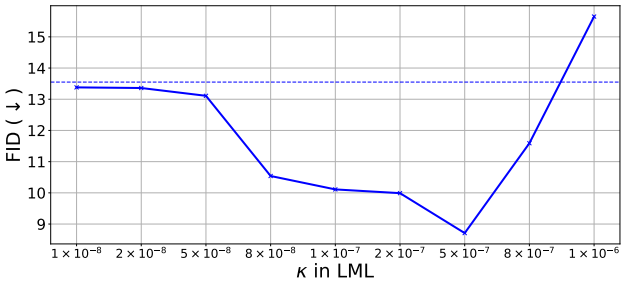


Figure 10. The performance of the LML method on CIFAR-10 generation with various choices of the damping coefficients κ. The dashed lines signify the performance when $\kappa = 0$. We fix using 10 NFEs and $\lambda = 0.003$. For simplicity, we adopt the DDIM denoise scheme here.

C. Additional Experimental Results

A visual comparison of LML and the baselines under 10 NFEs with the same seeds is provided in Figure 11, showing that our LML technique noticeably enhances detail visual realism.

C.1. CLIP v.s. FID Pareto curve experiment

Inspired by Imagen [69], we plot CLIP vs. FID Pareto curves by varying guidance values within the range [5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0] in Fig. 8. Specifically, Fréchet Inception Distance (FID) [29] calculates the Fréchet distance between the real data and the generated data. A lower FID implies more realistic generated data. While the Contrastive Language-Image Pre-training (CLIP) [61] score measures the similarity between the generated images and

the given prompts. A higher CLIP score means the generated images better match the input prompts. Our LML sampler exhibits substantial improvements over the baseline sampler. As the guidance scale increases, the LML sampler consistently maintains a lower FID compared to baseline for achieving a similar CLIP score. This emphasizes that our approach not only enhances image realism but also ensures better adherence to the input prompts.

C.2. More visual comparisons

We provide more visual quantitative comparisons of our LML sampler with baseline on CIAFR-10, as shown in Figure 11. It is shown that our LML generates samples with enhanced visual fidelity.

D. Discussions

D.1. Social Impacts

The enhanced image generation sampler proposed in this paper has substantial potential across multiple domains, including machine learning, healthcare, environmental modeling, and economics. However, while this research offers great promise for positive change, it is essential to consider potential adverse societal implications. The improved capabilities of generative models provided by LML might be misused. For example, it could be exploited to generate aesthetically enhanced deepfakes, thereby contributing to the spread of misinformation. In the healthcare sector, if not appropriately regulated, the use of synthetic patient data could give rise to ethical concerns. Thus, it is of utmost importance to ensure that the results of this research are applied ethically and responsibly, with adequate safeguards in place to prevent misuse and protect privacy.

D.2. Denoising Schemes and Timesteps

Theoretically, our LML is independent of existing denoising schemes such as DDIM, DPM-Solver, UniPC, and others. Due to restricted computational resources, we currently only integrate DDIM and DPM-Solver with our LML. However, it would be intriguing to incorporate more advanced denoising schemes and timestep selection methods to further improve the performance of our LML.

D.3. Connections to Mirror Duality

In convex optimization, the mirror mechanism [55] is a powerful framework that bridges the primary space (original variable domain) and the dual space (transformed domain via convex conjugation). The primary space hosts the optimization variables, while the dual space, derived through the Legendre-Fenchel transform, captures conjugate representations of convex functions. The Legendre map (or Legendre transform) acts as a bijection between these spaces, converting a convex function in the primary space into its dual counterpart. This duality enables leveraging geometric properties of both spaces to design efficient algorithms.

In optimization, the mirror mechanism is used to either handle a constraint problem or for acceleration. Our method can be viewed as doing a mirror diffusion model [49] in the dual space defined by a Legendre transform map of $\nabla \log p_t(\cdot) + \lambda \|\cdot\|$ to enhance sampling quality.

D.4. Limitations

In this paper, we employ a fixed λ for LML throughout all timesteps. As demonstrated in advanced LM literature [17, 57], a more effective strategy may be to adaptively control λ. Additionally, there is potential to develop a more refined rank approximation of the diffusion Hessian by following the concept of the L-BFGS-type method [6], which could enhance the accuracy of our diffusion Hessian approximation. Our technique may also enhance the generation quality of flow matching models [48, 95] or variational inference models [83, 96]. Our technique may also enhance the downstream applications of diffusion models in automated driving [79], touch-generation [78], domain-transfer [18, 19], language generation [21], exact inversion [82], and missing data imputation [11].

(a) DDIM (FID = 17.45)

(b) PNDM (FID = 7.16)

(c) DPM-Solver (FID = 11.13)

(d) DPM-Solver++ (FID = 10.71)

(e) UniPC (FID = 10.70)

(f) LML**(Ours)** (FID = **6.54**)

Figure 11. Comparison of the CIFAR-10 generation task performance between our LML method and the baseline methods under 10 NFEs. Our LML achieves a superior FID score and enhances visual realism. They are tested using the same pretrained model and seeds.