An Analysis of Regularization and Fokker–Planck Residuals in Diffusion Models for Image Generation

Onno Niemann
Universidad Autónoma de Madrid
onno.niemann@uam.es

Gonzalo Martínez Muñoz
Universidad Autónoma de Madrid
gonzalo.martinez@uam.es

Alberto Suárez Gonzalez
Universidad Autónoma de Madrid
alberto.suarez@uam.es

***Abstract*—Recent work has shown that diffusion models trained with the denoising score matching (DSM) objective often violate the Fokker–Planck (FP) equation that governs the evolution of the true data density. Directly penalizing these deviations in the objective function reduces their magnitude but introduces a significant computational overhead. It is also observed that enforcing strict adherence to the FP equation does not necessarily lead to improvements in the quality of the generated samples, as often the best results are obtained with weaker FP regularization. In this paper, we investigate whether simpler penalty terms can provide similar benefits. We empirically analyze several lightweight regularizers, study their effect on FP residuals and generation quality, and show that the benefits of FP regularization are available at substantially lower computational cost. Our code is available at https://github.com/OnnoNiemann/fp_diffusion_analysis.**

***Index Terms*—Diffusion Models, Score-Based Generative Models, Regularization, Fokker–Planck Equation**

I. Introduction

Diffusion models (DMs) have emerged as a powerful class of generative models, achieving state-of-the-art performance in image synthesis and likelihood-based evaluation [1]. Initially, DMs were formulated in discrete time, with noise added and removed over a finite sequence of steps. Two closely related frameworks shaped these developments: Score Matching with Langevin Dynamics (SMLD) [2] and Denoising Diffusion Probabilistic Models (DDPMs) [3]. SMLD leverages annealed Langevin dynamics, a process that generates samples by moving through a sequence of discrete noise levels, while following the score. DDPM uses a framework describing noise injection as a Markov chain and trains a neural network to remove the noise.

Both of these discrete-time models were later unified under a continuous-time SDE framework [1], in which the forward noising process that progressively adds noise to clean images, and the reverse, denoising process, are both described by Stochastic Differential Equations (SDEs) [4]. The forward process maps clean images to a simple, known prior distribution and the denoising process enables the generation of new images by starting from pure random noise and iteratively denoising until the generated samples converge to the distribution of the training data. This iterative noise removal requires access to the score of the data distribution, which is typically learned by training a neural network (called a noise-conditional score model) by Denoising Score Matching (DSM). Minimizing the DSM objective yields an accurate approximation of the scores of data distributions at different noise levels and enables sampling via iteratively following the reverse SDE [1].

Despite their success, recent work has uncovered a fundamental limitation of score-based generative models. The forward noising process follows a diffusion SDE and therefore the evolution of the ground truth marginal densities over noise levels is described by the Fokker–Planck (FP) equation [5]. However, the learned score approximations obtained via DSM often violate the corresponding condition that the FP equation imposes on the score [6], [7]. Explicitly enforcing this equation by including a FP-based regularization term in the training objective reduces the violation. However, empirical results show that strong FP regularization does not always yield the highest-quality samples. [6].

Moreover, computing the second-order derivatives that appear in the FP penalty is computationally demanding, and can significantly increase the training time with respect to non-regularized diffusion models [6]. On top of computational challenges, the FP penalty consists of many components and their interactions, making it difficult to attribute improvements to individual terms. This raises the question, whether the observed benefits of FP regularization require the full complexity of enforcing the FP equation, or can be achieved by leveraging more general regularization effects at lower computational expense and with more interpretable penalties.

The goal of this paper is to systematically investigate a set of such penalty terms. We evaluate their impact on FP residuals, score behavior across noise levels, and downstream sample quality. The results of this study show empirically that simpler regularizers can deliver benefits comparable to FP regularization at a substantially lower computational cost.

II. Background

A. Score-based diffusion models

The goal of generative models is to capture the structure, variability, and dependencies in the training data, making it

The authors acknowledge financial support from project PID2022-139856NB-I00 funded by MCIN/ AEI / 10.13039/501100011033 / FEDER, UE and project, IDEA-CM (TEC-2024/COM-89) from the Autonomous Community of Madrid and from the ELLIS Unit Madrid. The authors acknowledge computational support from the Centro de Computación Científica-Universidad Autónoma de Madrid (CCC-UAM).

possible to produce realistic and diverse samples that resemble the original ones. In image generation, the data available for induction are images whose distribution, $p_0(\boldsymbol{x})$, is unknown. Score-based generative models consist of a forward noise-injection and a backward denoising process. In the forward process, noise is injected into the original images according to a forward SDE

$$d\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}_t, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^D$ denotes an image with D pixels, $\boldsymbol{f}(\boldsymbol{x}, t) \in \mathbb{R}^D$ is the deterministic drift term, $g(t) \in \mathbb{R}$ is a scalar diffusion term that controls the strength of the noise injected at t, and $\boldsymbol{w}_t$ denotes a Wiener process (standard Brownian motion).

The forward process progressively injects noise, according to (1), starting from $\boldsymbol{x}(0) \sim p_0(\boldsymbol{x})$, one of the images of the training dataset. This yields a sequence of images $\{\boldsymbol{x}(t), t \in [0, T]\}$ where larger values of t correspond to higher noise levels. This sequence eventually converges to $\boldsymbol{x}(T) \sim p_T(\boldsymbol{x})$, a simple, known reference distribution. In practice, $\boldsymbol{f}(\boldsymbol{x}, t)$ and $g(t)$ are chosen so that both the reference distribution, $p_T(\boldsymbol{x})$, and the conditional distribution $p_t(\boldsymbol{x}(t) \mid \boldsymbol{x}(0))$ are Gaussian. If the forward SDE has a Gaussian transition kernel,

$$p_t(\boldsymbol{x}(t) \mid \boldsymbol{x}(0)) = \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}(0), t),\, \sigma^2(t)\boldsymbol{I}\big)\,, \tag{2}$$

where $\boldsymbol{\mu}(\boldsymbol{x}(0), t)$ and $\sigma(t)$ are determined by the choice of the functions $\boldsymbol{f}(\boldsymbol{x}, t)$ and $g(t)$, the corrupted image at time (noise level) t can be obtained in one step

$$\boldsymbol{x}(t) = \boldsymbol{\mu}(\boldsymbol{x}(0), t) + \sigma(t)\boldsymbol{z}, \quad z \sim \mathcal{N}(0, I). \tag{3}$$

Common choices of $\boldsymbol{f}(\boldsymbol{x}, t)$ and $g(t)$ are the variance-preserving (VP) and variance-exploding (VE) SDEs [1].

To generate new samples from $p_0(\boldsymbol{x})$, the distribution of the original images, the noise injection process can be reversed by solving the backward SDE [4]

$$d\boldsymbol{x} = [\boldsymbol{f}(\boldsymbol{x}, t) - g(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})]dt + g(t)d\boldsymbol{w}_t, \tag{4}$$

where $\boldsymbol{w}_t$ is standard Brownian motion going backward in time and $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ is the Stein score function, i.e. the gradient of the log-probability of the data distribution at time t. The generating process starts from an image sampled from the reference distribution at time T, $\overleftarrow{\boldsymbol{x}}(T) \sim p_T(\boldsymbol{x})$. From this sample the noise is iteratively removed by integrating the backwards SDE from T until 0, so that $\overleftarrow{\boldsymbol{x}}(0) \sim p_0(\boldsymbol{x})$. This implies that the generated image exhibits a statistical resemblance to the original ones in the training data.

In order to perform the backward process, it is necessary to estimate the term $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$. However, learning the score function directly is not possible, since the marginal $p_t(\boldsymbol{x})$ and the score are unknown. Notwithstanding, DSM provides an estimator for this score that leverages the known conditional given in (2). This allows us to approximate the score using a neural score network, $\boldsymbol{s}(\boldsymbol{x}(t), t; \boldsymbol{\theta})$. Examples of the backward and forward processes are illustrated in Fig. 1.

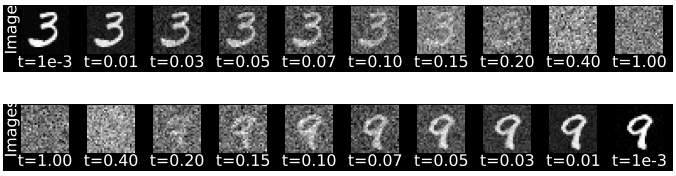

Fig. 1. Noise injection in the forward process following (1) (top) and image generation from noise according to (4) (bottom).

B. Denoising Score Matching

The goal of score matching is to train a model $s(\boldsymbol{x}(t), t; \boldsymbol{\theta})$ to approximate $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ without requiring access to the normalization constant of $p(\boldsymbol{x})$. Conceptually, this is achieved by training a neural network with entries and outputs in $\mathbb{R}^D$ to map noisy images, $\boldsymbol{x}(t)$, to the correct scores, $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$. However, as the ground truth score function is unknown in the context of images, this explicit score matching idea is replaced by Denoising Score Matching, which learns the score of noisy images conditioned on clean ones

$$\begin{aligned} L_{\text{DSM}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}(0), t, \boldsymbol{x}(t)|\boldsymbol{x}(0)} \Big[\big\| \boldsymbol{s}(\boldsymbol{x}(t), t; \boldsymbol{\theta}) \\ - \nabla_{\boldsymbol{x}(t)} \log p_t(\boldsymbol{x}(t) \mid \boldsymbol{x}(0)) \big\|_2^2 \Big]. \end{aligned} \tag{5}$$

This function can be used in practice, since we know the score of the conditional distribution (2)

$$\nabla_{\boldsymbol{x}(t)} \log p_t(\boldsymbol{x}(t) \mid \boldsymbol{x}(0)) = -\frac{\boldsymbol{x}(t) - \boldsymbol{\mu}(\boldsymbol{x}(0), t)}{\sigma(t)^2} = -\frac{\boldsymbol{z}}{\sigma(t)}, \tag{6}$$

where (3) is applied in the last step. Substituting this result into (5) and multiplying by $\sigma(t)$ for improved numerical stability gives the DSM loss used to train the score network

$$L_{\text{DSM}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}(0), t, \boldsymbol{z}} \left[\| \sigma(t)\, \boldsymbol{s}(\boldsymbol{x}(t), t; \boldsymbol{\theta}) + \boldsymbol{z} \|_2^2 \right]. \tag{7}$$

C. Fokker–Planck and Score Fokker–Planck Equation

The forward noising process defined in (1) is a diffusion process in which the evolution of the (unknown) ground truth data density over noise levels follows the FP equation [5]

$$\partial_t p_t(\boldsymbol{x}) = -\sum_{j=1}^{D} \partial_{x_j} \big(\boldsymbol{F}_j(\boldsymbol{x}, t)\, p_t(\boldsymbol{x}) \big), \tag{8}$$

where $\boldsymbol{F}(\boldsymbol{x}, t) = \boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2} g^2(t)\, \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$. However, calculating this FP equation is not feasible in the context of score-based generative models, since they do not model $p_t(\boldsymbol{x})$. For this reason, a corresponding condition on the score is derived in [6], the score FP equation

$$\begin{aligned} \partial_t \boldsymbol{s}(\boldsymbol{x}, t) = \nabla_{\boldsymbol{x}} \Big[\tfrac{1}{2} g^2(t)\, \mathrm{div}_{\boldsymbol{x}} \boldsymbol{s}(\boldsymbol{x}, t) + \tfrac{1}{2} g^2(t)\, \| \boldsymbol{s}(\boldsymbol{x}, t) \|_2^2 \\ - \langle \boldsymbol{f}(\boldsymbol{x}, t),\, \boldsymbol{s}(\boldsymbol{x}, t) \rangle - \mathrm{div}_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, t) \Big]. \end{aligned} \tag{9}$$

By defining

$$\mathcal{L}[\cdot] := \tfrac{1}{2} g^2\, \mathrm{div}_{\boldsymbol{x}}(\cdot) + \tfrac{1}{2} g^2\, \| \cdot \|_2^2 - \langle \boldsymbol{f}, \cdot \rangle - \mathrm{div}_{\boldsymbol{x}}(\boldsymbol{f}), \tag{10}$$

(9) can be written as $\partial_t \boldsymbol{s}(\boldsymbol{x}, t) = \nabla_{\boldsymbol{x}} \mathcal{L}[\boldsymbol{s}](\boldsymbol{x}, t)$.

An interesting finding of [6] is that while (9) must hold for the temporal evolution of the unknown ground truth score, score models trained using the DSM loss (7) often violate this condition. The deviation in the evolution of the approximate scores from the ones given by the FP equation can be quantified by subtracting the left and right side terms of (9)

$$\boldsymbol{\varepsilon}[\boldsymbol{s}](\boldsymbol{x}, t; \boldsymbol{\theta}) := \partial_t \boldsymbol{s}(\boldsymbol{x}, t; \boldsymbol{\theta}) - \nabla_{\boldsymbol{x}} \mathcal{L}[\boldsymbol{s}](\boldsymbol{x}, t; \boldsymbol{\theta}). \tag{11}$$

The average of the squared L2 norm of this error over the pixel dimension gives a time-dependent FP residual that measures the deviation from the score FP equation

$$r_{\mathrm{FP}}[\boldsymbol{s}](t; \boldsymbol{\theta}) := \frac{1}{D} \mathbb{E}_{\boldsymbol{x}(0), \boldsymbol{x}(t)|\boldsymbol{x}(0)} \left[\|\boldsymbol{\varepsilon}[\boldsymbol{s}](\boldsymbol{x}, t; \boldsymbol{\theta})\|_2^2 \right]. \tag{12}$$

This residual is found to be large, especially for small noise levels [6].

III. Regularization and Penalty Terms

The observation that diffusion models trained with DSM often violate the FP equation, suggests that adding a regularization term to the loss may produce models that adhere more closely to the FP equation, which should yield better score estimates. This section introduces some existing and proposed regularization/penalization terms.

A. Fokker–Planck Penalty

In [6], a regularization term based on the error given in (11) is proposed

$$P_{\mathrm{FP}} := \frac{1}{D} \mathbb{E}_{\boldsymbol{x}(0), t, \boldsymbol{z}} \left[\|\boldsymbol{\varepsilon}[\boldsymbol{s}](\boldsymbol{x}, t; \boldsymbol{\theta})\|_2 \right]. \tag{13}$$

This norm involves the computation of many terms that are both complicated to interpret and to compute. Specifically, it involves the computation of the squared norm of the gradients of all terms in (10), plus their interactions.

While this penalty is theoretically well-motivated, it is not obvious which of its components are responsible for improved empirical performance. This motivates our investigation of simpler penalties that isolate individual mechanisms, such as controlling score magnitude or smoothness, without explicitly enforcing the full score FP equation.

B. Deeper Understanding of Score and Divergence

To get a better understanding of the FP equation and the terms it involves, we will analyze in more detail the score and the divergence of the score and their role in the training dynamics of DMs. The divergence is the trace of the Jacobian of the score

$$\mathrm{div}_{\boldsymbol{x}}\, \boldsymbol{s}(\boldsymbol{x}, t; \boldsymbol{\theta}) = \sum_{i=1}^{D} \frac{\partial s_i(\boldsymbol{x}, t; \boldsymbol{\theta})}{\partial x_i} \tag{14}$$

and we know its value, if the model approximates the score very well (i.e. $L_{\mathrm{DSM}}(\boldsymbol{\theta}) \approx 0$). In such a situation the score model has learned to output the noise, so for a realization of $z \sim \mathcal{N}(0, I)$ at noise level t, the model would output $\boldsymbol{s}(\boldsymbol{x}(t), t; \boldsymbol{\theta}) \approx -\boldsymbol{z}/\sigma(t)$ and the divergence of the model would be $\mathrm{div}_{\boldsymbol{x}}\, \boldsymbol{s}(\boldsymbol{x}, t; \boldsymbol{\theta}) = \sum_{i=1}^{D} \partial s_i(\boldsymbol{x}, t; \boldsymbol{\theta})/\partial x_i \approx -D/\sigma(t)^2$, since $\boldsymbol{z} = (\boldsymbol{x}(t) - \boldsymbol{\mu}(t))/\sigma(t)$. This is verified empirically in Fig. 2 where the scores over noise levels calculated by a noise conditional score model trained on (7) in the VP framework are plotted. The third row shows $\sigma(t)\, \boldsymbol{s}(\boldsymbol{x}(t), t; \boldsymbol{\theta})$ and we see that it resembles $-z$ in the fourth row, meaning the model learned to predict the noise, as expected after a converged training. The bottom row shows the pixel-wise contributions to the divergence, mapped to the same scale: $\sigma(t)^2\, \partial s_i(\boldsymbol{x}, t; \boldsymbol{\theta})/\partial x_i$ and we see that the majority is negative and more or less close to the value of -1, except some outliers.

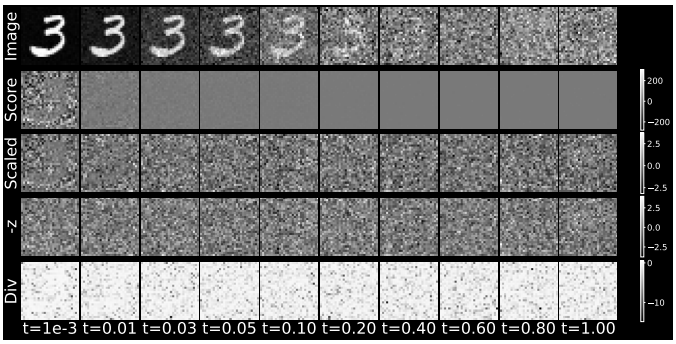

Fig. 2. Noise Injection according to (1) in the Forward Process (1st row). The learned scores are much larger for lower noise levels (2nd row), but multiplying by $\sigma(t)$ maps them to the same scale (3rd row) and we see that they resemble the noise $\boldsymbol{z}$ (4th row). The pixel-wise contribution to the divergence multiplied by $\sigma(t)^2$ is negative (5th row).

To understand how scores as noisy as the ones in Fig. 2 still point towards realistic images, we can take the average over several samples from Z and realize that there is some signal in the score and divergence. This is shown in Fig. 3. The score plots in rows 3 and 4 indicate that at higher noise levels around $t = 0.4$ the score is homogeneous and larger scores are assigned to the immediate surroundings of the digits than to the digit pixels themselves. At lower noise levels the score gets more granular and sometimes differs greatly between neighboring pixels. This is in line with the accepted understanding that in generation, DMs focus on the coarse structure first and then on the details [8]. At early stages of the generation, the model indicates that a homogeneous intensity increase of the pixels in the middle leads to higher likelihood. Since the digit pixels are already of higher intensity, they appear darker. At lower noise levels the model focuses on fine details and we see accentuated individual pixels rather than patches.

Visualizing the divergence can provide insights into how DMs learn to denoise at different stages of the process. The bottom rows of Fig. 3 show the pixel-wise contributions to the global divergence (14), $\partial s_i(\boldsymbol{x}, t; \boldsymbol{\theta})/\partial x_i$. The interpretation of these pixel-wise divergences is directly related to what the model considers likely/realistic changes in intensity: if we were to increase the intensity of one pixel, keeping all others the same, how would that change the score? In other words, a

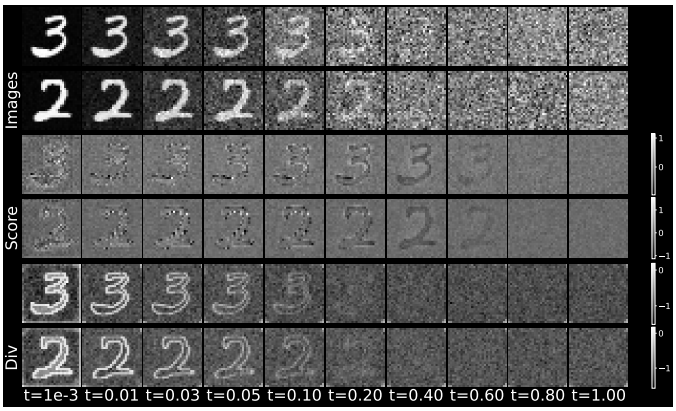

Fig. 3. The first two rows show noise injection according to (1). Rows 3 & 4, and 5 & 6 are the average of the score and the divergence, respectively, calculated over 200 realizations of the noise Z.

divergence of -1 indicates that the score would counteract the intensity increase, while a divergence of 0 means the score of pixel i does not react to an increase in intensity of that pixel.

Fig. 3 shows that at high noise levels, when there is little information in the images, the divergence contains almost no information. Around $t = 0.4$, we see some lighter patches in the middle of the frame, indicating that the score model knows that higher pixel intensities are more typical in the center than on the edges of MNIST images. At $t = 0.2$ and $t = 0.15$ we start to see the first vague outlines of the digits. In the next frames, the edges of the figures light up. The fact that the edges light up and the interior does not, indicates that the score model is less confident about intensities on the edges, while knowing that pixel neighborhoods on the interior are more similar and thus counteracting an increase in intensity more strongly.

C. Penalizing the squared norm of the score

The first proposed penalty is the squared norm of the score

$$P_{SN} = \frac{1}{D}\,\mathbb{E}_{\boldsymbol{x}(0),t,\boldsymbol{z}}\,\|\boldsymbol{s}(\boldsymbol{x},t;\boldsymbol{\theta})\|_2^2\,. \tag{15}$$

An important advantage of this penalty is that the computational cost of training the model is basically the same as the vanilla DSM loss, as the norm of the score is already computed in (7). A second advantage is that this term has a clear interpretation: it simply penalizes large score values in any pixel. As shown in Fig. 2, row 2, the magnitude of the score is significantly larger at small noise levels. As this is the part of the process where also the largest deviations from the FP equation have been observed, we hope to reduce the FP residual by penalizing extreme score approximations at low t.

D. Penalizing the squared Frobenius norm of the Jacobian of the score

Another proposed penalty with a similar motivation is the Frobenius norm of the Jacobian of the score. The Jacobian is the matrix of all pixel-wise partial derivatives and describes the effects of any change in the input images on the score model output. The Frobenius norm of this matrix is used as a measure of network complexity in [9] and is observed to increase drastically at low noise levels.

$$P_{JAC} = \frac{1}{D}\,\mathbb{E}_{\boldsymbol{x}(0),t,\boldsymbol{z}}\,\left\|\nabla_{\boldsymbol{x}(\boldsymbol{t})}\,\boldsymbol{s}(\boldsymbol{x}(t),t;\boldsymbol{\theta})\right\|_F^2 \tag{16}$$

As high score model complexity and high FP residuals coincide at small noise levels, we hope to improve the FP residual by explicitly penalizing model complexity.

E. Penalizing the divergence of the score

The final proposed penalty is the divergence of the score

$$P_{DIV} = \frac{1}{D}\,\mathbb{E}_{\boldsymbol{x}(0),t,\boldsymbol{z}}\left(\mathrm{div}_{\boldsymbol{x}}\,\boldsymbol{s}(\boldsymbol{x},t;\boldsymbol{\theta})\right)^2. \tag{17}$$

The divergence is the trace of the Jacobian and we expect a similar effect to the above penalty (16), but at lower computational cost. On top of that, the bottom row in Fig. 2 clearly shows great pixel-wise differences in the divergence. While most pixels are bright and only slightly negative, some have values below -10 and thus differ significantly from the value of -1, expected for exact score matching (see section on understanding div for motivation).

F. Efficient calculation of Jacobian and divergence

To efficiently compute divergences in P_{FP} and P_{DIV} and to compute the Jacobian of the score, we employ Hutchinson's stochastic trace estimator [10]. The divergence is the trace of the Jacobian and can be approximated as follows

$$\mathrm{div}_{\boldsymbol{x}}\,\boldsymbol{s}(\boldsymbol{x},t) \approx \frac{1}{K}\sum_{k=1}^{K} v_k\big(\nabla_{\boldsymbol{x}}\boldsymbol{s}(\boldsymbol{x},t)\big)v_k^\top, \tag{18}$$

where $v_k \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})$. Similarly, the whole Jacobian can be estimated by

$$\left\|\nabla_{\boldsymbol{x}}\boldsymbol{s}(\boldsymbol{x},t)\right\|_F^2 \approx \frac{1}{K}\sum_{k=1}^{K}\left\|v_k^\top\left(\nabla_{\boldsymbol{x}}\boldsymbol{s}(\boldsymbol{x},t)\right)\right\|_2^2, \tag{19}$$

where $v_k \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})$. In both approximations we set $K = 1$, following [6] and [1].

IV. Experiments

In order to analyze the proposed penalties, multiple experiments were carried out and the results compared to the baseline DSM model. All experiments are conducted on the MNIST dataset, which is composed of 70,000 grayscale images of 28×28 pixels, of which 60,000 images were used for training the models.

The neural networks used in the experiments follow a U-Net architecture with the same configuration as the one defined in [1]. The networks are trained using the DSM loss in (7) plus a weighted penalty term

$$L = L_{\text{DSM}} + \lambda\,P, \tag{20}$$

where P denotes each one of the proposed penalty terms and λ controls the strength of the regularization. The baseline model is trained using (7) without any penalty.

The models are trained in the variance-preserving SDE framework for 200 epochs with a learning rate of 5×10^{-4}, except for the FP penalty, for which the learning rate is 1×10^{-3}, following [6].

To evaluate the quality of the generated images, the following four metrics are used: (1) the Fréchet Inception Distance (FID) as a measure of overall perceptual quality; (2) Density, which measures how close the generated images are to the real ones in the space of extracted features [11]; (3) Coverage, which measures the mode coverage of the generated images with respect to the training set [11]; and (4) the Shannon entropy of the class distribution as an additional measure of diversity. For every trained model, 10,000 samples are generated and compared with the MNIST test set. The computation of the FID score requires extracting and comparing features from real and generated images. As MNIST differs from the training data of the Inception network, we use a pre-trained LeNet classifier for feature extraction. The Shannon entropy is computed on the class distribution given by the LeNet classifier over the generated images.

To evaluate these metrics, 20 training runs were performed for all models. The average performance on these four metrics and the training time per epoch is reported in Table I. Training was conducted on a machine with an NVIDIA GeForce RTX 5070 GPU and an AMD Ryzen AI 9 365 CPU, using Pytorch 2.9.

All tested regularizers improve the baseline performance on average in all four metrics. The FP penalty performs best with an average FID of 16.25 compared to 20.32 without regularization, but also the Jacobian and the squared norm penalty get close to this performance, with 17.02 and 17.50 on average, respectively. In terms of training times and computational expenses, the squared norm stands out. With 6.36 s/epoch it takes only marginally longer than the baseline with 6.25 s/epoch. The FP penalty, on the other hand, takes about twice as long at 12.32 s/epoch. This difference is expected to be more severe with more complex datasets, as was found by [6].

To more closely examine the behavior of models trained with different penalties, the score and the divergence at different stages of the noising process are plotted in Figs. 4 and 5, respectively. Although all models show similar behavior at high noise levels, there are striking differences at low ones. All regularization terms seem to suppress the attention to detail that the score maps of the baseline show at $t = 1e-3$ and $t = 1e-4$. Instead, the regularized score maps look more homogeneous, similar to higher noise levels.

In the divergence evolution in Fig. 5 these differences are even more extreme. The Jacobian and the divergence penalty seem to make the model completely indifferent to intensity changes of individual pixels.

The strong effect of regularization at small noise levels can also be observed in the plots of the DSM loss (Fig.6) and the Frobenius norm of the Jacobian (Fig. 8). As reported in [9] and [12], both increase exponentially at low noise levels without regularization. Compared to the baseline all penalties

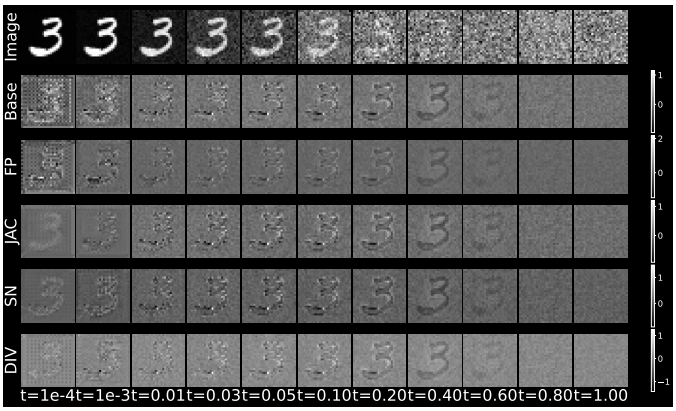

Fig. 4. Score depending on penalty term

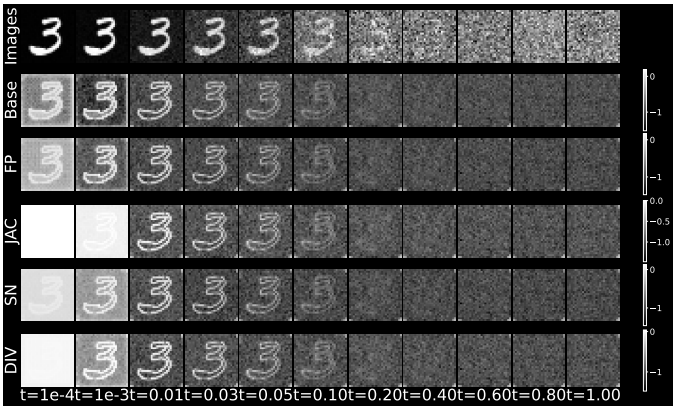

Fig. 5. Divergence depending on penalty term

accelerate the increase in L_{DSM} at low noise, while reducing the Frobenius norm of the Jacobian. The strongest effect in both plots is observed for the penalty on the Frobenius norm of the Jacobian itself.

With regards to the FP residual we observe that it is difficult to achieve a strong reduction. Fig. 7 indicates that all penalty terms reduce the FP residual, especially at small noise levels, but all types of regularization do not make a big difference. In Fig. 9 we see that aiming for a closer adherence to the FP equation by setting a higher λ results in an even worse FP residual. This counterintuitive observation strengthens the hypothesis that the benefits of FP regularization stem from general regularization effects rather than from specifically enforcing the FP equation.

V. Conclusion

We studied the role of regularization in DMs from the perspective of the Fokker–Planck equation. Enforcing the score FP equation via an explicit penalty reduces FP residuals and improves the quality of the generated images, but so do all other tested penalties. The counterintuitive finding that stronger FP regularization produces worse FP residuals than moderate regularization is consistent with the idea that the benefits of FP regularization stem from general regularization

TABLE I

COMPARISON OF PENALTY TERMS ON MNIST. P_{SN} AND P_{DIV} IMPROVE PERFORMANCE WITH MINIMAL COMPUTATIONAL OVERHEAD.

Loss	FID (↓)	Density (↑)	Coverage (↑)	Entropy (↑)	Time/epoch (s) (↓)
L_{DSM}	20.32 ± 11.41	80.10 ± 5.20	85.69 ± 1.89	2.263 ± 0.023	6.25
$+P_{FP}$	16.25 ± 7.16	82.85 ± 3.71	87.97 ± 1.67	2.269 ± 0.017	12.32
$+P_{SN}$	17.50 ± 7.89	80.44 ± 4.28	87.11 ± 1.59	2.265 ± 0.022	6.36
$+P_{JAC}$	17.02 ± 7.51	81.64 ± 4.94	87.29 ± 1.75	2.268 ± 0.0189	19.86
$+P_{DIV}$	18.46 ± 8.88	80.25 ± 4.57	86.61 ± 1.74	2.264 ± 0.023	7.34

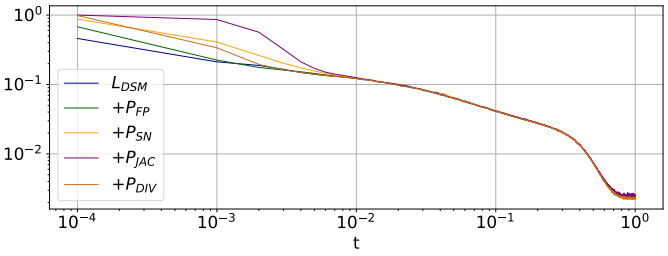

Fig. 6. The DSM loss over noise levels, evaluated for models trained with different penalty terms.

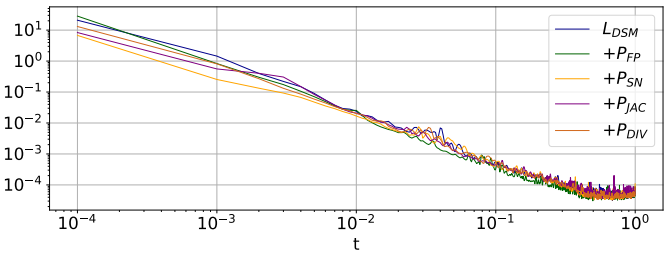

Fig. 7. The FP residual over noise levels, evaluated for models trained with different penalty terms.

effects rather than specifically enforcing the FP equation. Our experiments show that regularization most strongly affects small noise levels, which is sensible, as that is where the penalized quantities, network capacity, score magnitude and FP deviation, are largest.

We conclude that simple penalty terms can have similar benefits to the complex FP penalty, one of them at negligible computational cost. Our analysis is limited to MNIST and validating the observations on other datasets will be important.

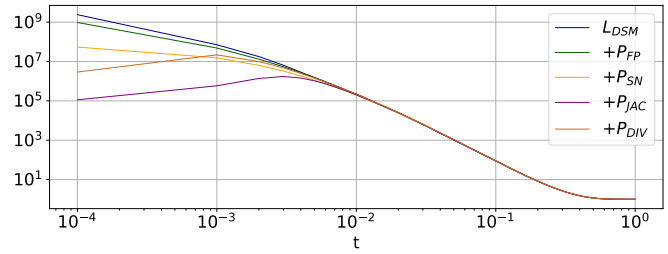

Fig. 8. The Frobenius norm of the Jacobian, evaluated for models trained with different penalty terms.

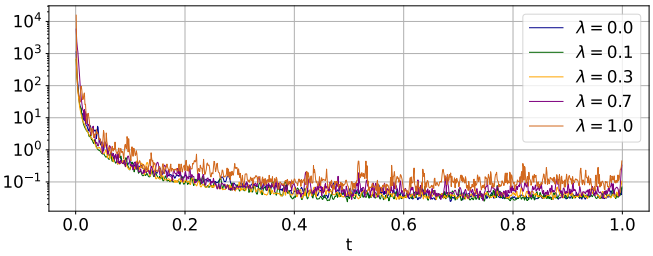

Fig. 9. FP Residual reduction depending on regularization strength

We expect that this will strengthen our claims, as MNIST is relatively simple and the computational overhead of regularization is expected to increase with dataset complexity.

REFERENCES

[1] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[2] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 11 895–11 907.

[3] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, 2020.

[4] B. D. Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982.

[5] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, 6th ed., ser. Universitext. Springer, 2003.

[6] C.-H. Lai, Y. Takida, N. Murata, T. Uesaka, Y. Mitsufuji, and S. Ermon, "FP-diffusion: Improving score-based diffusion models by enforcing the underlying score Fokker-Planck equation," in *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, 2023.

[7] M. Plainer, H. Wu, L. Klein, S. Günnemann, and F. Noé, "Consistent sampling and simulation: Molecular dynamics with energy-based diffusion models," in *Advances in Neural Information Processing Systems*, 2025.

[8] Y. Park, M. Kwon, J. Choi, J. Jo, and Y. Uh, "Understanding the latent space of diffusion models through the lens of riemannian geometry," in *Advances in Neural Information Processing Systems 36*, 2023.

[9] T. Dockhorn, A. Vahdat, and K. Kreis, "Score-based generative modeling with critically-damped langevin diffusion," in *International Conference on Learning Representations (ICLR)*, 2022.

[10] M. F. Hutchinson, "A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines," *Communications in Statistics – Simulation and Computation*, vol. 18, no. 3, pp. 1059–1076, 1989.

[11] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo, "Reliable fidelity and diversity metrics for generative models," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 2020, pp. 7176–7185.

[12] V. D. Bortoli, M. Hutchinson, P. Wirnsberger, and A. Doucet, "Target score matching," 2024.