
Efficiently Access Diffusion Fisher: Within the Outer Product Span Space

Fangyikang Wang^{1*‡} Hubery Yin^{2*} Shaobin Zhuang^{3‡} Huminhao Zhu¹ Yinan Li¹ Lei Qian¹
Chao Zhang¹ Hanbin Zhao¹ Hui Qian¹ Chen Li²

Abstract

Recent Diffusion models (DMs) advancements have explored incorporating the second-order diffusion Fisher information (DF), defined as the negative Hessian of log density, into various downstream tasks and theoretical analysis. However, current practices typically approximate the diffusion Fisher by applying auto-differentiation to the learned score network. This black-box method, though straightforward, lacks any accuracy guarantee and is time-consuming. In this paper, we show that the diffusion Fisher actually resides within a space spanned by the outer products of score and initial data. Based on the outer-product structure, we develop two efficient approximation algorithms to access the trace and matrix-vector multiplication of DF, respectively. These algorithms bypass the auto-differentiation operations with time-efficient vector-product calculations. Furthermore, we establish the approximation error bounds for the proposed algorithms. Experiments in likelihood evaluation and adjoint optimization demonstrate the superior accuracy and reduced computational cost of our proposed algorithms. Additionally, based on the novel outer-product formulation of DF, we design the first numerical verification experiment for the optimal transport property of the general PF-ODE deduced map.

1. Introduction

The emerging diffusion models (DMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019; Song et al., 2020), generating samples of data distribution from initial noise by learning a reverse diffusion process, have been proven to be an effective technique for modeling data distri-

bution, especially in generating high-quality images (Nichol et al., 2022; Dhariwal & Nichol, 2021a; Saharia et al., 2022; Ramesh et al., 2022; Rombach et al., 2022; Ho et al., 2022). The training process of DMs can be seen as employing a neural network to match the first-order diffusion score $\nabla_{\mathbf{x}} \log q_t(\mathbf{x})$ at varying noise levels.

There has been a growing trend to recognize the importance of the diffusion Fisher (DF) in DMs, defined as the negative Hessian of the diffused distributions' log density, $-\nabla_{\mathbf{x}}^2 \log q_t(\mathbf{x})$. The DF provides valuable second-order information on DMs and plays a crucial role in downstream tasks, e.g., likelihood evaluation (Lu et al., 2022a; Zheng et al., 2023), adjoint optimization (Pan et al., 2023a;b; Blasingame & Liu, 2024), and the optimal transport analysis of DMs (Zhang et al., 2024a). It is straightforward to access DFs by applying auto-differentiation to the learned first-order diffusion score. Nevertheless, the repeated auto-differentiation operations would be time-consuming for large-scale tasks, even when incorporating the Vector-Jacobian-product (VJP) and Hutchinson's estimation technique (Song & Lai, 2024). Recently, certain research efforts investigated the time-evolving property of DM and demonstrated that the DF can be explicitly expressed in terms of the score and its covariance matrix (Benton et al., 2024; Lu et al., 2022a). However, these results predominantly concentrate on the theoretical aspects and fail to enable efficient access to the DF.

In this paper, we derive a novel form for the diffusion Fisher, which is composed of weighted outer-product sums. This innovative formulation uncovers that the diffusion Fisher invariably lies within the span of a set of outer-product bases. Notably, these bases solely depend on the initial data and the noise schedule. Leveraging this outer-product structure, we devise two efficient approximation algorithms to access the trace and matrix-vector multiplication of DF, respectively. Moreover, we design the first numerical verification experiment to investigate the optimal transport property of the diffusion-ODE deduced map under diverse types of initial data and noise schedules. The main contributions of our paper are listed as follows:

- We derive an analytical formulation of the diffusion Fisher, which is the weighted summation of the outer

¹Zhejiang University. ²WeChat Vision, Tencent Inc. ³Shanghai Jiao Tong University. *Equal contribution. ‡Work done as interns at WeChat Vision, Tencent Inc. Correspondence to: Chao Zhang <zczju@zju.edu.cn>.

product, under the setting when the initial distribution is a sum of Dirac. The derivation process involves applying the consecutive partial differential chain rule to the data-dependent form of marginal distributions. Subsequently, we extend this formulation to a more general setting. In this general case, we only assume that the initial distribution has a finite second moment. These novel formulations reveal that the diffusion Fisher invariably lies within the span of a set of outer-product bases.

- Based on the outer-product structure of DF, we propose two efficient approximation algorithms, each tailored to access the trace and matrix-vector multiplication of the DF, respectively. When it comes to evaluating the trace of the diffusion Fisher, we introduce a parameterized network to learn the trace, significantly reducing the time complexity of trace evaluation from quadratic to linear w.r.t. the dimension. In situations where we need to access the matrix-vector multiplication of the diffusion Fisher, we present a training-free method that simplifies the complex linear transformation into several simple vector-product calculations. Furthermore, we rigorously establish the approximation error bounds for these two algorithms.
- Utilizing the novel outer-product formulation of DF, we derive a corollary for the optimal transport (OT) property of the diffusion-ODE deduced map. Subsequently, we design the first numerical experiment to examine the OT property of any general diffusion ODE deduced map based on the corollary. Our numerical experiments indicate that for all commonly employed noise schedules (VE (Ho et al., 2020), VP (Song & Ermon, 2019), sub-VP (Song et al., 2020), and EDM (Karras et al., 2022)), the diffusion ODE map exhibits the OT property when the initial data follows a single Gaussian distribution or has an affine form. Conversely, it does not exhibit this property when the initial data is non-affine. These numerical results not only validate the conclusions presented in (Khurlov et al., 2023; Zhang et al., 2024a; Lavenant & Santambrogio, 2022), but also open up new avenues for the exploration of more general cases.

We evaluate our DF access algorithms on likelihood evaluation and adjoint optimization tasks. The empirical results demonstrate our DF methods' enhanced accuracy and reduced time cost.

2. Preliminaries

Notation. The Euclidean norm over $\mathbb{R}^d$ is denoted by $\|\cdot\|$, and the Euclidean inner product is denoted by $\langle \cdot, \cdot \rangle$. Throughout, we simply write $\int g$ to denote the integral with respect

to the Lebesgue measure: $\int g(x)dx$. When the integral is with respect to a different measure μ , we explicitly write $\int g d\mu$. When clear from context, we sometimes abuse notation by identifying a measure μ with its Lebesgue density. We also use $\delta(\cdot)$ to denote the Dirac Delta function. For a vector $\mathbf{v} \in \mathbb{R}^d$, we denote the $d \times d$ matrix $\mathbf{v}\mathbf{v}^\top$ as the outer-product of $\mathbf{v}$.

2.1. Diffusion Models and Diffusion SDEs

Suppose that we have a d -dimensional random variable $\mathbf{x}_0 \in \mathbb{R}^d$ following an unknown target distribution $q_0(\mathbf{x}_0)$. Diffusion Models (DMs) define a forward process $\{\mathbf{x}_t\}_{t \in [0, T]}$ with $T > 0$ starting with $\mathbf{x}_0$, such that the distribution of $\mathbf{x}_t$ conditioned on $\mathbf{x}_0$ satisfies

$$q_{t|0}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha(t)\mathbf{x}_0, \sigma^2(t)\mathbf{I}), \quad (1)$$

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. According to (Kingma et al., 2021; Karras et al., 2022), with some assumption on $\alpha(\cdot)$ and $\sigma(\cdot)$, the forward process can be modeled as a linear SDE which is also called the Ornstein–Uhlenbeck process:

$$d\mathbf{x}_t = f(t)\mathbf{x}_t dt + g(t)d\mathbf{B}_t, \quad (2)$$

where $\mathbf{B}_t$ is the standard d -dimensional Brownian Motion (BM), $f(t) = \frac{d \log \alpha_t}{dt}$ and $g^2(t) = \frac{d\sigma_t^2}{dt} - 2 \frac{d \log \alpha_t}{dt} \sigma_t^2$. Under some regularity conditions, the above forward SDE equation 2 have a reverse SDE from time T to 0, which starts from $\mathbf{x}_t$ (Anderson, 1982):

$$d\mathbf{x}_t = [f(t)\mathbf{x}_t - g^2(t)\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t, t)] dt + g(t)d\tilde{\mathbf{B}}_t, \quad (3)$$

where $\tilde{\mathbf{B}}_t$ is the reverse-time Brownian motion and $q(\mathbf{x}_t, t)$ is the single-time marginal distribution of the forward process. In practice, DMs (Ho et al., 2020; Song et al., 2020) use $\epsilon_\theta(\mathbf{x}_t, t)$ to estimate $-\sigma(t)\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t, t)$ and the parameter θ is optimized by the following loss:

$$\mathcal{L} = \mathbb{E}_t \left\{ \lambda_t \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t} [\|s_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t, t|\mathbf{x}_0, 0)\|^2] \right\}, \quad (4)$$

where s_θ represents the parameterized score, i.e., $s_\theta(\mathbf{x}_t, t) = -\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sigma_t}$. This familiar parameterization is called ϵ -prediction. There are also $\mathbf{y}$ -prediction and $\mathbf{v}$ -prediction (Salimans & Ho, 2022). The corresponding loss is equal to replace the term $|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)|$ with $\frac{\alpha_t}{\sigma_t}|\mathbf{x}_0 - \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)|$ and $|\alpha_t \epsilon - \sigma_t \mathbf{x}_0 - \mathbf{v}_\theta(\mathbf{x}_t, t)|$. The learned $\epsilon_\theta(\mathbf{x}_t, t)$ can be also transformed to a $\mathbf{y}$ -prediction form by $\bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sigma_t \epsilon_\theta(\mathbf{x}_t, t)}{\alpha_t}$.

2.2. Diffusion Models Inference as Neural ODE

It is noted that the reverse diffusion SDE in equation 3 has an associated probability flow ODE (PF-ODE, also called diffu-

sion ODE), which is a deterministic process that shares the same single-time marginal distribution (Song et al., 2020):

$$d\mathbf{x}_t = \left[f(t)\mathbf{x}_t - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t, t) \right] dt. \quad (5)$$

By replacing the score function in equation 5 with the noise predictor ϵ_θ , the inference process of DMs can be constructed by the following neural ODE:

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{h}_\theta(\mathbf{x}_t, t) := f(t)\mathbf{x}_t + \frac{g^2(t)}{2\sigma_t} \epsilon_\theta(\mathbf{x}_t, t), \quad (6)$$

where initial $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \mathbf{I})$.

2.3. Fisher Information in Diffusion Models

The diffusion Fisher information matrix in DMs is defined as the negative Hessian of the marginal diffused log-density function, which takes the following matrix-valued form (Song et al., 2021; Song & Lai, 2024):

$$\mathbf{F}_t(\mathbf{x}_t, t) := -\frac{\partial^2}{\partial \mathbf{x}_t^2} \log q_t(\mathbf{x}_t, t) \quad (7)$$

The current technique typically approximately accesses the diffusion Fisher by accessing the scaled Jacobian matrix of the learned score estimator network ϵ_θ :

$$\begin{aligned} \mathbf{F}_t(\mathbf{x}_t, t) &= -\frac{\partial}{\partial \mathbf{x}_t} \left(\frac{\partial}{\partial \mathbf{x}_t} \log p(\mathbf{x}_t, t) \right) \\ &\approx -\frac{\partial}{\partial \mathbf{x}_t} \left(-\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sigma_t} \right) = \frac{1}{\sigma_t} \frac{\partial \epsilon_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \end{aligned} \quad (8)$$

The full diffusion Fisher matrix within DMs cannot be obtained due to dimensional constraints. For instance, the Stable Diffusion-1.5 model (Rombach et al., 2022) features a latent dimension of $d = 4 \times 64 \times 64 = 16384$, resulting in a diffusion Fisher matrix of 16384×16384 . Fortunately, for applications that only need to access the trace or multiplication of diffusion Fisher, it is feasible to use the Vector-Jacobian-product (VJP) to access diffusion Fisher. For any d -dimensional vector $\mathbf{v}$, the approximation of $\mathbf{v}$ left multiplied by $\mathbf{F}_t(\mathbf{x}_t, t)$ using VJP is as follows:

$$\begin{aligned} (\text{VJP}) \quad \mathbf{v}^\top \mathbf{F}_t(\mathbf{x}_t, t) &\approx \frac{1}{\sigma_t} \mathbf{v}^\top \frac{\partial \epsilon_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \\ &= \frac{1}{\sigma_t} \frac{\partial [\langle \epsilon_\theta(\mathbf{x}_t, t) | \mathbf{v} \rangle]}{\partial \mathbf{x}_t} \end{aligned} \quad (9)$$

The VJP is a time-consuming process due to its requirement for gradient calculations within the neural network. In addition, empirical evidence from synthetic distributions, as demonstrated in Lu et al. (2022a), shows that the approximation results from the VJP significantly deviate from the true underlying diffusion Fisher. To our knowledge, there

Access	Methods	Theoretical Time-cost	Practical Time-cost (s)	Theoretical Error Bound
Trace	VJP (eq. 15)	$c_1 d + c_2 d^2$	2195.48	✗
	DF-TM (Ours)	$2c_1 d$	0.072 _{99% ↓}	✓ (Prop. 7)
Operator	VJP (eq. 19)	$c_1 d + c_2 d$	0.155	✗
	DF-EA (Ours)	$2c_1 d$	0.063 _{59% ↓}	✓ (Prop. 8)

Table 1: Comparison of Vector-Jacobian-product (VJP) and our proposed efficient access methods in terms of per-iteration theoretical time-cost, practical time-cost, and approximation error bound. The theoretical time cost is based on assumptions of a network access cost of $c_1 d$ time and backpropagation on network cost of $c_2 d$ time ($c_2 \approx 4c_1$). The practical time-cost is tested using the SD-v1.5 model over 10k COCO prompts. Here, our VJP baseline calculates every element of the trace, and discussion on its approximation is deferred to Appendix C.5.

is no theoretical guarantee that the diffusion Fisher can be accurately accessed through the VJP. Moreover, the VJP fails to provide any theoretical insight into the diffusion Fisher.

3. Diffusion Fisher in Outer-Products Span

Accessing diffusion Fisher via the VJP as shown in equation 9 is straightforward, but it does not take advantage of any inherent structure of the diffusion Fisher. Though certain recent research efforts (Lu et al., 2022a; Benton et al., 2024) demonstrated that the DF can be explicitly expressed in terms of the score and its covariance matrix. Nevertheless, their formulations cannot directly facilitate efficient access to the diffusion Fisher. In this section, we derive a novel form for the diffusion Fisher, which is composed of weighted outer-product sums. We first conduct this under a simplified scenario, assuming that the initial distribution q_0 is a sum of Dirac distributions. Subsequently, we generalize this outer-product span form to a more general setting. Significantly, the outer-product span form of the diffusion Fisher obtained in both settings does not require any gradient calculations and depends solely on the initial data and the noise schedule. Notably, there exist fast methods for accessing the trace and matrix-vector multiplication of an outer-product matrix. This advantageous characteristic allows for the development of novel algorithms for efficiently accessing the diffusion Fisher. We start with a simple setting where we assume that the initial distribution is characterized as a sum of Dirac distributions composed of the set of samples in the dataset. If we suppose the dataset is denoted as $\{\mathbf{y}_i\}_{i=0}^N$, then the initial distribution follows

$$(\text{Dirac Setting}) \quad q(\mathbf{x}, t)|_{t=0} = \frac{1}{N} \sum_{i=0}^N \delta(\mathbf{x} - \mathbf{y}_i), \quad (10)$$

where exists a $0 < \mathcal{D}_y < \infty$ such that $\|\mathbf{y}_i\| \leq \mathcal{D}_y$ holds true for every i . In this Dirac setting, we derive the following formulation of diffusion Fisher, which is a weighted outer-product sum devoid of gradients and composed solely of the

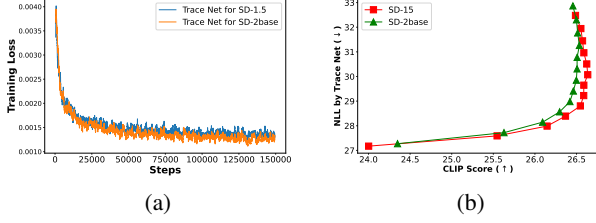


Figure 1: (a) The training loss of DF-TM for SD-1.5 and SD-2base. It demonstrates commendable convergence behavior. (b) The trade-off curve of NLL and Clip score of SD-1.5 and SD-2base across various guidance scales in [1.5, 2.5, ..., 12.5, 13.5]

initial distribution and the noise schedule.

Proposition 1. Defines $v_i(\mathbf{x}_t, t)$ as $\exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}_i|^2}{2\sigma_t^2}\right) \in \mathbb{R}$ and $w_i(\mathbf{x}_t, t)$ as $\frac{v_i(\mathbf{x}_t, t)}{\sum_j v_j(\mathbf{x}_t, t)} \in \mathbb{R}$. If q_0 takes the form as in equation 10, the diffusion Fisher matrix of the diffused distribution q_t for $t \in (0, 1]$ can be analytically formulated as follows:

$$\mathbf{F}_t(\mathbf{x}_t, t) = \frac{1}{\sigma_t^2} \mathbf{I} - \frac{\alpha_t^2}{\sigma_t^4} \left[\sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top - \left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top \right] \quad (11)$$

where we have simplified $w_i(\mathbf{x}_t, t)$ to w_i , as it does not lead to any confusion.

We also find that the $\sum_i w_i \mathbf{y}_i$ component in equation 11 can be effectively approximated by the trained score network in the form of y -prediction, as demonstrated in the following proposition.

Proposition 2. Given the diffusion training loss in equation 4, and if q_0 conforms to the form presented in equation 10, then the optimal $\bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)$ can accurately estimate $\sum_i w_i \mathbf{y}_i$.

The General Setting We further extend the diffusion Fisher formulation in equation 11 to a more general setting, where we only assume that the initial distribution q_0 is a probability measure on $\mathbb{R}^d$ with finite second moments.

$$(\text{General Setting}) \quad q_0 \in \mathcal{P}_2(\mathbb{R}^d), \quad (12)$$

which means that $q_0 \in \mathcal{P}(\mathbb{R}^d)$, $\int_{\mathbb{R}^d} \|\mathbf{x}\|^2 q_0(\mathbf{x}) d\mathbf{x} < \infty$. In this general setting, we derive the following form of diffusion Fisher, which is a weighted outer-product integral, which resides in a space spanned by an infinite set of outer products.

Proposition 3. Let us define $v(\mathbf{x}_t, t, \mathbf{y})$ as $\exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}|^2}{2\sigma_t^2}\right) \in \mathbb{R}$ and $w(\mathbf{x}_t, t, \mathbf{y})$ as $\frac{v(\mathbf{x}_t, t, \mathbf{y})}{\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}) d q_0(\mathbf{y})} \in \mathbb{R}$. If q_0 takes the form as in equation 12, the diffusion Fisher matrix of the diffused distribution q_t for $t \in (0, 1]$ can be analytically formulated as follows:

$$\mathbf{F}_t(\mathbf{x}_t, t) = \frac{1}{\sigma_t^2} \mathbf{I} - \frac{\alpha_t^2}{\sigma_t^4} \left[\int w(\mathbf{y}) \mathbf{y} \mathbf{y}^\top d q_0 - \left(\int w(\mathbf{y}) \mathbf{y} d q_0 \right) \left(\int w(\mathbf{y}) \mathbf{y} d q_0 \right)^\top \right] \quad (13)$$

where we simply write $w(\mathbf{x}_t, t, \mathbf{y})$ as $w(\mathbf{y})$, as long as it does not lead to any confusion.

We further ascertain that the $\int w(\mathbf{y}) \mathbf{y} d q_0(\mathbf{y})$ component in equation 13 can be effectively approximated by the score network in the form of y -prediction, as demonstrated in the following proposition.

Proposition 4. Given the diffusion loss in equation 4, and if q_0 conforms to the form in equation 12, then the optimal $\bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)$ can accurately estimate $\int w(\mathbf{y}) \mathbf{y} d q_0(\mathbf{y})$.

The derivation of the outer-product form of diffusion Fisher under the general setting is akin to the Dirac case, but in an integral form. For the remainder of the paper, we will focus on developing our method based on the Dirac setting diffusion Fisher. However, the same results can be naturally extended to the general setting.

4. Diffusion Fisher Trace Matching

The likelihood evaluation of DMs would require access to diffusion Fisher’s trace. In this section, we introduce a network to learn the trace, thus facilitating effective likelihood evaluation in DMs.

Log-Likelihood in DMs Log-likelihood is a classic and significant metric for probabilistic generative models, extensively utilized for comparison between samples or models (Bengio et al., 2013; Theis et al., 2015). According to Chen et al. (2018); Song et al. (2021), the log-likelihood of samples generated by PF-ODE in equation 6 from DMs can be computed through a connection to continuous normalizing flows as follows:

$$\begin{aligned} \frac{\partial \log q_t(\mathbf{x}_t, t)}{\partial t} &= -\text{tr} \left(\frac{\partial}{\partial \mathbf{x}_t} \left(f(t) \mathbf{x}_t - \frac{1}{2} g^2(t) \partial_{\mathbf{x}_t} \log q_t(\mathbf{x}_t, t) \right) \right) \\ &= -\text{tr} \left(\left(f(t) \mathbf{I} - \frac{1}{2} g^2(t) \frac{\partial^2}{\partial \mathbf{x}_t^2} \log q_t(\mathbf{x}_t, t) \right) \right) \\ &= -f(t)d - \frac{g^2(t)}{2} \text{tr}(\mathbf{F}_t(\mathbf{x}_t, t)) \end{aligned} \quad (14)$$

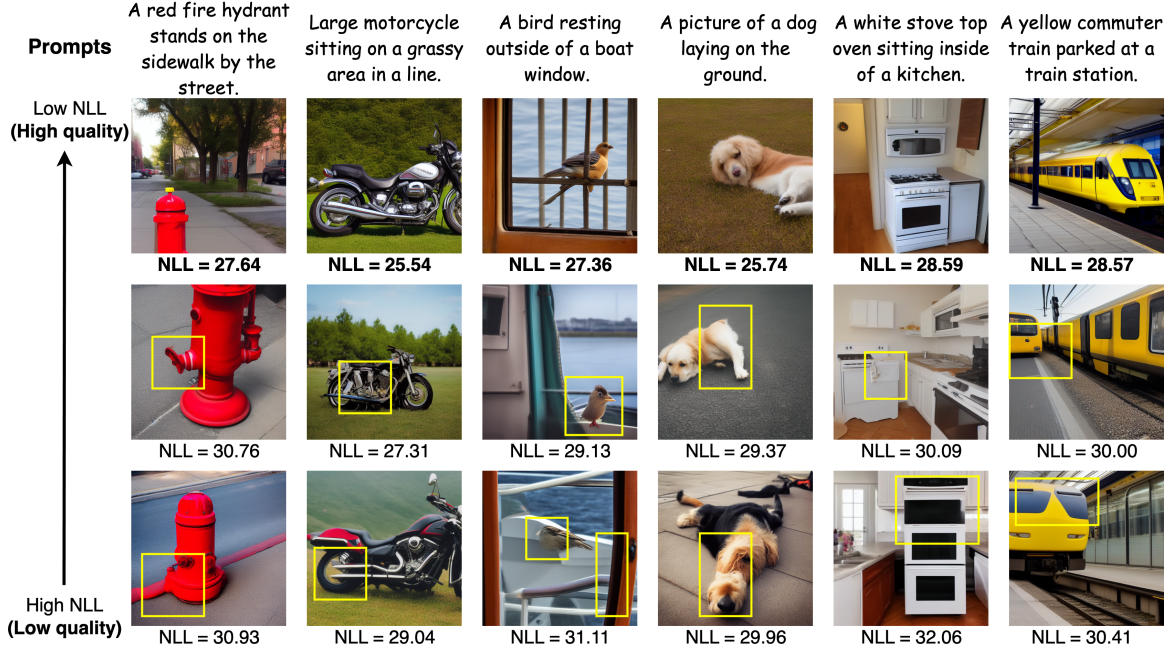


Figure 2: Our DF-TM method facilitates the effective evaluation of the NLL of generated samples with varying seeds. It can be demonstrated that a lower NLL signifies a region of higher possibility, thereby consistently indicating superior image quality.

where $\text{tr}(\cdot)$ denotes the trace of a matrix, which is defined to be the sum of elements on the diagonal.

Log-Likelihood Evaluation via VJP The current technique is only capable of conducting backpropagation of scalar value to the neural network. Therefore, the VJP in equation 9 cannot directly calculate the trace of the diffusion Fisher. The VJP must iterate through each dimension to compute the individual elements on the diagonal, and then sum them up as follows

$$\text{tr}(\mathbf{F}_t(\mathbf{x}_t, t)) \approx \frac{1}{\sigma_t^2} \sum_{i=1}^d \frac{\partial [\langle \varepsilon_\theta(\mathbf{x}_t, t) | e^{(i)} \rangle]}{\partial \mathbf{x}_t}. \quad (15)$$

Evaluating the trace using the VJP method would be extremely time-consuming due to the curse of dimensionality. If the time-complexity of a single backpropagation is $\mathcal{O}(d)$, then the calculation in equation 15 would have a time-complexity of $\mathcal{O}(d^2)$. In practice, as demonstrated in Table 1, evaluating the trace of diffusion Fisher on the SD-v1.5 model would require half an hour, rendering it nearly infeasible. Some Monte-Carlo type approximations can be used to accelerate the NLL evaluation of models, but they still fall short in per-sample NLL evaluation, see a detailed discussion in C.5.

Log-Likelihood Evaluation via DF trace matching To overcome the limitations of the VJP method in evaluating

Methods	The relative error of NLL evaluation					
	t = 1.0	t = 0.8	t = 0.6	t = 0.4	t = 0.2	t = 0.0
VJP (eq. 15)	6.68%	5.79%	10.46%	20.13%	51.14%	70.95%
DF-TM (Ours)	3.41%	4.56%	4.13%	4.28%	5.33%	5.81%

Table 2: Comparison of the VJP method and our DF-TM in terms of the diffusion Fisher trace evaluation error across different timesteps. The error is evaluated on the 2-D chessboard data with the VE schedule.

the trace of the diffusion Fisher, we propose to directly obtain its analytical form and train a network to match it. Note that the trace of an outer-product of the vector results in the vector’s norm. Thus, given the diffusion Fisher in Proposition 1, we can also derive its trace in the form of a weighted norm sum, as highlighted in the following proposition:

Proposition 5. In the same context as Proposition 1, the trace of the diffusion Fisher matrix for the diffused distribution q_t , where $t \in (0, 1]$, is given by:

$$\text{tr}(\mathbf{F}_t(\mathbf{x}_t, t)) = \frac{d}{\sigma_t^2} - \frac{\alpha_t^2}{\sigma_t^4} \left[\sum_i w_i \|\mathbf{y}_i\|^2 - \left\| \sum_i w_i \mathbf{y}_i \right\|^2 \right] \quad (16)$$

As demonstrated in Proposition 2, the $\|\sum_i w_i \mathbf{y}_i\|^2$ can be directly estimated by $\|\bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)\|^2$. Therefore, the only

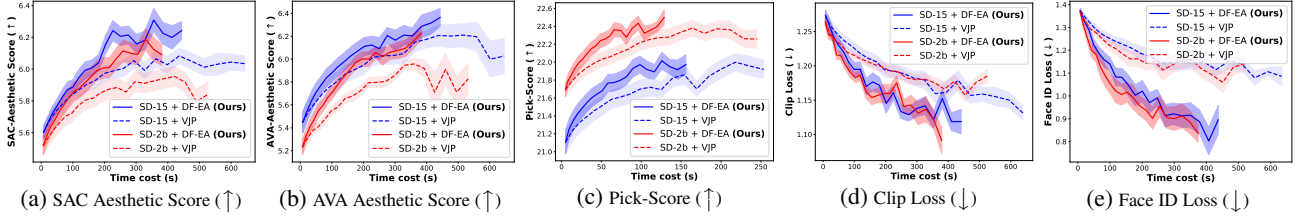


Figure 3: Comparison between our DF method and the VJP method on adjoint guidance sampling across five objective scores: SAC/AVA aesthetic score, Pick-Score, clip loss, and Face ID loss. Notably, our DF consistently achieves superior scores with less time expenditure.

unknown element in equation 16 is $\sum_i w_i \|y_i\|^2$. Consequently, we suggest learning this term using a scalar-valued network, as per the following training scheme:

Algorithm 1 Training of DF-TM Network

- 1: **Input:** data space dimension d , initial network $t_\theta(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R} \mapsto \mathbb{R}$, noise schedule $\{\alpha_t\}$ and $\{\sigma_t\}$.
- 2: **repeat**
- 3: $x_0 \sim q_0(x_0)$
- 4: $t \sim \text{Uniform}(\{1, \dots, T\})$
- 5: $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 6: $x_t = \alpha_t x_0 + \sigma_t \varepsilon$
- 7: Take gradient descent on $\nabla_\theta \left| t_\theta(x_t, t) - \frac{\|x_0\|^2}{d} \right|^2$
- 8: **until** converged
- 9: **Output:** $t_\theta(\cdot, \cdot)$

The training scheme detailed in Algorithm 1 can indeed enable $t_\theta(x_t, t)$ to estimate the weighted norm term $\frac{1}{d} \sum_i w_i(x_t, t) \|y_i\|^2$. This is substantiated by the convergence analysis Proposition 6, as presented below.

Proposition 6. $\forall (x_t, t) \in \mathbb{R}^d \times \mathbb{R}_{\geq 0}$, the optimal $t_\theta(x_t, t)$ s trained by the objective in Algorithm 1 are equal to $\frac{1}{d} \sum_i w_i(x_t, t) \|y_i\|^2$.

Once we have obtained t_θ , we can evaluate the trace of diffusion Fisher efficiently, as illustrated below. This approach is a straightforward result of equation 16 and Propositions 2 and 6, which we refer to as DF trace matching (DF-TM).

$$\text{tr}(F_t(x_t, t)) \approx \frac{d}{\sigma_t^2} - \frac{\alpha_t^2}{\sigma_t^4} \left(d * t_\theta(x_t, t) - \|y_\theta(x_t, t)\|^2 \right) \quad (17)$$

To estimate the trace using DF-TM in equation 17, we simply need one access to t_θ and ε_θ . DF-TM enables us to effectively evaluate the log-likelihood in a gradient-free manner with linear time complexity. We can further substantiate the theoretical approximation error bound when using DF-TM to calculate the trace of the diffusion Fisher, as illustrated in the Proposition 7.

Proposition 7. Assume the approximation error on $t_\theta(x_t, t)$ is δ_1 and on $\varepsilon_\theta(x_t, t)$ is δ_2 , then the approximation error of the approximated Fisher trace in equation 17 is at most $\frac{\alpha_t^2}{\sigma_t^4} \delta_1 + \frac{1}{\sigma_t^2} \delta_2^2$.

Experiments We conduct toy experiments to show the accuracy and commercial-level experiments to show the efficiency of our DF-TM method. In toy experiments, as shown in Table 2, we trained a simple DF-TM network on 2-D chessboard data and then evaluated the relative error of the trace estimated by VJP and DF-TM. It is shown that our DF-TM method consistently achieves more accurate trace estimation of DF compared to the VJP method.

In commercial-level experiments, we trained two DF-TM networks for the SD-1.5 and SD-2base pipeline on the Laion2B-en dataset (Schuhmann et al., 2022), which contains 2.32 billion text-image pairs. Our t_θ follows the U-net structure, similar to stable diffusion models, but with an added MLP head to produce a scalar-valued output. We utilize the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of 1e-4. The training is executed across 8 V100 chips with a batch size of 384 and completed after 150K steps. In Figure 1a, we demonstrate that the training loss of DF-TM nets converges smoothly, indicating the robustness of the DF-TM training scheme in Algorithm 1.

In Figure 1b, we evaluate the average NLL and Clip score of samples generated by various SD models, using 10k randomly selected prompts from the COCO dataset (Lin et al., 2014). A lower NLL suggests more realistic data generation, while a higher Clip score indicates a better match between the generated images and the input prompts. The results imply that the NLL and the Clip score form a trade-off curve across different guidance scales. This phenomenon, previously hypothesized in theory (Wu et al., 2024), is now confirmed in the SD models, thanks to the effective NLL evaluation via the DF-TM method.

In Figure 2, we display images with varying NLL under the same prompt with 10 steps on SD-1.5 using DDIM. It's clear that images with lower NLL exhibit greater visual realism, while those with higher NLL often contain deformed elements (emphasized by the yellow rectangle). Our proposed

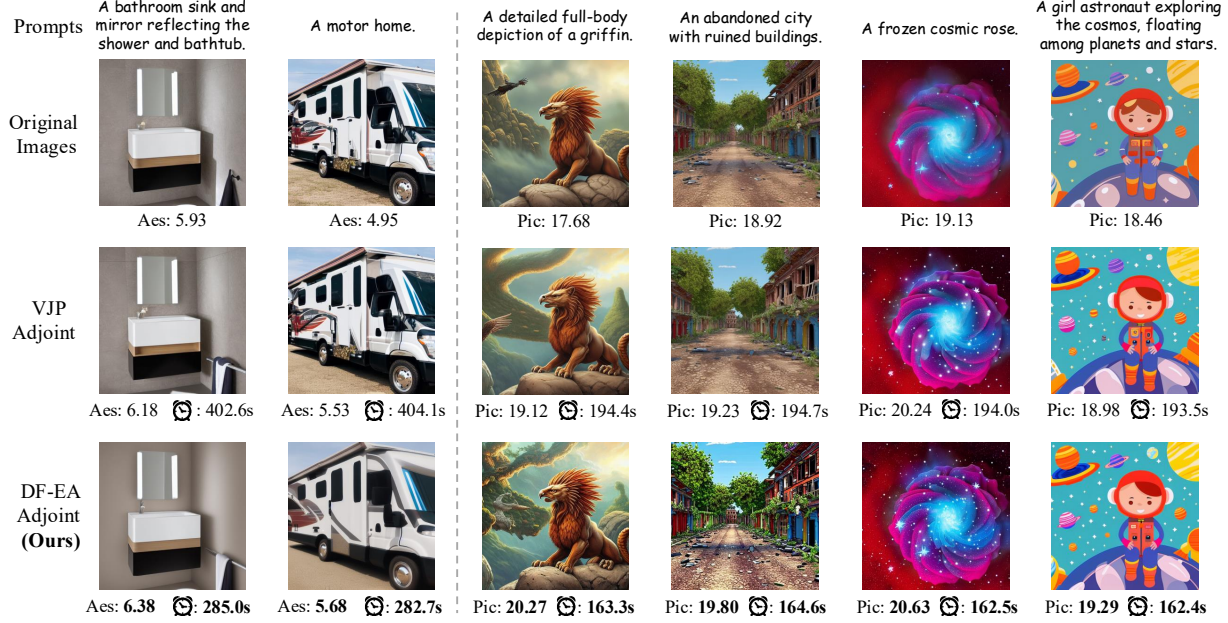


Figure 4: Visual comparison of DF-EA (**Ours**) and VJP in the adjoint improvement task on (left) SAC aesthetic score and (right) Pick-Score. DF-EA consistently generates images with better visual effects and reduced time expenditure.

NLL evaluation method proves to be an effective tool for automatic sample selection.

5. Diffusion Fisher Endpoint Approximation

The adjoint optimization of DMs would require access to the matrix-vector multiplication of the diffusion Fisher, which can also be seen as applying diffusion Fisher as a linear operator. In this section, we present a training-free method that simplifies the complex linear transformation calculations, thus enabling faster and more accurate adjoint optimization based on the outer-product form diffusion Fisher.

Adjoint optimization sampling. Guided sampling techniques are extensively utilized in diffusion models to facilitate controllable generation. Recently, to address the inflexibility of commonly used classifier-based guidance (Dhariwal & Nichol, 2021b) and classifier-free guidance (Ho & Salimans, 2022), a series of adjoint guidance methods have been explored (Pan et al., 2023a;b).

Consider optimizing a scalar-valued loss function $\mathcal{L}(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}$, which takes $\mathbf{x}_0$ in the data space as input. Adjoint guidance is implemented by applying gradient descent on $\mathbf{x}_t$ in the direction of $\frac{\partial \mathcal{L}(\mathbf{x}_0(\mathbf{x}_t))}{\partial \mathbf{x}_t}$. The essence of adjoint guidance is to use the gradient at $t = 0$ and follow the adjoint ODE (Pollini et al., 2018; Chen et al., 2018) to compute $\lambda_t := \frac{\partial \mathcal{L}(\mathbf{x}_0(\mathbf{x}_t))}{\partial \mathbf{x}_t}$ for $t > 0$.

$$\frac{d\lambda_t}{dt} = -\lambda_t^\top \frac{\partial \mathbf{h}_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}_t}, \quad \lambda_0 = \frac{\partial \mathcal{L}(\mathbf{x}_0)}{\partial \mathbf{x}_0} \quad (18)$$

Adjoint ODE via VJP. Regardless of the ODE solver being used, it is necessary to compute the right-hand side of equation 18, or equivalently, $\mathbf{F}(\mathbf{x}_t, t)^\top \lambda_t$. This computation can be interpreted as applying the diffusion Fisher matrix as a linear operator to the adjoint state λ_t , from a functional analysis perspective (Yosida, 2012). Current practices utilize the VJP technique to approximate this linear transformation operation as follows:

$$\begin{aligned} \mathbf{F}(\mathbf{x}_t, t)^\top \lambda_t &\approx \frac{1}{\sigma_t} \frac{\partial \varepsilon_\theta(\mathbf{x}_t, t)^\top}{\partial \mathbf{x}_t} \lambda_t \\ &\approx \frac{1}{\sigma_t} \frac{\partial [\langle \varepsilon_\theta(\mathbf{x}_t, t) | \lambda_t \rangle]}{\partial \mathbf{x}_t} \end{aligned} \quad (19)$$

This process involves computationally expensive neural network auto-differentiations, and the approximation errors introduced by the VJP technique have no theoretical bound.

Adjoint ODE via DF-EA. As previously discussed in Section 3, the DF inherently doesn't require gradients, suggesting that we could potentially apply the diffusion Fisher as a linear operator in a gradient-free manner. The challenging part in equation 11 is $\sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top$, which represents a weighted form of outer-products of data. Based on the definition of w_i , the closest $\mathbf{y}_i$ to $\mathbf{x}_0$ will dominate as $t \rightarrow 0$. This makes it intuitive to replace this sum with a single final sample outer-product $\mathbf{x}_0 \mathbf{x}_0^\top$. It's also important to note that the adjoint guidance itself needs to compute $\mathbf{x}_0$ at each guidance step, eliminating the need for additional computation to obtain $\mathbf{x}_0$. Given that we utilize the endpoint sample $\mathbf{x}_0$, we refer to this approximation technique as DF Endpoint

Approximation (EA). The formulation for DF-EA in the adjoint ODE is as follows:

$$\begin{aligned}
& \mathbf{F}(\mathbf{x}_t, t)^\top \boldsymbol{\lambda}_t \\
& \approx \left(\frac{1}{\sigma_t^2} \mathbf{I} - \frac{\alpha_t^2}{\sigma_t^4} \left(\sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top - \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)^\top \right) \right)^\top \boldsymbol{\lambda}_t \\
& \approx \left(\frac{1}{\sigma_t^2} \mathbf{I} - \frac{\alpha_t^2}{\sigma_t^4} \left(\mathbf{x}_0 \mathbf{x}_0^\top - \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)^\top \right) \right)^\top \boldsymbol{\lambda}_t \\
& = \frac{1}{\sigma_t^2} \boldsymbol{\lambda}_t - \frac{\alpha_t^2}{\sigma_t^4} \langle \mathbf{x}_0, \boldsymbol{\lambda}_t \rangle \mathbf{x}_0 + \frac{\alpha_t^2}{\sigma_t^4} \langle \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t), \boldsymbol{\lambda}_t \rangle \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)
\end{aligned} \tag{20}$$

The DF-EA approximation leads to a scalar-weighted combination of $\boldsymbol{\lambda}_t$, $\mathbf{x}_0$, and $\bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)$, which importantly, does not involve any gradients. Additionally, we derive the theoretical approximation error bound of the DF-EA in Proposition 8. To measure the accuracy of DF-EA as a linear operator, we opt to use the Hilbert–Schmidt norm (Gohberg et al., 1990) for measurement, as follows:

Proposition 8. Assume that the approximation error on $\varepsilon_\theta(\mathbf{x}_t, t)$ is δ_2 , the approximation error of the DF-EA linear operator, as referenced in 20, is at most $\frac{\alpha_t^2}{\sigma_t^4} (2\mathcal{D}_y^2 + \sqrt{d}\delta_2)$ when measured in terms of the Hilbert–Schmidt norm.

Experiments on DF-EA. As depicted in Figure 3, we conducted experiments comparing our DF-EA and VJP methods in adjoint guidance sampling, using five different scores and two different base models. DF-EA consistently achieves better scores due to its bounded approximation error. Furthermore, DF-EA requires less processing time as it eliminates the need for time-consuming gradient operations. DF-EA and VJP are compared under the same guidance scales and schemes across various numbers of steps. Details regarding the score function can be found in Appendix B.2.

As depicted in Figure 4, our DF-EA consistently generates samples with higher scores with a reduced time cost compared to VJP. All samples are generated within 50 steps, with adjoint applied from the 15th to the 35th step. Details on hyperparameters can be found in Appendix B.2.

6. Numerical OT Verification of DMs

There is an increasing trend towards analyzing the probability modeling capabilities of DMs by interpreting them from an optimal transport (OT) perspective (Albergo et al., 2023; Chen et al., 2024). The foundational concepts of optimal transport can be found in Appendix A.9. One of the central questions is whether the map deduced by the PF-ODE could represent an optimal transport. Khrulkov et al. (2023) have proven that, given single-Gaussian initial data and a VE noise schedule, the PF-ODE deduced map is

optimal transport. Zhang et al. (2024a) has demonstrated that affine initial data suffices for the PF-ODE map to be optimal transport. However, the OT property of the general PF-ODE map remains an open question.

In this section, we propose the first numerical OT verification experiment for a general PF-ODE deduced map based on the outer-product form diffusion Fisher we obtained in section 3. We first derive the following corollary for the OT property of the PF-ODE deduced map.

Corollary 1. Denote the diffeomorphism deduced by the PF-ODE in equation 5 as follows

$$T_{s,t} : \mathbb{R}^n \longrightarrow \mathbb{R}^n; \mathbf{x}_s \longmapsto \mathbf{x}_t, \quad \forall t \geq s > 0. \tag{21}$$

The diffeomorphism $T_{s,t}$ is a Monge optimal transport map **if and only if** the normalized fundamental matrix for $\mathbf{B}(t) \equiv \mathbf{B}(t, \mathbf{x}_t)$ at s is s.p.d. for every PF-ODE chain that starts from a $\mathbf{x}_T \in \mathbb{R}^d$, where

$$\begin{aligned}
\mathbf{B}(t, \mathbf{x}_t) = & \left[f(t) - \frac{g^2(t)}{2\sigma_t^2} \right] \mathbf{I} + \frac{\alpha_t^2 g^2(t)}{2\sigma_t^4} \left[\sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top \right. \\
& \left. - \left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top \right].
\end{aligned} \tag{22}$$

The definition of the normalized fundamental matrix is deferred to Appendix A.10.

Note that $T_{s,t}$ is well-posed, guaranteed by the global version of the Picard-Lindelöf theorem (Amann, 2011; Zhang et al., 2024a). Detailed proofs can be found in Appendix A.10. We then design the numerical OT verification experiment for a given noise schedule and initial data as shown in the Algorithm 2. The detailed version of Algorithm 2 can be found in Appendix B.3.

Algorithm 2 Numerical OT test for PF-ODE map

- 1: **Input:** initial data $\{\mathbf{y}_i\}_{i=1}^N$, noise schedule $\{\alpha_t\}$ and $\{\sigma_t\}$, discretization steps M .
- 2: Initialize $\mathbf{A}_M = \mathbf{I}$, $\mathbf{x}_M \sim \mathcal{N}(0, \sigma_T \mathbf{I})$.
- 3: **for** $i = M, M-1, \dots, 1$ **do**
- 4: $dt = t_{i-1} - t_i$.
- 5: Calculate $\mathbf{B}_i$ by equation 22.
- 6: $\mathbf{A}_{i-1} = \mathbf{A}_i + dt * \mathbf{A}_i^\top \mathbf{B}_i$ {solve fundamental matrix.}
- 7: $\mathbf{x}_{i-1} = \text{PF-ODE Solver}(\mathbf{x}_i, i)$
- 8: **end for**
- 9: **Output:** $\mathbf{A}_0$. {The result fundamental matrix.}

In Table 3, we numerically tested the OT property of the PF-ODE map under different noise schedules and initial data. We tested four commonly used noise schedules, VE (Ho et al., 2020), VP (Song & Ermon, 2019), sub-VP (Song et al., 2020), and EDM (Karras et al., 2022) on 2-D data. We first tested when the initial data is single-Gaussian and affine, we found that the result normalized fundamental

Initial Data Noise Schedule	Single-Gaussian		Affine		Non-affine	
	Asym.	OT	Asym.	OT	Asym.	OT
VE (Song & Ermon, 2019)	0.00%	✓	0.00%	✓	25.28%	✗
VP (Ho et al., 2020)	0.00%	✓	0.00%	✓	23.36%	✗
sub-VP (Song et al., 2020)	0.00%	✓	0.00%	✓	13.84%	✗
EDM (Karras et al., 2022)	0.00%	✓	0.00%	✓	27.09%	✗

Table 3: Comparison of numerical OT verification results of four commonly used noise schedulers with different initial data.

matrix is p.s.d., which means the PF-ODE map is optimal transport. These results coincide with the previous proposed theoretical results in (Khrulkov et al., 2023) and (Zhang et al., 2024a). We further tested an extremely simple non-affine case, that is, there are only three initial data points, (0,0), (0,0.5), and (0.5,0). We showed that all noise schedules result in an obvious asymmetric fundamental matrix, which means that the PF-ODE map fails to be OT in these cases. Here, we use the Frobenius norm of the difference of A and its transpose divided by the Frobenius norm of A to measure its asymmetric rate. Based on this finding, we hypothesize that the PF-ODE map can only be OT in the affine initial data (Single-Gaussian is a special case of affine data), but fails to be OT in general non-affine data. Our hypothesis coincides with the statement that the flow map of a Fokker–Planck equation is not necessarily OT as proposed in (Lavenant & Santambrogio, 2022). (Note that PF-ODE is a special class of Fokker–Planck equations)

7. Conclusions

This paper derives an outer-product span formulation of the diffusion Fisher and introduces two approximation algorithms for different types of access to the diffusion Fisher: diffusion Fisher trace matching (DF-TM) and diffusion Fisher endpoint approximation (DF-EA). Both methods are theoretically guaranteed in terms of approximation error bounds, and offer improved accuracy and reduced time-cost compared to the traditional VJP method. We also designed the numerical OT property verification experiment for the PF-ODE map based on the outer-product form of DF. This work not only improves the efficiency of diffusion Fisher access but also widens our understanding of the diffusion models. Please refer to further discussions in appendix C. The code is available at <https://github.com/zituitui/DiffusionFisher>.

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. We would like to thank all the reviewers for their constructive

comments. Fangyikang Wang wishes to express gratitude to Pengze Zhang from ByteDance, Binxin Yang, and Xinhang Leng from WeChat Vision for their insightful discussions on the experiments.

Impact Statement

The development of accurate diffusion Fisher, as discussed in this paper, holds significant potential for several domains, including machine learning, healthcare, environmental modeling, and economics. However, while this research holds great potential for positive impacts, it is also important to consider potential negative societal impacts. The enhanced ability of generative models given by DF could potentially be misused. For instance, it could be exploited to create aesthetically improved deepfakes, leading to misinformation. In healthcare, if not properly regulated, the use of synthetic patient data could lead to ethical issues. Therefore, it is crucial to ensure that the findings of this research are applied ethically and responsibly, with necessary safeguards in place to prevent misuse and protect privacy.

References

- Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.
- Amann, H. *Ordinary differential equations: an introduction to nonlinear analysis*, volume 13. Walter de gruyter, 2011.
- Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.
- Bao, F., Li, C., Sun, J., Zhu, J., and Zhang, B. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. *arXiv preprint arXiv:2206.07309*, 2022a.
- Bao, F., Li, C., Zhu, J., and Zhang, B. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022b.
- Bengio, Y., Yao, L., and Cho, K. Bounding the test log-likelihood of generative models. *arXiv preprint arXiv:1311.6184*, 2013.
- Benton, J. et al. Nearly d -linear convergence bounds for diffusion models via stochastic localization. In *The Twelfth International Conference on Learning Representations*, 2024.

- Blasingame, Z. W. and Liu, C. Adjointdeis: Efficient gradients for diffusion models. *arXiv preprint arXiv:2405.15020*, 2024.
- Brenier, Y. Décomposition polaire et réarrangement monotone des champs de vecteurs. *CR Acad. Sci. Paris Sér. I Math.*, 305:805–808, 1987.
- Brenier, Y. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.
- Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- Chen, Z., Li, H., Wang, F., Zhang, O., Xu, H., Jiang, X., Song, Z., and Wang, E. H. Rethinking the diffusion models for numerical tabular data imputation from the perspective of wasserstein gradient flow. *arXiv preprint arXiv:2406.15762*, 2024.
- Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021a.
- Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021b.
- Feng, Q., Zhao, H., Zhang, C., Dong, J., Ding, H., Jiang, Y.-G., and Qian, H. Pectp: Parameter-efficient cross-task prompts for incremental vision transformer. *arXiv preprint arXiv:2407.03813*, 2024a.
- Feng, Q., Zhou, D., Zhao, H., Zhang, C., and Qian, H. Lw2g: Learning whether to grow for prompt-based continual learning. *arXiv preprint arXiv:2409.18860*, 2024b.
- Fu, H., Zhao, H., Dong, J., Zhang, C., and Qian, H. Iap: Improving continual learning of vision-language models via instance-aware prompting. *arXiv preprint arXiv:2503.20612*, 2025.
- Gohberg, I., Goldberg, S., Kaashoek, M. A., Gohberg, I., Goldberg, S., and Kaashoek, M. A. Hilbert-schmidt operators. *Classes of Linear Operators Vol. I*, pp. 138–147, 1990.
- Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.
- Kantorovich, L. V. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.
- Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.
- Khrulkov, V., Ryzhakov, G., Chertkov, A., and Oseledets, I. Understanding DDPM latent codes through optimal transport. In *International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=6PIrhAx1j4i>.
- Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 21696–21707. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b578f2a52a0229873fefc2a4b06377fa-Paper.pdf.
- Lang, S. *Fundamentals of differential geometry*, volume 191. Springer Science & Business Media, 2012.
- Lavenant, H. and Santambrogio, F. The flow map of the fokker–planck equation does not provide optimal transport. *Applied Mathematics Letters*, 133:108225, 2022.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.
- Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- Liu, L., Ren, Y., Lin, Z., and Zhao, Z. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.
- Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.
- Lu, C., Zheng, K., Bao, F., Chen, J., Li, C., and Zhu, J. Maximum likelihood training for score-based diffusion ODEs by high order denoising score matching. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14429–14460. PMLR, 17–23 Jul 2022a. URL <https://proceedings.mlr.press/v162/lu22f.html>.

- Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022b.
- Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022c.
- Masuyama, H. Limit formulas for the normalized fundamental matrix of the northwest-corner truncation of markov chains: Matrix-infinite-product-form solutions of block-hessenberg markov chains. *arXiv preprint arXiv:1603.07787*, 2016.
- Monge, G. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704, 1781.
- Nichol, A. Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 16784–16804, 2022.
- Osler, T. J. Leibniz rule for fractional derivatives generalized and an application to infinite series. *SIAM Journal on Applied Mathematics*, 18(3):658–674, 1970.
- Pan, J., Liew, J. H., Tan, V. Y., Feng, J., and Yan, H. Adjoint-dpm: Adjoint sensitivity method for gradient backpropagation of diffusion probabilistic models. *arXiv preprint arXiv:2307.10711*, 2023a.
- Pan, J., Yan, H., Liew, J. H., Feng, J., and Tan, V. Y. Towards accurate guided diffusion sampling through symplectic adjoint method. *arXiv preprint arXiv:2312.12030*, 2023b.
- Pollini, N., Lavan, O., and Amir, O. Adjoint sensitivity analysis and optimization of hysteretic dynamic systems with nonlinear viscous dampers. *Structural and Multidisciplinary Optimization*, 57:2273–2289, 2018.
- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- Rockafellar, R. T. Convex analysis:(pms-28). 2015.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.
- Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494, 2022.
- Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.
- Santambrogio, F. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.
- Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C. W., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S. R., Crowson, K., Schmidt, L., Kaczmarczyk, R., and Jitsev, J. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 25278–25294, 2022.
- Skorski, M. Modern analysis of hutchinson’s trace estimator. In *2021 55th CISS. IEEE*, 2021.
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.
- Song, K. and Lai, H. Fisher information improved training-free conditional diffusion model. *arXiv preprint arXiv:2404.18252*, 2024.
- Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34: 1415–1428, 2021.
- Theis, L., Oord, A. v. d., and Bethge, M. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.
- Tu, J., Fu, H., Yang, F., Zhao, H., Zhang, C., and Qian, H. Texttoucher: Fine-grained text-to-touch generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7455–7463, 2025a.

- Tu, J., Ji, W., Zhao, H., Zhang, C., Zimmermann, R., and Qian, H. Driveditfit: Fine-tuning diffusion transformers for autonomous driving data generation. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(3):1–29, 2025b.
- Villani, C. et al. *Optimal transport: old and new*, volume 338. Springer, 2009.
- Wallace, B., Gokul, A., and Naik, N. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22532–22541, 2023.
- Wang, F., Yin, H., Dong, Y.-J., Zhu, H., Zhao, H., Qian, H., Li, C., et al. Belm: Bidirectional explicit linear multi-step sampler for exact inversion in diffusion models. *Advances in Neural Information Processing Systems*, 37:46118–46159, 2024a.
- Wang, F., Zhu, H., Zhang, C., Zhao, H., and Qian, H. Gad-pvi: A general accelerated dynamic-weight particle-based variational inference framework. *Entropy*, 26(8):679, 2024b.
- Wu, Y., Chen, M., Li, Z., Wang, M., and Wei, Y. Theoretical insights for diffusion guidance: A case study for gaussian mixture models. *arXiv preprint arXiv:2403.01639*, 2024.
- Yang, Z., Feng, R., Zhang, H., Shen, Y., Zhu, K., Huang, L., Zhang, Y., Liu, Y., Zhao, D., Zhou, J., et al. Lipschitz singularities in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.
- Yosida, K. *Functional analysis*, volume 123. Springer Science & Business Media, 2012.
- Zhang, G., Lewis, J. P., and Kleijn, W. B. Exact diffusion inversion via bi-directional integration approximation. *arXiv preprint arXiv:2307.10829*, 2023.
- Zhang, P., Yin, H., Li, C., and Xie, X. Formulating discrete probability flow through optimal transport. *Advances in Neural Information Processing Systems*, 36, 2024a.
- Zhang, P., Yin, H., Li, C., and Xie, X. Tackling the singularities at the endpoints of time intervals in diffusion models. *arXiv preprint arXiv:2403.08381*, 2024b.
- Zheng, K., Lu, C., Chen, J., and Zhu, J. Improved techniques for maximum likelihood estimation for diffusion ODEs. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42363–42389. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/zheng23c.html>.
- Zhu, H., Wang, F., Ding, T., Qu, Q., and Zhu, Z. Analyzing and improving model collapse in rectified flow models. *arXiv preprint arXiv:2412.08175*, 2024a.
- Zhu, H., Wang, F., Zhang, C., Zhao, H., and Qian, H. Neural sinkhorn gradient flow. *arXiv preprint arXiv:2401.14069*, 2024b.

Appendix

A. Proofs and Formulations

A.1. Proof of Proposition 1

Notice that, in the subsection, we can do an interchange of sum and gradient; this is due to Leibniz's rule (Osler, 1970) and the boundness condition we set in equation 10. Before we give the proof of Proposition 1, we would like to establish two technical lemmas. The first lemma is about the first partial derivative of $v_i(\mathbf{x}_t, t)$ with respect to $\mathbf{x}_t$.

Lemma 1.

$$\begin{aligned} \frac{\partial v_i(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} &= \frac{\partial \exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}_i|^2}{2\sigma_t^2}\right)}{\partial \mathbf{x}_t} \\ &= -\frac{1}{\sigma_t^2} (\mathbf{x}_t - \alpha_t \mathbf{y}_i) \exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}_i|^2}{2\sigma_t^2}\right) \\ &= -\frac{1}{\sigma_t^2} (\mathbf{x}_t - \alpha_t \mathbf{y}_i) v_i(\mathbf{x}_t, t) \end{aligned} \quad (23)$$

The second lemma is about the Jacobian of $\sum_i w_i(\mathbf{x}_t, t) \mathbf{y}_i$ w.r.t. $\mathbf{x}_t$.

Lemma 2.

$$\begin{aligned} &\frac{\partial \sum_i w_i(\mathbf{x}_t, t) \mathbf{y}_i}{\partial \mathbf{x}_t} \\ &= \sum_i \mathbf{y}_i \left(\frac{\partial w_i(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \right)^\top \\ &= \sum_i \mathbf{y}_i \left(\frac{\partial}{\partial \mathbf{x}_t} \left[\frac{v_i(\mathbf{x}_t, t)}{\sum_j v_j(\mathbf{x}_t, t)} \right] \right)^\top \\ &= \sum_i \mathbf{y}_i \left\{ \frac{\frac{\partial v_i(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} [\sum_k v_k(\mathbf{x}_t, t)] - v_i(\mathbf{x}_t, t) \frac{\partial}{\partial \mathbf{x}_t} [\sum_j v_j(\mathbf{x}_t, t)]}{[\sum_k v_k(\mathbf{x}_t, t)]^2} \right\}^\top \\ &= \frac{1}{(\sum_k v_k)^2} \sum_i \mathbf{y}_i \left\{ -\frac{1}{\sigma_t^2} (\mathbf{x}_t - \alpha_t \mathbf{y}_i) v_i \sum_k v_k - v_i(\mathbf{x}_t, t) \left(-\frac{1}{\sigma_t^2} \right) \sum_j (\mathbf{x}_t - \alpha_t \mathbf{y}_j) v_j \right\}^\top \\ &= \frac{1}{(\sum_k v_k)^2} \left(-\frac{1}{\sigma_t^2} \right) \sum_i v_i \mathbf{y}_i \left\{ (\mathbf{x}_t - \alpha_t \mathbf{y}_i) \sum_k v_k - \sum_j (\mathbf{x}_t - \alpha_t \mathbf{y}_j) v_j \right\}^\top \\ &= \frac{1}{(\sum_k v_k)^2} \left(-\frac{1}{\sigma_t^2} \right) \sum_i v_i \mathbf{y}_i \left\{ -\alpha_t \mathbf{y}_i \sum_k v_k + \alpha_t \sum_j \mathbf{y}_j v_j \right\}^\top \\ &= \frac{\alpha_t}{\sigma_t^2} \left[\frac{\sum_i v_i \mathbf{y}_i \mathbf{y}_i^\top (\sum_k v_k)}{(\sum_k v_k)^2} - \left(\sum_i v_i \mathbf{y}_i \right) \left(\sum_i v_i \mathbf{y}_i \right)^\top \right] \\ &= \frac{\alpha_t}{\sigma_t^2} \left[\sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top - \left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top \right] \end{aligned} \quad (24)$$

Now we are ready to give the Proof of Proposition 1

Proof. According to the initial distribution (equation 10) and the diffusion kernel (equation 1), the marginal distribution at

some time $t > 0$ would be

$$p(\mathbf{x}_t, t) = \frac{1}{N} \sum_i (2\pi\sigma_t^2)^{-\frac{d}{2}} \exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}_i|^2}{2\sigma_t^2}\right) \quad (25)$$

Thus, the log-density has the following analytical formulation

$$\begin{aligned} \log p(\mathbf{x}_t, t) &= \log \left[\frac{1}{N} (2\pi\sigma_t^2)^{-\frac{d}{2}} \sum_i \exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}_i|^2}{2\sigma_t^2}\right) \right] \\ &= \log \left[\sum_i \exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}_i|^2}{2\sigma_t^2}\right) \right] + C \\ &= \log \left[\sum_i v_i(\mathbf{x}_t, t) \right] + C \end{aligned} \quad (26)$$

The score can be expressed as follows

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}_t} \log p(\mathbf{x}_t, t) &= \frac{\partial}{\partial \mathbf{x}_t} \log \left[\sum_i v_i(\mathbf{x}_t, t) \right] \\ &= \frac{\frac{\partial}{\partial \mathbf{x}_t} [\sum_i v_i(\mathbf{x}_t, t)]}{\sum_i v_i(\mathbf{x}_t, t)} \\ &= \frac{-\frac{1}{\sigma_t^2} \sum_j (\mathbf{x}_t - \alpha_t \mathbf{y}_j) v_j}{\sum_i v_i(\mathbf{x}_t, t)} \\ &= -\frac{1}{\sigma_t^2} \left[\mathbf{x}_t - \alpha_t \sum_j w_j(\mathbf{x}_t, t) \mathbf{y}_j \right] \end{aligned} \quad (27)$$

The Fisher information we want can then be calculated by further applying a gradient on the score.

$$\begin{aligned} \mathbf{F}_t(\mathbf{x}_t, t) &= -\frac{\partial}{\partial \mathbf{x}_t} \left(\frac{\partial}{\partial \mathbf{x}_t} \log p(\mathbf{x}_t, t) \right) \\ &= -\frac{\partial}{\partial \mathbf{x}_t} \left\{ -\frac{1}{\sigma_t^2} \left[\mathbf{x}_t - \alpha_t \sum_j w_j(\mathbf{x}_t, t) \mathbf{y}_j \right] \right\} \\ &= \frac{1}{\sigma_t^2} \mathbf{I} - \frac{\alpha_t}{\sigma_t^2} \frac{\partial \sum_i w_i(\mathbf{x}_t, t) \mathbf{y}_i}{\partial \mathbf{x}_t} \\ &= \frac{1}{\sigma_t^2} \mathbf{I} - \frac{\alpha_t^2}{\sigma_t^4} \left[\sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top - \left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top \right] \quad (\text{by Lemma 2}) \end{aligned} \quad (28)$$

□

This proof is inherently the calculation of the Hessian of a log-convolution of a density, we provide the detailed derivation here for completeness.

A.2. Proof of Proposition 2

Proof. Given fixed (x_t, t) , $\mathcal{L}$ is a quadratic form of y_θ . To obtain the optimal $\bar{\mathbf{y}}_\theta$, we differentiate $\mathcal{L}$ and set this derivative equal to zero, resulting in the following

$$\begin{aligned}
 0 &= \frac{\partial \mathcal{L}}{\partial \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)} = \frac{\partial}{\partial \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)} \sum_j \underbrace{\frac{1}{N} (2\pi\sigma_t^2)^{-\frac{d}{2}}}_{A_t} v_j(\mathbf{x}_t, t) \lambda_t \frac{\alpha_t^2}{\sigma_t^2} \|\bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) - \mathbf{y}_j\|^2 \\
 &= 2A_t \lambda_t \frac{\alpha_t^2}{\sigma_t^2} \sum_j v_j(\mathbf{x}_t, t) (\bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) - \mathbf{y}_j),
 \end{aligned} \tag{29}$$

which yields

$$\bar{\mathbf{y}}_\theta^*(\mathbf{x}_t, t) = \sum_k \frac{v_k(\mathbf{x}_t, t)}{\sum_j v_j(\mathbf{x}_t, t)} \mathbf{y}_k = \sum_i w_i \mathbf{y}_i. \tag{30}$$

□

A.3. Proof of Proposition 3

Notice that, in the subsection, we can do an interchange of integral and gradient; this is due to Leibniz's rule (Osler, 1970) and the bounded moments condition we set in equation 12. Before we give the proof of Proposition 3, we would like to establish two technical lemmas. The first lemma is about the first partial derivative of $v(\mathbf{x}_t, t, \mathbf{y})$ w.r.t. $\mathbf{x}_t$.

Lemma 3.

$$\begin{aligned}
 \frac{\partial v(\mathbf{x}_t, t, \mathbf{y})}{\partial \mathbf{x}_t} &= \frac{\partial \exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}|^2}{2\sigma_t^2}\right)}{\partial \mathbf{x}_t} \\
 &= -\frac{1}{\sigma_t^2} (\mathbf{x}_t - \alpha_t \mathbf{y}) \exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}|^2}{2\sigma_t^2}\right) \\
 &= -\frac{1}{\sigma_t^2} (\mathbf{x}_t - \alpha_t \mathbf{y}) v(\mathbf{x}_t, t, \mathbf{y})
 \end{aligned} \tag{31}$$

Lemma 4.

$$\begin{aligned}
 &\frac{\partial \int_{\mathbb{R}^d} w(\mathbf{x}_t, t, \mathbf{y}) \mathbf{y} d q_0(\mathbf{y})}{\partial \mathbf{x}_t} \\
 &= \int_{\mathbb{R}^d} \mathbf{y} \left(\frac{\partial w(\mathbf{x}_t, t, \mathbf{y})}{\partial \mathbf{x}_t} \right)^\top d q_0(\mathbf{y}) \\
 &= \int_{\mathbb{R}^d} \mathbf{y} \left(\frac{\partial}{\partial \mathbf{x}_t} \left[\frac{v(\mathbf{x}_t, t, \mathbf{y})}{\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}') d q_0(\mathbf{y}')} \right] \right)^\top d q_0(\mathbf{y}) \\
 &= \int_{\mathbb{R}^d} \mathbf{y} \left\{ \frac{\frac{\partial v(\mathbf{x}_t, t, \mathbf{y})}{\partial \mathbf{x}_t} \left[\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}') d q_0(\mathbf{y}') \right] - v(\mathbf{x}_t, t, \mathbf{y}) \frac{\partial}{\partial \mathbf{x}_t} \left[\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}') d q_0(\mathbf{y}') \right]}{\left[\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}') d q_0(\mathbf{y}') \right]^2} \right\}^\top d q_0(\mathbf{y}) \\
 &= \frac{1}{\left[\int v(\mathbf{y}'') d q_0(\mathbf{y}'') \right]^2} \int \mathbf{y} \left\{ -\frac{(\mathbf{x}_t - \alpha_t \mathbf{y}) v(\mathbf{y})}{\sigma_t^2} \int v(\mathbf{y}') d q_0(\mathbf{y}') - v(\mathbf{y}) \left(-\frac{1}{\sigma_t^2} \right) \int (\mathbf{x}_t - \alpha_t \mathbf{y}') v(\mathbf{y}') d q_0(\mathbf{y}') \right\}^\top d q_0(\mathbf{y}) \tag{32} \\
 &= \frac{1}{\left[\int v(\mathbf{y}'') d q_0(\mathbf{y}'') \right]^2} \left(-\frac{1}{\sigma_t^2} \right) \int v(\mathbf{y}) \mathbf{y} \left\{ (\mathbf{x}_t - \alpha_t \mathbf{y}) \int v(\mathbf{y}') d q_0(\mathbf{y}') - \int (\mathbf{x}_t - \alpha_t \mathbf{y}') v(\mathbf{y}') d \mathbf{y}' \right\}^\top d q_0(\mathbf{y}) \\
 &= \frac{1}{\left[\int v(\mathbf{y}'') d q_0(\mathbf{y}'') \right]^2} \left(-\frac{1}{\sigma_t^2} \right) \int v(\mathbf{y}) \mathbf{y} \left\{ -\alpha_t \mathbf{y} \int v(\mathbf{y}') d q_0(\mathbf{y}') + \alpha_t \int v(\mathbf{y}') \mathbf{y}' d \mathbf{y}' \right\}^\top d q_0(\mathbf{y}) \\
 &= \frac{\alpha_t}{\sigma_t^2} \left[\frac{\int v(\mathbf{y}) \mathbf{y} \mathbf{y}^\top d q_0(\mathbf{y}') \left[\int v(\mathbf{y}') d q_0(\mathbf{y}') \right]}{\left[\int v(\mathbf{y}'') d q_0(\mathbf{y}'') \right]^2} - \left(\frac{\int v(\mathbf{y}) \mathbf{y} d q_0(\mathbf{y})}{\int v(\mathbf{y}'') d q_0(\mathbf{y}'')} \right) \left(\frac{\int v(\mathbf{y}') \mathbf{y}' d q_0(\mathbf{y}')}{\int v(\mathbf{y}'') d q_0(\mathbf{y}'')} \right)^\top \right] \\
 &= \frac{\alpha_t}{\sigma_t^2} \left[\int w(\mathbf{y}) \mathbf{y} \mathbf{y}^\top d q_0(\mathbf{y}) - \left(\int w(\mathbf{y}) \mathbf{y} d q_0(\mathbf{y}) \right) \left(\int w(\mathbf{y}) \mathbf{y} d q_0(\mathbf{y}) \right)^\top \right]
 \end{aligned}$$

Proof. According to the initial distribution (equation 12) and the diffusion kernel (equation 1), the marginal distribution at

some time $t > 0$ would be

$$p(\mathbf{x}_t, t) = \int_{\mathbb{R}^d} (2\pi\sigma_t^2)^{-\frac{d}{2}} \exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}|^2}{2\sigma_t^2}\right) dq_0(\mathbf{y}) \quad (33)$$

Thus the log-density has the following analytical formulation

$$\begin{aligned} \log q_t(\mathbf{x}_t, t) &= \log \left[\int_{\mathbb{R}^d} (2\pi\sigma_t^2)^{-\frac{d}{2}} \exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}|^2}{2\sigma_t^2}\right) dq_0(\mathbf{y}) \right] \\ &= \log \left[\int_{\mathbb{R}^d} \exp\left(-\frac{|\mathbf{x}_t - \alpha_t \mathbf{y}|^2}{2\sigma_t^2}\right) dq_0(\mathbf{y}) \right] + C \\ &= \log \left[\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}) dq_0(\mathbf{y}) \right] + C \end{aligned} \quad (34)$$

The score can be expressed as follows

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}_t} \log p(\mathbf{x}_t, t) &= \frac{\partial}{\partial \mathbf{x}_t} \log \left[\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}) dq_0(\mathbf{y}) \right] \\ &= \frac{\frac{\partial}{\partial \mathbf{x}_t} \left[\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}) dq_0(\mathbf{y}) \right]}{\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}) dq_0(\mathbf{y})} \\ &= \frac{\int_{\mathbb{R}^d} \frac{\partial}{\partial \mathbf{x}_t} [v(\mathbf{x}_t, t, \mathbf{y})] dq_0(\mathbf{y})}{\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}) dq_0(\mathbf{y})} \quad (\text{by xx and Leibniz integral rule}) \\ &= \frac{-\frac{1}{\sigma_t^2} \int_{\mathbb{R}^d} (\mathbf{x}_t - \alpha_t \mathbf{y}) v(\mathbf{x}_t, t, \mathbf{y}) dq_0(\mathbf{y})}{\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}) dq_0(\mathbf{y})} \\ &= -\frac{1}{\sigma_t^2} \left[\mathbf{x}_t - \alpha_t \int_{\mathbb{R}^d} w(\mathbf{x}_t, t, \mathbf{y}) \mathbf{y} dq_0(\mathbf{y}) \right] \end{aligned} \quad (35)$$

The Fisher information we want can then be calculated by further applying a gradient on the score.

$$\begin{aligned} \mathbf{F}_t(\mathbf{x}_t, t) &= -\frac{\partial}{\partial \mathbf{x}_t} \left(\frac{\partial}{\partial \mathbf{x}_t} \log p(\mathbf{x}_t, t) \right) \\ &= -\frac{\partial}{\partial \mathbf{x}_t} \left\{ -\frac{1}{\sigma_t^2} \left[\mathbf{x}_t - \alpha_t \int_{\mathbb{R}^d} w(\mathbf{x}_t, t, \mathbf{y}) \mathbf{y} dq_0(\mathbf{y}) \right] \right\} \\ &= \frac{1}{\sigma_t^2} \mathbf{I} - \frac{\alpha_t}{\sigma_t^2} \frac{\partial \int_{\mathbb{R}^d} w(\mathbf{x}_t, t, \mathbf{y}) \mathbf{y} dq_0(\mathbf{y})}{\partial \mathbf{x}_t} \\ &= \frac{1}{\sigma_t^2} \mathbf{I} - \frac{\alpha_t^2}{\sigma_t^4} \left[\int w_i \mathbf{y} \mathbf{y}^\top dq_0(\mathbf{y}) - \left(\int w_i \mathbf{y} dq_0(\mathbf{y}) \right) \left(\int w_i \mathbf{y} dq_0(\mathbf{y}) \right)^\top \right] \quad (\text{by Lemma 4}) \end{aligned} \quad (36)$$

□

This proof is also inherently the calculation of the Hessian of a log-convolution of a density, we provide the detailed derivation here for completeness.

A.4. Proof of Proposition 4

Proof. Given fixed (x_t, t) , $\mathcal{L}$ is a quadratic form of y_θ . To obtain the optimal $\bar{y}_\theta$, we differentiate $\mathcal{L}$ and set this derivative equal to zero, resulting in the following

$$\begin{aligned}
 0 &= \frac{\partial \mathcal{L}}{\partial \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)} = \frac{\partial}{\partial \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)} \int_{\mathbb{R}^d} \underbrace{\frac{1}{N} (2\pi\sigma_t^2)^{-\frac{d}{2}}}_{A_t} v(\mathbf{x}_t, t, \mathbf{y}) \lambda_t \frac{\alpha_t^2}{\sigma_t^2} \|\bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) - \mathbf{y}\|^2 d q_0(\mathbf{y}) \\
 &= 2A_t \lambda_t \frac{\alpha_t^2}{\sigma_t^2} \int_{\mathbb{R}^d} \mathbf{v}(\mathbf{x}_t, t, \mathbf{y}) (\bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) - \mathbf{y}) d q_0(\mathbf{y}),
 \end{aligned} \tag{37}$$

which yields

$$\bar{\mathbf{y}}_\theta^*(\mathbf{x}_t, t) = \int_{\mathbb{R}^d} \frac{v(\mathbf{x}_t, t, \mathbf{y}')}{\int_{\mathbb{R}^d} v(\mathbf{x}_t, t, \mathbf{y}'') d q_0(\mathbf{y}'')} \mathbf{y}' d q_0(\mathbf{y}') = \int_{\mathbb{R}^d} w(\mathbf{x}_t, t, \mathbf{y}') \mathbf{y}' d q_0(\mathbf{y}'). \tag{38}$$

□

A.5. Proof of Proposition 5

Lemma 5. Given a vector $\mathbf{v} \in \mathbb{R}^d$, the trace of the outer-product matrix of this vector is precisely equal to the square of its 2-norm. This can be shown as follows:

$$\text{tr}(\mathbf{v}\mathbf{v}^T) = \sum_{i=1}^d (\mathbf{v}\mathbf{v}^T)_{i,i} = \sum_{i=1}^d v_i * v_i = \|\mathbf{v}\|^2 \tag{39}$$

Proof. Then we can start to give the derivation of Proposition 5

$$\begin{aligned}
 \text{tr}(\mathbf{F}_t(\mathbf{x}_t, t)) &= \text{tr} \left(\frac{1}{\sigma_t^2} \mathbf{I} - \frac{\alpha_t^2}{\sigma_t^4} \left[\sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top - \left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top \right] \right) \\
 &= \frac{1}{\sigma_t^2} \text{tr}(\mathbf{I}) - \frac{\alpha_t^2}{\sigma_t^4} \left[\sum_i w_i \text{tr}(\mathbf{y}_i \mathbf{y}_i^\top) - \text{tr} \left(\left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top \right) \right] \\
 &= \frac{d}{\sigma_t^2} - \frac{\alpha_t^2}{\sigma_t^4} \left[\sum_i w_i \|\mathbf{y}_i\|^2 - \left\| \sum_i w_i \mathbf{y}_i \right\|^2 \right]
 \end{aligned} \tag{40}$$

□

A.6. Proof of Proposition 6

Proof. The objective in Algorithm 1 obviously equals to:

$$\arg \min_{t_\theta} \mathbb{E}_{\mathbf{x}_0 \sim q_0(\mathbf{x}_0), \mathbf{x}_t \sim \mathcal{N}(\alpha(t)\mathbf{x}_0, \sigma^2(t))} \left| \mathbf{t}_\theta(\mathbf{x}_t, t) - \frac{\|\mathbf{x}_0\|^2}{d} \right|^2. \tag{41}$$

By expressing the expectation of Equation equation 41 in the form of a marginal distribution, we can transform the objective as follows:

$$\arg \min_{t_\theta} \sum_i \frac{1}{N} (2\pi\sigma_t^2)^{-\frac{d}{2}} \left| \mathbf{t}_\theta(\mathbf{x}_t, t) - \frac{\|\mathbf{y}_i\|^2}{d} \right|^2 \tag{42}$$

The optimal t_θ^* must satisfy the condition that the gradient of the loss equals 0. Therefore, we have:

$$\begin{aligned}
 0 &= \nabla_{t_\theta^*(\mathbf{x}_t, t)} \left[\sum_i \underbrace{\frac{1}{N} (2\pi\sigma_t^2)^{-\frac{d}{2}} v_i(\mathbf{x}_t, t)}_{A_t} \left| \frac{\|\mathbf{y}_i\|^2}{d} - t_\theta^*(\mathbf{x}_t, t) \right|^2 \right] \\
 &= \sum_i A_t v_i(\mathbf{x}_t, t) (t_\theta^*(\mathbf{x}_t, t) - \frac{\|\mathbf{y}_i\|^2}{d}) \\
 &= A_t \sum_j v_j(\mathbf{x}_t, t) t_\theta^*(\mathbf{x}_t, t) - A_t \sum_i v_i(\mathbf{x}_t, t) \frac{\|\mathbf{y}_i\|^2}{d},
 \end{aligned}$$

Thus

$$\begin{aligned}
 t_\theta^*(\mathbf{x}_t, t) &= \frac{A_t \sum_i v_i(\mathbf{x}_t, t) \frac{\|\mathbf{y}_i\|^2}{d}}{A_t \sum_j v_j(\mathbf{x}_t, t)} \\
 &= \sum_i \frac{v_i(\mathbf{x}_t, t)}{\sum_j v_j(\mathbf{x}_t, t)} \frac{\|\mathbf{y}_i\|^2}{d} \\
 &= \frac{1}{d} \sum_i w_i(\mathbf{x}_t, t) \|\mathbf{y}_i\|^2
 \end{aligned} \tag{43}$$

We have successfully completed the proof that the optimal $t_\theta(\mathbf{x}_t, t)$, as trained by Algorithm 1, is equivalent to $\frac{1}{d} \sum_i w_i(\mathbf{x}_t, t) \|\mathbf{y}_i\|^2$. $\square$

A.7. Proof of Proposition 7

The approximation error of the estimated trace equation 17 will be its difference from the true Fisher information trace equation 16. We use consecutive Cauchy–Schwarz and triangle inequality to get the bound of the approximation error:

$$\begin{aligned}
 &\left| \frac{d}{\sigma_t^2} - \frac{\alpha_t^2}{\sigma_t^4} \left[\sum_i w_i \|\mathbf{y}_i\|^2 - \left\| \sum_i w_i \mathbf{y}_i \right\|^2 \right] - \left\{ d \left[\frac{1}{\sigma_t^2} - \frac{\alpha_t^2}{\sigma_t^4} \left(t_\theta(\mathbf{x}_t, t) - \left\| \frac{\mathbf{x}_t - \sigma_t \boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t)}{\alpha_t} \right\|^2 \right) \right] \right\} \right| \\
 &= \frac{\alpha_t^2}{\sigma_t^4} \left| \sum_i w_i \|\mathbf{y}_i\|^2 - \left\| \sum_i w_i \mathbf{y}_i \right\|^2 - \left(t_\theta(\mathbf{x}_t, t) - \left\| \frac{\mathbf{x}_t - \sigma_t \boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t)}{\alpha_t} \right\|^2 \right) \right| \\
 &\leq \frac{\alpha_t^2}{\sigma_t^4} \left[\left| \sum_i w_i \|\mathbf{y}_i\|^2 - t_\theta(\mathbf{x}_t, t) \right| + \left\| \sum_i w_i \mathbf{y}_i - \frac{\mathbf{x}_t - \sigma_t \boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t)}{\alpha_t} \right\| \right] \\
 &\leq \frac{\alpha_t^2}{\sigma_t^4} \left[\delta_1 + \frac{\sigma_t^2}{\alpha_t^2} \delta_2^2 \right] \\
 &= \frac{\alpha_t^2}{\sigma_t^4} \delta_1 + \frac{1}{\sigma_t^2} \delta_2^2
 \end{aligned} \tag{44}$$

A.8. Proof of Proposition 8

$$\begin{aligned}
 & \left\| \frac{1}{\sigma_t} \mathbf{I} - \frac{\alpha_t^2}{\sigma_t^3} \left[\sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top - \left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top \right] - \left(\frac{1}{\sigma_t} \mathbf{I} - \frac{\alpha_t^2}{\sigma_t^3} (\mathbf{x}_0 \mathbf{x}_0^\top - \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)^\top) \right) \right\|_{HS} \\
 &= \left\| -\frac{\alpha_t^2}{\sigma_t^3} \left[\sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top - \left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top \right] - \left(-\frac{\alpha_t^2}{\sigma_t^3} (\mathbf{x}_0 \mathbf{x}_0^\top - \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)^\top) \right) \right\|_{HS} \\
 &\leq \frac{\alpha_t^2}{\sigma_t^3} \left\| \sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top - \mathbf{x}_0 \mathbf{x}_0^\top \right\|_{HS} + \frac{\alpha_t^2}{\sigma_t^3} \left\| \left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top - \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)^\top \right\|_{HS} \\
 &= \frac{\alpha_t^2}{\sigma_t^3} \sum_i w_i \left\| \mathbf{y}_i \mathbf{y}_i^\top - \mathbf{x}_0 \mathbf{x}_0^\top \right\|_{HS} + \frac{\alpha_t^2}{\sigma_t^3} \left\| \left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top - \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)^\top \right\|_{HS} \\
 &\leq \frac{\alpha_t^2}{\sigma_t^3} \sum_i w_i \max_i \left\| \mathbf{y}_i \mathbf{y}_i^\top - \mathbf{x}_0 \mathbf{x}_0^\top \right\|_{HS} + \frac{\alpha_t^2}{\sigma_t^3} \sum_j \left\| \sum_i w_i \mathbf{y}_i[j] - \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t)[j] \right\| \left\| \sum_i w_i \mathbf{y}_i - \bar{\mathbf{y}}_\theta(\mathbf{x}_t, t) \right\| \\
 &\leq \frac{\alpha_t^2}{\sigma_t^3} (2\mathcal{D}_y^2 + \sqrt{d}\delta_2)
 \end{aligned} \tag{45}$$

A.9. Preliminaries on Optimal Transport

The optimal transport is the general problem of moving one distribution of mass to another as efficiently as possible. The *optimal transport problem* can be formulated in two primary ways, namely the Monge formulation (Monge, 1781) and the Kantorovich formulation (Kantorovich, 1960). Suppose there are two probability measures μ and ν on $(\mathbb{R}^n, \mathcal{B})$, and a cost function $c : \mathbb{R}^n \times \mathbb{R}^n \rightarrow [0, +\infty]$. The *Monge problem* is

$$(\text{MP}) \quad \inf_{\mathbf{T}} \left\{ \int c(x, \mathbf{T}(x)) d\mu(x) : \mathbf{T}_\# \mu = \nu \right\}. \tag{46}$$

The measure $\mathbf{T}_\# \mu$ is defined through $\mathbf{T}_\# \mu(A) = \mu(\mathbf{T}^{-1}(A))$ for every $A \in \mathcal{B}$ and is called the *pushforward* of μ through $\mathbf{T}$.

It is evident that the Monge Problem (MP) transports the entire mass from a particular point, denoted as x , to a single point $\mathbf{T}(x)$. In contrast, Kantorovich provided a more general formulation, referred to as the *Kantorovich problem*:

$$(\text{KP}) \quad \inf_{\gamma} \left\{ \int_{\mathbb{R}^n \times \mathbb{R}^n} c d\gamma : \gamma \in \Pi(\mu, \nu) \right\}, \tag{47}$$

where $\Pi(\mu, \nu)$ is the set of *transport plans*, i.e.,

$$\Pi(\mu, \nu) = \{ \gamma \in \mathcal{P}(\mathbb{R}^n \times \mathbb{R}^n) : (\pi_x)_\# \gamma = \mu, (\pi_y)_\# \gamma = \nu \}, \tag{48}$$

where π_x and π_y are the two projections of $\mathbb{R}^n \times \mathbb{R}^n$ onto $\mathbb{R}^n$. For measures absolutely continuous with respect to the Lebesgue measure, these two problems are equivalent (Villani et al., 2009). However, when the measures are discrete, they are entirely distinct as the constraint of the Monge Problem may never be fulfilled.

A.10. Proof of Corollary 1

To prove the Corollary 1, we first introduce two theorems to transform the problem of whether the PF-ODE mapping is a Monge map into the task of deciding the convexity of the potential function of $T_{s,T}$.

Theorem 1. (Santambrogio, 2015, Theorem 1.48) Suppose that μ is a probability measure on $(\mathbb{R}^n, \mathcal{B})$ such that $\int |x|^2 d\mu(x) < \infty$ and that $u : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is convex and differentiable μ -a.e. Set $\mathbf{T} = \nabla u$ and suppose $\int |\mathbf{T}(x)|^2 d\mu(x) < \infty$. Then $\mathbf{T}$ is optimal for the transport cost $c(x, y) = \frac{1}{2}|x - y|^2$ between the measures μ and $\nu = \mathbf{T}_\# \mu$.

Theorem 2. The Brenier's Theorem. (Santambrogio, 2015, Theorem 1.22) (Brenier, 1987; 1991) Let μ, v be probabilities over $\mathbb{R}^d$ and $c(x, y) = \frac{1}{2}|x - y|^2$. Suppose $\int |x|^2 dx, \int |y|^2 dy < +\infty$, which implies $\min(\text{KP}) < +\infty$ and suppose that μ gives no mass to $(d - 1)$ surfaces of class C^2 . Then there exists, uniquely, an optimal transport map T from μ to v , and it is of the form $T = \nabla u$ for a convex function u .

To ensure the existence of the potential function, we need to leverage the following

Theorem 3. The Poincaré's Theorem. (Lang, 2012, Theorem 4.1 of Chapter V, §4) Let U be an open ball in $\mathbb{R}^n$ and let ω be a differential form of degree ≥ 1 on U such that $d\omega = 0$. Then there exists a differential form ϕ on U such that $d\phi = \omega$.

Remark 1. The conclusion remains valid when the open ball U is substituted with the entirety of $\mathbb{R}^n$

For $s > 0$, it is clear that $\frac{dq_T(x_T)}{dq_s(x_s)}$ is non-singular, and therefore, it satisfies the requirements of Brenier's Theorem 2, leading to the existence of a unique optimal transport map. According to Theorem 1, if we can establish that the potential function of $T_{s,T}$ is convex, then the PF-ODE mapping will indeed be a Monge map. Notice that the existence of the potential map is guaranteed by Poincaré's Theorem 3.

We can now convert the condition of the potential function of $T_{s,T}$ being convex into the condition that its Jacobian, $\frac{\partial T_{s,T}(x_s)}{\partial x_s}$, is positive semi-definite, as per the following theorem in convex analysis.

Theorem 4. (Rockafellar, 2015, Theorem 4.5) Let f be a twice continuously differentiable real-valued function on an open convex set C in $\mathbb{R}^n$. Then f is convex on C if and only if its Hessian matrix

$$Q_x = (q_{ij}(x)), \quad q_{ij}(x) = \frac{\partial^2 f}{\partial \xi_i \partial \xi_j}(\xi_1, \dots, \xi_n) \quad (49)$$

is positive semi-definite for every $x \in C$.

If we denote that

$$\mathbf{A}(t) = \frac{\partial T_{t,T}(x_t)}{\partial x_t} \quad (50)$$

obviously, $\mathbf{A}(T) = I$ is p.s.d., our goal is to answer when $\mathbf{A}(t)$ is p.s.d.. We try to answer this to set up a connection between $\mathbf{A}(t)$ and $\mathbf{A}(T) = I$. We can derive that:

$$\begin{aligned} \frac{d\mathbf{A}(t)}{dt} &= \lim_{\epsilon \rightarrow 0^+} \frac{\mathbf{A}(t + \epsilon) - \mathbf{A}(t)}{\epsilon} \\ &= \lim_{\epsilon \rightarrow 0^+} \frac{\mathbf{A}(t + \epsilon) - \mathbf{A}(t + \epsilon) \nabla_{x_t} \left[x_t + \epsilon \left(f(t)x_t - \frac{g^2(t)}{2} \nabla_{x_t} \log_t q_t(x_t) \right) + \mathcal{O}(\epsilon^2) \right]}{\epsilon} \\ &= \lim_{\epsilon \rightarrow 0^+} \frac{\epsilon f(t) \mathbf{A}(t + \epsilon) + \frac{g^2(t)}{2} \epsilon \mathbf{A}(t + \epsilon) \nabla_{x_t} \log_t q_t(x_t) + \mathcal{O}(\epsilon^2)}{\epsilon} \\ &= f(t) \mathbf{A}(t) + \frac{g^2(t)}{2} \mathbf{A}(t) \nabla_{x_t} \log_t q_t(x_t) \\ &= f(t) \mathbf{A}(t) - \frac{g^2(t)}{2} \mathbf{A}(t) \left\{ \frac{1}{\sigma_t^2} \mathbf{I} - \frac{\alpha_t^2}{\sigma_t^4} \left[\int w(\mathbf{y}) \mathbf{y} \mathbf{y}^\top dq_0(\mathbf{y}) - \left(\int w(\mathbf{y}) \mathbf{y} dq_0(\mathbf{y}) \right) \left(\int w(\mathbf{y}) \mathbf{y} dq_0(\mathbf{y}) \right)^\top \right] \right\} \\ &= \mathbf{A}(t) \underbrace{\left\{ \left[f(t) - \frac{g^2(t)}{2\sigma_t^2} \right] \mathbf{I} + \frac{\alpha_t^2 g^2(t)}{2\sigma_t^4} \left[\sum_i w_i \mathbf{y}_i \mathbf{y}_i^\top - \left(\sum_i w_i \mathbf{y}_i \right) \left(\sum_i w_i \mathbf{y}_i \right)^\top \right] \right\}}_{B(t)} \end{aligned} \quad (51)$$

Notice that the above ODE starts from T .

According to the Solution Matrices theory (Masuyama, 2016)¹, let us denote the $C(t)$ is the normalized fundamental matrix at T for $B(T)$, which implies $C(t)$ is the solution to the following ODE:

$$C'(t) = C(t)B(t), C(T) = I, \quad (\text{flow from } T \text{ to } t) \quad (52)$$

Then we can deduce that

$$\begin{aligned} \frac{\partial T_{t,T}(x_t)}{\partial x_t} &= A(t) \\ &= C(t)A(T) \\ &= C(t)I \\ &= C(t) \end{aligned} \quad (53)$$

Thus, the diffeomorphism $T_{s,T}$ is a Monge optimal transport map if and only if $C(s)$ is semi-positive definite. Note that the above requirement needs to be satisfied for every PF-ODE chain $x_t, t \in [T, t]$.

B. Experiments Details

B.1. Evaluation of NLL

For toy experiments, we follow the experiment design in Lu et al. (2022a) and adopt the same network architecture for the score network and our DF-TM network. We calculate the true diffusion Fisher with 5,000 data samples from the cheeseboard distribution.

For commercial-level experiments, we employ the explicit Euler method to compute the NLL, excluding the final step near $t = 0$, for reasons discussed in Appendix C.1. We evaluate the NLL across 10 steps throughout the timeline of the PF-ODE. The DF-TM network we trained uses the float-point16 data type.

Network architectures In terms of network architecture, we employ an SD Unet structure with an additional MLP head. However, we believe that a lighter network could potentially be sufficient for DF-TM.

Training cost of DF-TM For training the DF-TM network, we approximately spend 24 hours using 8 Tesla V100 chips. Given the size of the Laion2B-en dataset (which contains 2.32 billion images), this is quite an efficient speed. Additionally, the convergence behavior of the training loss is robust, as illustrated in Figure 1a. We also hypothesize that our network design has redundancy, suggesting that we could further reduce costs by opting for lighter networks. We will provide more details about training costs in our revised manuscript.

B.2. Adjoint Guidance Sampling

For Figure 3, we examined varying numbers of adjoint guidance, ranging from 0 to 20, under a full inference number of 50. The adjoint guidance scale was grid-searched by the VJP method. For experiments on Pick-Score, we use NVIDIA V100 chips, and the rest experiments use Tesla V100 chips.

Base method for DF-EA Several variants of adjoint-optimization algorithms exist, such as AdjointDPM (Pan et al., 2023a), AdjointDES (Blasingame & Liu, 2024), and SAG (Pan et al., 2023b). However, while these algorithms differ in their design of solvers for the adjoint ODE, they all utilize VJP when accessing the diffusion Fisher. We selected SAG as our base method due to its state-of-the-art performance. We believe that replacing VJP with DF-EA could also enhance the performance of algorithms like AdjointDPM and AdjointDES.

The Design of Score functions

- Aesthetic Score (SAC/AVA/Pick-Score)

¹The Definition 6.2 in <https://math.mit.edu/~jorloff/supnotes/supnotes03/1s6.pdf> suffice the result here.

For the aesthetic score predictor $f_{aes} : \mathbb{R}^d \mapsto \mathbb{R}$, the adjoint optimization target is simply f_{aes} itself.

$$\mathcal{L}(\mathbf{x}_0) := f_{aes}(\mathbf{x}_0) \quad (54)$$

- Clip Loss

Following the implementation of (Pan et al., 2023a), we use the features from the CLIP image encoder as our feature vector. The loss function is L_2 -norm between the Gram matrix of the style image and the Gram matrix of the estimated clean image.

$$\mathcal{L}(\mathbf{x}_0) := \|\text{clip}(\mathbf{x}_0)\text{clip}(\mathbf{x}_0)^\top - \text{clip}(\mathbf{x}_{ref})\text{clip}(\mathbf{x}_{ref})^\top\| \quad (55)$$

- FaceID Loss

Following the implementation of (Pan et al., 2023b), we use ArcFace to extract the target features of reference faces to represent face IDs and compute the l_2 Euclidean distance between the extracted ID features of the estimated clean image and the reference face image as the loss function.

$$\mathcal{L}(\mathbf{x}_0) := \|\text{ArcFace}(\mathbf{x}_0) - \text{ArcFace}(\mathbf{x}_{ref})\| \quad (56)$$

Hyperparameters For the hyperparameters in adjoint-guided sampling, we ensure a fair comparison between the VJP and DF-EA methods. For most hyperparameters, we directly adopt the settings from previous works (Pan et al., 2023b) for both the baseline VJP method and our DF-EA method. For the guidance scale, we tune the value for the VJP method and use the same value for our DF-EA method. The DF-EA method does not introduce additional hyperparameters, and for mutual hyperparameters, our DF-EA method uses the exact same values as the VJP method. We adopt this strategy because our DF-EA method solely improves the approximation of the diffusion Fisher linear operator, without altering the adjoint sampling mechanism. Therefore, the suitable hyperparameters should remain unchanged, and we simply use the same parameters from the VJP method for our DF-EA method. For all experiments, we set the number of sampling steps to $T = 50$. Adjoint guidance is applied starting from steps ranging from 15 to 35 and ending at step 35, with one guidance per step. Thus the only parameter we tune is the guidance strength. We determine this value for the VJP method via a grid search from 0.1 to 0.5 with a step size of 0.1 and find that the optimal guidance strength for VJP is 0.2. We then use this value for our DF-EA method. Notice that, we apply a normalization to the guidance gradient for both VJP and DF-EA methods, making our optimal guidance strength consistent across different scores. The tuning is conducted on 1k COCO prompts, and the computational budget for tuning is $4 * 5 * 3$ GPU hours (4 tasks * 5 grids * 3 hours per single test) on Tesla V100 chips.

B.3. Numerical OT Experiments

The detailed algorithm The detailed version of the Algorithm 2 is presented in Algorithm 3. We use all the data points in the initial dataset to calculate every quantum we need in the sampling trajectory to validate the condition discussed in Corollary 1.

Initial data In Table 3, we adopt numerical verification of OT on 2-D synthesized data. For the single-Gaussian case, we fix the one mode central in (0.5,0.5) and test at $s = 0.1$; For the affine data case, we set three data points as (0.2,-0.4), (0.2,0.0), and (0.2,0.9), and test at $s = 0.0$; For the non-affine data case, we set three data points as (0.0,0.5), (0.0,0.0), and (0.5,0.0), and test at $s = 0.0$.

The definition of Asymmetric rate In Table 3, we need the asymmetry rate I to point out that the PF-ODE map in very simple non-affine data cannot be OT. Let A be an $n \times n$ matrix. The asymmetry rate I is defined as:

$$I = \frac{\|A - A^T\|_F}{\sqrt{2}\|A\|_F}$$

where $\|M\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n m_{ij}^2}$ is the Frobenius norm of the matrix M . The denominator $\sqrt{2}\|A\|_F$ is used to normalize the index to the range of $[0, 1]$. The asymmetry rate of the non-affine data in Table 3 has significantly deviated from zero.

Algorithm 3 Detailed numerical OT test for PF-ODE map

```

1: Input: initial data  $\{\mathbf{y}_j\}_{j=1}^N$ , noise schedule  $\{\alpha(t)\}$  and  $\{\sigma(t)\}$ , discretization steps  $M$ .
2: Initialize  $\mathbf{A}_M = \mathbf{I}$ ,  $\mathbf{x}_M \sim \mathcal{N}(0, \sigma_T \mathbf{I})$ .
3: for  $i = M, M-1, \dots, 1$  do
4:    $\alpha_i = \alpha(t_i)$ ,  $\sigma_i = \sigma(t_i)$ 
5:    $dt = t_{i-1} - t_i$ .
6:    $f_i = \frac{d \log \alpha_i}{dt}$ .
7:    $g_i = \sqrt{\frac{d\sigma_i^2}{dt} - 2 \frac{d \log \alpha_i}{dt} \sigma_i^2}$ .
8:   for  $j = 1, \dots, N$  do
9:     calculate  $v_j$  via  $v_j = \exp\left(-\frac{|\mathbf{x}_i - \alpha_i \mathbf{y}_j|^2}{2\sigma_i^2}\right)$ 
10:   end for
11:   for  $j = 1, \dots, N$  do
12:     calculate  $w_j$  via  $w_j = \frac{v_j}{\sum_k v_k}$ .
13:   end for
14:    $\mathbf{B}_i = \left[f_i - \frac{g_i^2}{2\sigma_i^2}\right] \mathbf{I} + \frac{\alpha_i^2 g_i^2}{2\sigma_i^4} \left[\sum_j w_j \mathbf{y}_j \mathbf{y}_j^\top - \left(\sum_j w_j \mathbf{y}_j\right) \left(\sum_j w_j \mathbf{y}_j\right)^\top\right]$ .
15:    $\mathbf{A}_{i-1} = \mathbf{A}_i + dt * \mathbf{A}_i^\top \mathbf{B}_i$  {solve fundamental matrix}
16:    $s_i = \frac{\mathbf{x}_i - \sum_j w_j \mathbf{y}_j}{\sigma_i^2}$ 
17:    $\mathbf{x}_{i-1} = \mathbf{x}_i + \left[f_i \mathbf{x}_i - \frac{1}{2} g_i^2 s_i\right] * dt$ 
18: end for
19: Output:  $\mathbf{A}_0$ .

```

B.4. Pretrained Models

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

- stable-diffusion-v1-5
<https://huggingface.co/runwayml/stable-diffusion-v1-5>
- stable-diffusion-2-base
<https://huggingface.co/stabilityai/stable-diffusion-2-base>
- SAC-aesthetic score predictor
<https://github.com/christophschuhmann/improved-aesthetic-predictor/blob/main/sac%2Blogos%2Baval-114-linearMSE.pth>
- AVA-aesthetic score predictor
<https://github.com/christophschuhmann/improved-aesthetic-predictor/blob/main/ava%2Blogos-114-linearMSE.pth>
- ArcFace ID loss
https://github.com/TreBleN/InsightFace_Pytorch
- Clip loss
<https://huggingface.co/openai/clip-vit-large-patch14>
- Pick-Score
<https://github.com/yuvalkirstain/PickScore>

C. Discussions

C.1. Singularity of Fisher information at $t = 0$

Previous studies (Yang et al., 2023; Zhang et al., 2024b) have shown that the diffusion model, particularly when learned in ϵ -prediction, can encounter a singularity issue at $t = 0$. Our DF in equation 11 reaffirms this issue, as this formulation becomes ill-formed at $t = 0$ due to division by zero (σ_0). Consequently, our formulation does not describe the behavior at $t = 0$. The deep theoretical exploration of the singularity problem remains an open question in the diffusion model field. However, as it is not the primary focus of this paper, we will not discuss the DF at $t = 0$.

C.2. Statistical Caliber of Negative Log-Likelihood

When dealing with high-dimensional data such as images, direct likelihood comparisons may encounter scaling issues due to the dimensionality. In this study, unless explicitly indicated otherwise, we adopt the approach of (Zheng et al., 2023) and typically use Negative Log-Likelihood (NLL) to refer to Bits Per Dimension (BPD).

$$\text{BPD} = \mathbb{E}_{\mathbf{x}_0 \sim q_0} \left[\frac{-\log P_0(\mathbf{x}_0)}{d \log 2} \right] \quad (57)$$

C.3. The ODE Solvers

To compute the numerical solutions for the PF-ODE in equation 5, the likelihood ODE in equation 14, and the adjoint ODE in equation 18, we require ODE solvers. In our paper’s experiments, we consistently use the explicit Euler method (referred to as DDIM when applied to PF-ODE). However, it’s important to note that our approach is not dependent on a specific ODE solver. We can also utilize alternatives like fast ODE solvers (Lu et al., 2022b;c; Liu et al., 2022) or exact inversion ODE solvers (Wallace et al., 2023; Zhang et al., 2023).

C.4. The relation of our outer-product form DF to Covariance in DMs

We note that there is a series of studies aiming to learn the covariance of the reverse diffusion Stochastic Differential Equation (SDE) (Bao et al., 2022b;a). In Bayesian statistics, Fisher information is defined as the covariance of the score. These studies derive their formulation by analyzing the covariance of the score and obtaining the Fisher information in terms of the score. However, our DF is derived directly from the marginal distribution and is composed solely of the initial distribution and noise schedule. Furthermore, our application is unique; we are the first to replace the use of VJP with DF, while the focus of these studies is to enhance the performance of Diffusion Models (DMs) with analytical covariance.

A very similar form of the Fisher information is proposed in Lemma 5 of (Benton et al., 2024). The difference is that our Proposition 3 presents the specific form of Fisher information in terms of data distribution, which is not included in Lemma 5 of (Benton et al., 2024). This distinction is crucial as it facilitates the derivation of our new algorithms. Also, we adopt the currently more commonly used α_t, σ_t notation to represent the noise schedule. Instead, (Benton et al., 2024) $\alpha_t \equiv e^{-2t}$. We gave out this detailed formulation to avoid unnecessary misunderstandings in the development of the subsequent training scheme.

C.5. Comparison of DF-TM to Hutchinson trace estimator

We notice that the naive trace calculation of the VJP method can be accelerated by the Hutchinson trace estimator. We have not conducted experiments using the Hutchinson estimator, as it is a Monte-Carlo type estimation, unlike the direct evaluation methods like the full-VJP baseline and our DF-TM. Furthermore, the VJP method, with the help of the Hutchinson method and its variants, is still considerably more expensive than our method, and its applicability is also more restricted.

- **The Hutchinson method is much more costly:**

To attain a relative error less than ϵ with a probability of $1 - \delta$, the Hutchinson method requires $\frac{2(1-\frac{8}{3}\epsilon) \log \frac{1}{\delta}}{\epsilon^2}$ samples (Skorski, 2021). Assuming that the goal is to obtain an estimation with a relative error of less than 10% with a probability exceeding 90%, a minimum of 351 NFEs is required. This equates to 1 or 2 minutes on SD-v1.5 for a single trace estimation. For a complete NLL estimation of an image with 20 steps, this would take 20 minutes, which is entirely impractical in any business context. In contrast, our DF-TM only requires 1 NFE for a single trace estimation, needing merely 10 seconds for the full NLL estimation of an image.

- **The application of the Hutchinson method is more restricted:**

Due to its Monte-Carlo characteristics, the Hutchinson method is more appropriate for contributing to the computation of certain training objectives, as in (Lu et al., 2022a), where the unbiased property is sufficient, and large variance may be absorbed into network training. However, the Hutchinson method may encounter difficulties with accurate per-sample trace estimation. Our method can accommodate both scenarios, including per-sample computation.

There are already attempts to use the Hutchinson method to expedite the VJP of trace estimation in diffusion models (Lu et al., 2022a). Nonetheless, due to Hutchinson’s limitations, these practices are restricted to relatively small DMs (CIFAR-10 at most). We will add discussions on the Hutchinson method in our revision.

C.6. Discussions on the theoretical bound of DF-EA

We notice that the error bound in Proposition 8 does not vanish as the training error decreases. From the rigorous theoretical perspective, considering the error bounds in Propositions 7 and 8, the DF-EA method is less valid than the DF-TM method, as we currently cannot establish vanishing bounds for it. However, from an empirical perspective, our DF-EA outperforms the naive VJP method in terms of score improvement across various tasks and pretrained models, as demonstrated in Figure 4. Thus, the DF-EA approximation proves to be valid in a practical sense. The replacement $\sum_i \mathbf{y}_i \mathbf{y}_i^\top \approx \mathbf{x}_0 \mathbf{x}_0^\top$ originates from the observation that w_i is a weighting of the summation equal to 1, and as t approaches 0, the w_i closest to $\mathbf{x}_0$ will dominate due to the diffusion kernel. Therefore, this approximation is intuitively reasonable near $t = 0$, which is precisely where we apply adjoint guidance.

C.7. Limitations

This paper does not explore the integration of DF into accelerated ODE solvers (Lu et al., 2022b), or exact inversion ODE samplers (Wang et al., 2024a). This paper is constrained in the scope of DMs, but similar second-order information may also exist in flow-matching generative models (Lipman et al., 2022; Zhu et al., 2024a) or variational inference models (Zhu et al., 2024b; Wang et al., 2024b). Our technique may also have downstream applications in automated driving (Tu et al., 2025b), touch-generation (Tu et al., 2025a), domain-transfer (Feng et al., 2024b;a), language generation (Fu et al., 2025), and missing data imputation (Chen et al., 2024). The DF may also contribute to a more effective inference method, like L-BFGS, which we did not explore.