

# Unifying Bayesian Flow Networks and Diffusion Models through Stochastic Differential Equations

Kaiwen Xue<sup>\*1</sup> Yuhao Zhou<sup>\*2</sup> Shen Nie<sup>1</sup> Xu Min<sup>3</sup> Xiaolu Zhang<sup>3</sup> Jun Zhou<sup>3</sup>  
Chongxuan Li<sup>1</sup>

## Abstract

Bayesian flow networks (BFNs) iteratively refine the parameters, instead of the samples in diffusion models (DMs), of distributions at various noise levels through Bayesian inference. Owing to its differentiable nature, BFNs are promising in modeling both continuous and discrete data, while simultaneously maintaining fast sampling capabilities. This paper aims to understand and enhance BFNs by connecting them with DMs through stochastic differential equations (SDEs). We identify the linear SDEs corresponding to the noise-addition processes in BFNs, demonstrate that BFN’s regression losses are aligned with denoise score matching, and validate the sampler in BFN as a first-order solver for the respective reverse-time SDE. Based on these findings and existing recipes of fast sampling in DMs, we propose specialized solvers for BFNs that markedly surpass the original BFN sampler in terms of sample quality with a limited number of function evaluations (e.g., 10) on both image and text datasets. Notably, our best sampler achieves an increase in speed of  $5 \sim 20$  times for free. Our code is available at <https://github.com/ML-GSAI/BFN-Solver>.

## 1. Introduction

Deep generative models (DGMs) are effective in capturing complex data distributions and producing realistic samples, substantially influencing fields such as computer vision (Rombach et al., 2022; Ramesh et al., 2022; Podell et al., 2023) and natural language processing (Brown et al., 2020; OpenAI, 2023). The fundamental challenge in DGMs

is to represent a flexible probability distribution that facilitates effective parameter learning and efficient inference simultaneously, greatly depending on the data (or modality).

Autoregressive models (ARMs) (OpenAI, 2023), for example, excel in modeling sequential and discrete data (e.g., text) but face limitations in the inference speed, which is proportional to the number of variables. Diffusion models (DMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021), on the other hand, better balance generation quality and efficiency with a coarse-to-fine approach. Although considered state-of-the-art in image generation, DMs encounter challenges in handling discrete variables, where score matching algorithms (Hyvärinen, 2005; Vincent, 2011) do not directly apply.

A new class of generative models, Bayesian Flow Networks (BFNs) (Graves et al., 2023), has been developed recently to overcome these challenges. While inspired by DMs, BFNs distinguish themselves by focusing on iteratively refining the parameters (instead of the samples) of a distribution set at different noise levels through Bayesian inference (see Sec. 3 for more details). This strategy enables BFNs to facilitate fast sampling and maintain a continuous nature, even when processing discrete data. With carefully designed regression losses, BFNs have shown considerable promise in both image and language modeling. Notably, BFN is primarily developed based on a message-sending process with minimum communication length, and the exact relation between BFNs and DMs remains unclear.

As summarized in Table 1, this paper primarily contributes by unifying BFNs and DMs through stochastic differential equations (SDEs), a pivotal step in understanding their relationship and enhancing BFNs. Initially, by slightly truncating the time, we identify linear SDEs corresponding to the noise-adding processes in BFN on both continuous (see Sec. 4) and discrete data (see Sec. 5) and derive the reverse-time SDEs for sampling. Note that the SDEs for discrete data operate on a set of latent variables, which the original BFN formulation marginalizes out, rather than distribution parameters. Furthermore, we demonstrate that, especially on discrete data, BFN’s regression losses align with denoising score matching (DSM) (Vincent, 2011) w.r.t. variables

<sup>\*</sup>Equal contribution <sup>1</sup>Gaoling School of AI, Renmin University of China, Beijing, China <sup>2</sup>Department of Computer Science and Technology, Tsinghua University, Beijing, China <sup>3</sup>Ant Group, Hangzhou, China. Correspondence to: Chongxuan Li <chongxuanli@ruc.edu.cn>.

Table 1. Technical contributions of the paper include the theory on unifying BFN and DM (in red) and new samplers for BFN inspired by the theory (in blue). “SDE-solver1” means a first-order solver for the corresponding SDE and “Approx.” is a shorthand for “Approximate”.

	NOISE-ADDING PROCESS	LOSS FUNCTION	ORIGINAL SAMPLER	NEW SAMPLERS
BFN ON CONTINUOUS DATA	CORRESPONDING SDE <b>THEOREM 4.1</b>	EQUIVALENT TO DSM TRIVIAL	SDE-SOLVER1 <b>PROPOSITION 4.2</b>	BFN-SOLVERS <b>ALGOS. 1-3 IN APPENDIX</b>
BFN ON DIS-CREATE DATA	CORRESPONDING SDE <b>THEOREM 5.1</b>	EQUIVALENT TO DSM <b>THEOREM 5.2</b>	APPROX. SDE-SOLVER1 <b>PROPOSITION 5.3</b>	BFN-SOLVERS <b>ALGOS. 4-7 IN APPENDIX</b>

in the corresponding SDE, positioning the trained networks to naturally parameterize the reverse-time SDEs. Finally, the original BFN sampler is proven as an (approximate) first-order solver for the corresponding reverse-time SDE.

The explicit connection between BFNs and DMs brings immediate benefits, particularly in applying fast sampling methods (Lu et al., 2022b;c) from DMs to BFNs. We derive the corresponding probability flow ordinary differential equations (ODEs) (Song et al., 2021) for BFNs on both continuous and discrete data. We propose high-order solvers (named *BFN-Solvers*) tailored to BFNs’ special (e.g., semi-linear) structure, for both SDEs and ODEs. Empirically, using the same pre-trained model, our best solver significantly outperforms the original BFN sampler with a few (e.g., 10) number of function evaluations (NFE) under sample quality on both the CIFAR10 and text8 datasets, achieving a 5 ~ 20 times increase in speed for free (see Sec. 6 for details).

We believe our discovery offers a rigorous and systematic perspective for analyzing and improving the training and inference processes of BFNs, grounded in the existing results of DMs, and may inspire future work as detailed in Sec. 7.

## 2. Related Work

**Score-baese DMs.** Built upon the score matching algorithms (Hyvärinen, 2005; Vincent, 2011; Song et al., 2019; Pang et al., 2020), DMs (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) are currently SOTA to model continuous variables (Dhariwal & Nichol, 2021; Chen et al., 2020; Kong et al., 2020; Ho et al., 2022; Singer et al., 2022; Poole et al., 2022; Wang et al., 2023). In particular, large-scale text-to-image models (Rombach et al., 2022; Ramesh et al., 2022; Saharia et al., 2022; Bao et al., 2023; Balaji et al., 2023; Xue et al., 2023b; Podell et al., 2023) have made remarkable progress and attracted significant attention.

**Solvers for DMs.** Since Song et al. (2021) introduced the SDE and probability flow ODE formulation of DMs, there have been extensive solvers for both SDE (Ho et al., 2020; Song et al., 2021; Karras et al., 2022; Lu et al., 2022c; Bao et al., 2022b;a; Jolicœur-Martineau et al., 2021; Xue et al., 2023a; Guo et al., 2023) and ODE (Song et al., 2020; Liu

et al., 2022; Lu et al., 2022b;c; Zhang et al., 2022; Karras et al., 2022; Zhao et al., 2023) to improve the sampling process. In particular, ODE samplers are proven effective with limited NFEs while SDE samplers are robust to prior mismatch (Lu et al., 2022a; Nie et al., 2023) and perform better in a sufficient number of NFEs (Lu et al., 2022c).

**Discrete DMs.** Several DMs have been proposed to model discrete data with discrete states (Sohl-Dickstein et al., 2015; Hoogeboom et al., 2021; Austin et al., 2023), depending on a probability transition matrix. It is nontrivial to leverage the features associated with continuous-state DMs, such as guidance and ODE fast sampling. Efforts have been made to define the score in the discrete state (Lou et al., 2023; Meng et al., 2023; Campbell et al., 2022; Sun et al., 2023); however, this remains a challenging endeavor. Other works (Chen et al., 2022; Dieleman et al., 2022; Li et al., 2022) have attempted to identify a continuous equivalent for discrete data and apply continuous DMs, but this may result in information loss during the transformation and greatly rely on the noise schedule (Ye et al., 2023). Mahabadi et al. (2023) defines a continuous-time diffusion process on continuous latent variables but is trained with cross-entropy loss rather than regression loss. Several studies (Richemond et al., 2022; Lou & Ermon, 2023) have attempted to establish the diffusion process using SDEs on discrete data. Specifically, Richemond et al. (2022) introduced an SDE defined on the probability simplex, but it suffers from intractability in high-dimensional space. Lou & Ermon (2023) proposed a diffusion SDE with an additional boundary constraint, which also increases the complexity of discretization (e.g., requiring thresholding in SDE).

In comparison, this paper reveals that BFNs applied to discrete data solve a linear SDE and are trained using DSM, which aligns seamlessly with continuous DMs. Consequently, without changing the discrete data, BFNs are significantly simpler and more scalable and efficient than the related work, leveraging advancements in continuous DMs.

## 3. Background

In this section, we present the elementary notations and background of DMs and BFNs.

### 3.1. Elementary Notations

We use lowercase letters (e.g.,  $t$ ) and boldface lowercase letters (e.g.,  $\mathbf{x}$ ) to denote scalars and vectors respectively. Variables indexed by uncountable indices are denoted in the form of functions, (e.g.,  $\beta(t)$  and  $\boldsymbol{\mu}(t)$ ). Given finite indices (e.g.,  $\{t_i\}_{i=1}^M$ ), the corresponding variables are denoted with subscripts (e.g.,  $\boldsymbol{\mu}_i$ ).

### 3.2. Score-based DMs

Score-based DMs (Kingma et al., 2021) characterize the data distribution through a diffusion process  $\{\mathbf{x}(t) \sim \mathcal{N}(\alpha(t)\mathbf{x}, \sigma^2(t)\mathbf{I})\}$  indexed by a continuous-time variable  $t \in [0, T]$  according to an Itô SDE as follows

$$d\mathbf{x} = f(t)\mathbf{x}dt + g(t)d\mathbf{w}, \quad (1)$$

where  $\mathbf{w}$  is the standard Wiener process, and  $f(t) = \frac{d \log \alpha(t)}{dt}$  and  $g(t) = \frac{d\sigma^2(t)}{dt} - \frac{1}{2} \frac{d \log \alpha(t)}{dt} \sigma^2(t)$  are the drift and diffusion coefficients respectively. For instance, denoising diffusion probabilistic models (Ho et al., 2020) consider a process given by the following SDE:

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}, \quad (2)$$

where  $0 < \beta(t) < 1$ . Let  $p_t(\mathbf{x})$  denote the marginal density of  $\mathbf{x}(t)$ . The generative process of score-based DMs is given by a reverse-time SDE (Song et al., 2021; Anderson, 1982)

$$d\mathbf{x} = [f(t)\mathbf{x} - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})]dt + g(t)d\bar{\mathbf{w}}, \quad (3)$$

where  $\bar{\mathbf{w}}$  is the time-reversed Wiener process. Then the score is parameterized with a time-dependent score-based model  $\hat{\mathbf{s}}(\mathbf{x}, t)$  and trained with the following denoising score matching loss (Vincent, 2011)

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}_{\mathbf{x}(t), \mathbf{x}(0)} [\|\hat{\mathbf{s}}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}} \log p_{0t}(\mathbf{x}(t)|\mathbf{x}(0))\|_2^2], \quad (4)$$

where the conditional distribution  $p_{0t}(\mathbf{x}|\mathbf{x}(0))$  is designed as a Gaussian kernel with a closed form score function  $\nabla_{\mathbf{x}} \log p_{0t}(\mathbf{x}|\mathbf{x}(0))$ . For fast sampling, Song et al. (2021) introduce the corresponding *probability flow ODE* of the reverse SDE in Eq. (3) as follows

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

which produces the same data distribution as the corresponding SDE with infinitesimally small stepsize and enjoys a smaller discretization error with a large stepsize due to its deterministic nature (Kloeden et al., 1992). To solve the ODE in Eq. (5) efficiently, DPM-Solvers (Lu et al., 2022b;c) explicitly leverage the semi-linear property of Eq. (5) and further simplify it to an exponentially weighted integral of

the neural network by applying change-of-variable. Consequently, the exact solution of ODE is given by

$$\mathbf{x}(t) = \frac{\alpha(t)}{\alpha(s)}\mathbf{x}(s) - \alpha(t) \int_{\lambda(s)}^{\lambda(t)} e^{-\lambda} \hat{\mathbf{e}}_{\theta}(\hat{\mathbf{x}}(\lambda), \lambda) d\lambda, \quad (6)$$

where  $\lambda(t) = \log(\alpha(t)/\sigma(t))$  is the half of the log signal-noise ratio. DPM-Solver solves Eq. (6) numerically leading to a small discretization error. Taking DPM-Solver1 as an example, given time steps  $\{t_i\}_{i=1}^n$  and initial value  $\mathbf{x}_0$ , a sequence  $\{\mathbf{x}_i\}_{i=1}^n$  can be solved iteratively as follows:

$$\mathbf{x}_i = \frac{\alpha(t_i)}{\alpha(t_{i-1})}\mathbf{x}_{i-1} - \sigma(t_i)(e^{h_i} - 1)\hat{\mathbf{e}}_{\theta}(\mathbf{x}_{i-1}, t_{i-1}) + \mathcal{O}(h_i^2), \quad (7)$$

where  $h_i = \lambda(t_i) - \lambda(t_{i-1})$ . Empirically, DPM-Solver achieves excellent results with a limited number of NFEs and is widely adopted.

### 3.3. Bayesian Flow Networks

Due to the space limit, we briefly present the motivation and formulation of BFNs (Graves et al., 2023) here and please refer to the original paper for more details. Inspired by DMs, BFNs iteratively refine the parameters of a distribution set at different noise levels through Bayesian inference. This strategy enables BFNs to facilitate fast sampling and be differentiable on both continuous and discrete data.

For  $D$ -dimensional continuous data<sup>1</sup>  $\mathbf{x} \in \mathbb{R}^D$ , a continuous-time BFN operates on parameters of a set of Gaussian distributions (of noisy data with different noise levels) with means  $\{\boldsymbol{\mu}(t)\}_{t=0}^1$  and covariance matrices  $\{\rho(t)\mathbf{I}\}_{t=0}^1$ . Equivalently,  $\boldsymbol{\mu}(t)$  can also be regarded as a noisy version of  $\mathbf{x}$  by injecting a Gaussian noise and follows the distribution

$$q_{\mathbf{F}}(\boldsymbol{\mu}(t)|\mathbf{x}, \gamma(t)) = \mathcal{N}(\gamma(t)\mathbf{x}, \gamma(t)(1 - \gamma(t))\mathbf{I}), \quad (8)$$

where  $\gamma(t) = 1 - \sigma_1^{2(1-t)}$  is a schedule function<sup>2</sup> and  $\sigma_1 \in (0, 1)$  is a hyperparameter.  $\rho(t)$  has a closed form as  $\rho(t) = \frac{1}{1-\gamma(t)}$ . Similar to DMs, a BFN on continuous data trains a neural network  $\hat{\mathbf{e}}(\boldsymbol{\mu}(t), t)$  to predict the injected Gaussian noise  $\epsilon$  by minimizing the following loss:

$$q_{\mathbf{F}}(\boldsymbol{\mu}(t)|\mathbf{x}, \gamma(t)), t \sim U(0, 1) \quad \mathbb{E} \left[ -\frac{\ln \sigma_1}{\sigma_1^{2t}} \|\epsilon - \hat{\mathbf{e}}(\boldsymbol{\mu}(t), t)\|^2 \right]. \quad (9)$$

Given time steps  $\{t_i\}_{i=0}^n$  and i.i.d. noises  $\{\mathbf{u}_i\}_{i=0}^n \sim \mathcal{N}(0, \mathbf{I})$ , the BFN sampler (Graves et al., 2023) iterates

<sup>1</sup>We say  $\mathbf{x}$  is a continuous data if its distribution has density w.r.t. the Lebesgue measure.

<sup>2</sup>For a clear alignment with DMs, we adopt a reverse time notation in this paper as originally used by Graves et al. (2023). Specifically, the schedule  $\gamma(t)$  in our paper is equivalent to  $\gamma(1-t)$  in Graves et al. (2023). We retain the other notational conventions for ease of reading, which do not affect our derivations.

as follows.

$$\mu_i = -\frac{\gamma(t_i) - \gamma(t_{i-1})}{\sqrt{\gamma(t_{i-1})(1 - \gamma(t_{i-1}))}} \hat{\epsilon}(\mu_{i-1}, t_{i-1}) + \frac{\gamma(t_i)}{\gamma(t_{i-1})} \mu_{i-1} + \sqrt{\frac{1 - \gamma(t_i)}{1 - \gamma(t_{i-1})}} (\gamma(t_i) - \gamma(t_{i-1})) \mathbf{u}_i. \quad (10)$$

On  $D$ -dimensional discrete data  $\mathbf{x} \in \{1, \dots, K\}^D$ , where  $K$  is the number of classes, the BFN operates on parameters  $\theta(t)$  of the multivariate categorical distributions of noisy data. The distribution of  $\theta$  is

$$q_F(\theta(t) | \mathbf{x}, \beta(t)) = \mathbb{E}_{q(\mathbf{z}(t) | \mathbf{x}, \beta(t))} \delta(\theta(t) - \text{softmax}(\mathbf{z}(t))),$$

where  $\delta(\cdot)$  is the Dirac distribution,  $\mathbf{z}(t)$  is a set of latent variables with Gaussian marginal distributions as

$$q(\mathbf{z}(t) | \mathbf{x}, \beta(t)) = \mathcal{N}(\beta(t) \mathbf{w}_x, K\beta(t) \mathbf{I}), \quad (11)$$

and  $\mathbf{w}_x := K\mathbf{e}_x - \mathbf{1}$ ,  $\mathbf{e}_x := \{e_{x^{(1)}}, \dots, e_{x^{(D)}}\} \in \mathbb{R}^{KD}$  where  $e_j$  is the one-hot vector defined by  $(e_j)_k = \delta_{x_j k}$  and  $\mathbf{1}$  is a vector of length  $KD$  filled with ones.  $\beta(t) = (1 - t)^2 \beta_1$  is a schedule function with a hyperparameter  $\beta_1 > 0$ . A BFN on discrete data trains a neural network  $\hat{\epsilon}(\theta(t), t)$  that predicts the data in a one-hot form given noisy inputs using the following regression loss

$$\mathcal{L}^\infty(\mathbf{x}) = \mathbb{E}_{q_F(\theta | \mathbf{x}, t), t \sim U(0, 1)} K\beta_1 t \|\mathbf{e}_x - \hat{\epsilon}(\theta(t), t)\|^2. \quad (12)$$

Let  $\{\mathbf{u}_i\}_{i=0}^n \sim \mathcal{N}(0, \mathbf{I})$  be independent and use  $\hat{\epsilon}_s(\mathbf{z}(t), t)$  as a shorthand for  $\hat{\epsilon}(\text{softmax}(\mathbf{z}(t)), t)$ . The sampling rule of BFN (Graves et al., 2023) can be written as follows

$$e_k \sim \text{Cat}(\hat{\epsilon}_s(\mathbf{z}_{i-1}, t_{i-1})), \quad (13)$$

$$\mathbf{z}_i = \mathbf{z}_{i-1} + \alpha_i (K\mathbf{e}_k - \mathbf{1}) + \sqrt{K\alpha_i} \mathbf{u}_i, \quad (14)$$

where  $\alpha_i = \beta(t_i) - \beta(t_{i-1})$  and Cat represents the one-hot categorical distribution.<sup>3</sup>

Based on the formulation, BFNs have shown considerable promise in both image and language modeling. Although inspired by DMs, and the exact relation between BFNs and DMs remains unclear. To this end, this paper unifies them through stochastic differential equations (SDEs) for understanding and accelerating BFNs on both continuous data (see Sec. 4) and discrete data (see Sec. 5).

## 4. Continuous-time BFN on Continuous Data

This section bridges BFNs on continuous data with DMs by establishing a linear SDE for noise modeling in BFN

<sup>3</sup>Originally, Graves et al. (2023) obtain samples through  $\theta(t)$ , while we present the equivalent form in terms of  $\mathbf{z}(t)$  for convenience.

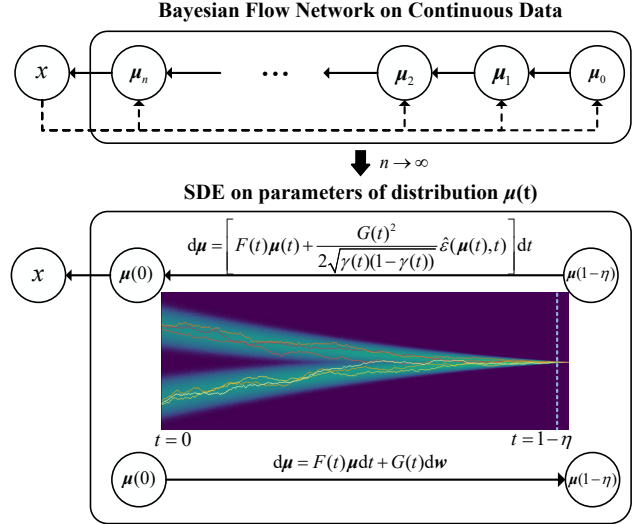


Figure 1. Illustration of BFN on continuous data and the corresponding SDEs. The SDEs are defined w.r.t.  $\mu$  on time  $[0, 1 - \eta]$ .

(Sec. 4.1), aligning training objectives with DSM (Sec. 4.2), and validating the sampler as discretization of the reverse-time SDE (Sec. 4.3). Further, fast samplers are developed based on the recipe in DMs in Sec. 4.4.

### 4.1. Formulating BFN on Continuous Data as SDEs

As illustrated in Fig. 1, we establish that the (truncated) noise-adding process of the continuous-time BFN on continuous data in Eq. (8) uniquely solves a linear SDE, summarized as follows.

**Theorem 4.1** (Proof in Appendix A.1). *Let  $\eta > 0$  be an arbitrarily small constant. The BFN in Eq. (8) at time  $[0, 1 - \eta]$  is the unique solution of the following linear SDE:*

$$d\mu = F(t)\mu dt + G(t)dw. \quad (15)$$

Here  $w$  is a standard Wiener process and

$$F(t) = \frac{\gamma'(t)}{\gamma(t)} = 2 \frac{\sigma_1^{2(1-t)}}{1 - \sigma_1^{2(1-t)}} \ln \sigma_1, \quad (16)$$

$$G(t)^2 = -\gamma'(t) = -2\sigma_1^{2(1-t)} \ln \sigma_1, \quad (17)$$

where  $\sigma_1 \in (0, 1)$  is the hyperparameter defined in Eq. (8).

The time  $t$  is truncated by  $1 - \eta$  in Theorem 4.1 for two reasons. On one hand, the reverse-time SDE derived later (see Eq. (20)) is ill-defined at  $t = 1$  since the distribution of  $\mu$  collapses to a Dirac distribution whose score tends to infinity. On the other hand, it is convenient to satisfy certain regularity conditions for the uniqueness of the solution in Theorem 4.1, as detailed in the proof. As  $\eta$  is small (e.g.,  $10^{-3} \sim 10^{-5}$ ) in our implementation, the effect of truncation is negligible. The exact distribution of  $\mu(1 - \eta)$  is



unknown, we approximate it by an isotropic Gaussian with a zero mean and small variance (see details in Sec. 6).

Here a linear SDE also applies to the latent variable  $\mathbf{z}$ , a linear transformation of  $\boldsymbol{\mu}$  in Eq. (8). The choice of  $\boldsymbol{\mu}$  in Theorem 4.1 aligns with the sampling process in BFN (Graves et al., 2023), facilitating a later analysis in Sec. 4.3.

The finding in Theorem 4.1 directly connects to DMs (Song et al., 2021; Kingma et al., 2021), which are formulated as an SDE in Eq. (1) with a different noise schedule. We believe this may inspire new classes of BFNs and leave a systematic comparison of the schedules for future work.

Similar to Eq. (3), the linear SDE in Eq. (15) has an associated reverse-time SDE (Anderson, 1982; Song et al., 2021) in  $[0, 1 - \eta]$  for generative modeling:

$$d\boldsymbol{\mu} = [F(t)\boldsymbol{\mu} - G(t)^2 \nabla_{\boldsymbol{\mu}} \log p_t(\boldsymbol{\mu})] dt + G(t) d\bar{\mathbf{w}}, \quad (18)$$

where  $\nabla_{\boldsymbol{\mu}} \log p_t(\boldsymbol{\mu})$  is the (time-conditional) score function to be estimated and  $\bar{\mathbf{w}}$  is the time-reversed Wiener process.

## 4.2. Training as Parameterizing the Reverse-time SDE

The continuous-time BFN on continuous data trains a neural network to optimize the mean square error in Eq. (9), which directly aligns with the widely employed DSM loss in Eq. (4). In other words, BFN equivalently parameterizes the reverse-time SDE in Eq. (18) by estimating the time-conditional score function as

$$\hat{\mathbf{s}}(\boldsymbol{\mu}(t), t) = -\frac{1}{\sqrt{\gamma(t)(1 - \gamma(t))}} \hat{\boldsymbol{\epsilon}}(\boldsymbol{\mu}(t), t), \quad (19)$$

where  $\hat{\mathbf{s}}(\boldsymbol{\mu}(t), t)$  and  $\hat{\boldsymbol{\epsilon}}(\boldsymbol{\mu}(t), t)$  denote the estimate of the score function and the network trained by BFN, respectively, and  $\gamma(t)$  follows Eq. (8).

## 4.3. Sampling as Discretizing the Reverse-time SDE

Plugging Eq. (19) into Eq. (18), we get a parameterized reverse-time SDE in  $[0, 1 - \eta]$  for sampling as follows

$$d\boldsymbol{\mu} = \left[ F(t)\boldsymbol{\mu}(t) + \frac{G(t)^2 \hat{\boldsymbol{\epsilon}}(\boldsymbol{\mu}(t), t)}{\sqrt{\gamma(t)(1 - \gamma(t))}} \right] dt + G(t) d\bar{\mathbf{w}}, \quad (20)$$

which is ill-defined at  $t = 1$  because  $\lim_{t \rightarrow 1} \gamma(t) = 1$ . Interestingly, even without an explicit SDE formulation, the sampler proposed in the original BFN paper discretizes the reverse-time SDE, as characterized in the following Proposition 4.2.

**Proposition 4.2** (Proof in Appendix A.2). *The BFN sampler in Eq. (10) is a first-order discretization of an equivalent form of the parameterized reverse-time SDE in Eq. (20).*

## 4.4. Probability Flow ODE and Faster Sampling

Establishing an explicit connection between BFNs and DMs through SDEs yields an immediate and significant benefit: the fast sampling recipe from DMs directly applies to BFN.

Formally, according to Eq. (5), we obtain the following equivalent probability flow ODE of the parameterized reverse-time SDE of Eq. (20):

$$d\boldsymbol{\mu} = \left[ F(t)\boldsymbol{\mu}(t) + \frac{G(t)^2}{2\sqrt{\gamma(t)(1 - \gamma(t))}} \hat{\boldsymbol{\epsilon}}(\boldsymbol{\mu}(t), t) \right] dt. \quad (21)$$

Further, we propose *BFN-Solver*, a customized ODE solver for BFN in analogy to DPM-Solver in Eq. (7). As detailed in Appendix A.3, we integrate all linear terms and apply a change of variable from  $t$  to  $\lambda(t) = \frac{1}{2} \log \frac{\gamma(t)}{1 - \gamma(t)}$  to obtain a simplified exact solution of Eq. (21)

$$\boldsymbol{\mu}(t) = \frac{\gamma(t)}{\gamma(s)} \boldsymbol{\mu}(s) - \gamma(t) \int_{\lambda(s)}^{\lambda(t)} e^{-\lambda} \hat{\boldsymbol{\epsilon}}(\boldsymbol{\mu}(t_\lambda(\lambda)), t_\lambda(\lambda)) d\lambda, \quad (22)$$

where  $t_\lambda(\cdot)$  is the inverse function of  $\lambda(t)$  for  $0 \leq t < s < 1 - \eta$ . Eq. (22) differs from Eq. (6) only in certain coefficients. Given an initial value  $\boldsymbol{\mu}_0$  and time steps  $\{t_i\}_{i=0}^n$  from  $t_0 = 1 - \eta$  to  $t_n = 0$ , BFN-Solver1 is derived similarly to Eq. (7) and given by

$$\begin{aligned} \boldsymbol{\mu}_i = & -\sqrt{\gamma(t_i)(1 - \gamma(t_i))} (e^{h_i} - 1) \hat{\boldsymbol{\epsilon}}(\boldsymbol{\mu}_{i-1}, t_{i-1}) \\ & + \frac{\gamma(t_i)}{\gamma(t_{i-1})} \boldsymbol{\mu}_{i-1}, \end{aligned} \quad (23)$$

where  $h_i = \lambda(t_i) - \lambda(t_{i-1})$ . We refer the readers to Appendix A.3 for higher-order solvers of both ODE and SDE.<sup>4</sup>

Empirically, as presented in Sec. 6.2, BFN-Solvers of different orders significantly outperform the original BFN sampler with a limited number of NFEs based on the same model.

## 5. Continuous-time BFN on Discrete Data

In a manner akin to Sec. 4, this section unifies BFNs on discrete data and (continuous) DMs through SDEs and develops fast samplers for BFNs. However, this adaptation to discrete data is far from straightforward, as it involves SDEs operating on latent variables  $\mathbf{z}$  — a significant departure from the original BFN formulation that marginalizes out these variables, rather than updating the distribution parameters  $\boldsymbol{\theta}$ . Consequently, it is surprising that the training and

<sup>4</sup>A more straightforward way to get BFN-Solver on continuous data is to treat BFN as a DM with a special noise schedule  $\alpha_t = \gamma_t$  and  $\sigma_t^2 = \gamma_t(1 - \gamma_t)$ . However, it is infeasible on discrete data. Therefore, we use a slightly complex yet coherent way to derive BFN-Solver throughout the paper.

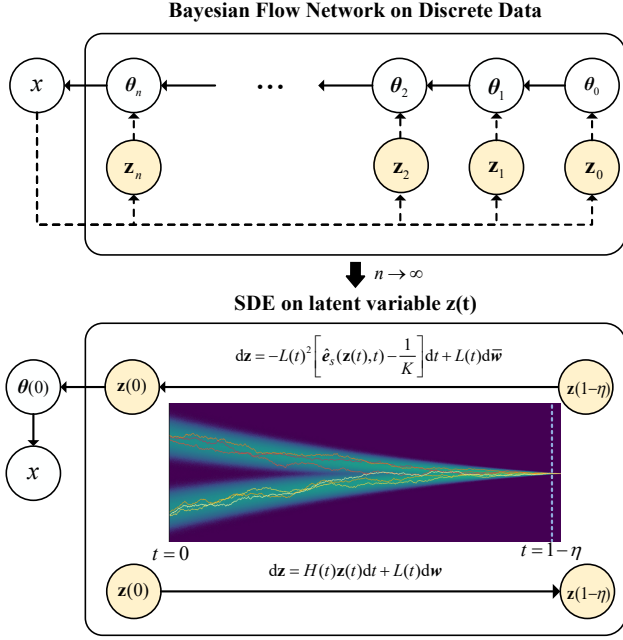


Figure 2. **Illustration of BFN on discrete data and the corresponding SDEs.** The SDEs are defined w.r.t. the latent variables  $\mathbf{z}$ , which are marginalized in BFN, on time  $[0, 1 - \eta]$ .

sampling of BFN on discrete data still connect to the SDE formulation on  $\mathbf{z}$ .

### 5.1. Formulating BFN on Discrete Data as SDEs

Similar to Theorem 4.1, the truncated noise-adding process of the continuous-time BFN on discrete data in Eq. (11) uniquely solves a linear SDE, summarized as follows.

**Theorem 5.1** (Proof in Appendix B.1). *Let  $\eta > 0$  be an arbitrarily small constant. The BFN in Eq. (11) with  $t \in [0, 1 - \eta]$  is the unique solution of the following linear SDE:*

$$d\mathbf{z} = H(t)\mathbf{z} dt + L(t) d\mathbf{w}. \quad (24)$$

Here  $\mathbf{w}$  is a standard Wiener process and

$$H(t) = \frac{\beta'(t)}{\beta(t)} = -\frac{2}{1-t}, \quad (25)$$

$$L(t)^2 = -K\beta'(t) = 2K\beta_1(1-t), \quad (26)$$

where  $K$  and  $\beta_1$  are hyperparameters defined in Eq. (11).

The rationale for truncation of  $t$  and the way to deal with  $\eta$  and  $\mathbf{z}(1 - \eta)$  is similar to the continuous data case, detailed in the proof and Sec. 6.1, respectively.

Notably, Theorem 5.1 characterizes the dynamics of  $\mathbf{z}$  instead of  $\theta$ , as illustrated in Fig. 2. Indeed, the dynamics of  $\theta$  do not correspond to a linear SDE as  $\theta$  is a nonlinear transformation of  $\mathbf{z}$  as shown in Eq. (11). It is implied that

the original sampling process in Eq. (14) does not directly discretize the linear SDE, as detailed in Sec. 5.3.

The associated reverse-time SDE (Song et al., 2021) for the linear SDE in Eq. (24) in  $[0, 1 - \eta]$  is given by

$$d\mathbf{z} = [H(t)\mathbf{z} - L(t)^2 \nabla_{\mathbf{z}} \log p_t(\mathbf{z})] dt + L(t) d\bar{\mathbf{w}}, \quad (27)$$

where  $\nabla_{\mathbf{z}} \log p_t(\mathbf{z})$  is the unknown score function, defined on  $\mathbf{z}$  instead of  $\theta$ .

### 5.2. Training as Parameterizing the Reverse-time SDE

It is nontrivial to see yet can be proven that the training objective of the continuous-time BFN on discrete data in Eq. (12) is a reparameterized form of DSM (Vincent, 2011) w.r.t.  $\mathbf{z}$ , as summarized in the following Theorem 5.2.

**Theorem 5.2** (Proof in Appendix B.2). *Minimizing the continuous-time loss of BFN on discrete data in Eq. (12) is equivalent to minimizing the DSM loss in Eq. (4). Besides, the corresponding estimate of the score function is given by*

$$\hat{s}(\mathbf{z}(t), t) = -\frac{\mathbf{z}(t)}{K\beta(t)} + \hat{e}_s(\mathbf{z}(t), t) - \frac{1}{K}, \quad (28)$$

where  $\hat{e}_s(\mathbf{z}(t), t)$  is the network trained by BFN.

Theorem 5.1 and Theorem 5.2 distinct BFNs from existing discrete DMs. Specifically, BFNs applied to discrete data solve a linear SDE and are trained using DSM, which aligns seamlessly with continuous DMs. Consequently, without changing the discrete data, BFNs are significantly simpler and more scalable and efficient than the related work, leveraging advancements in continuous DMs. We provide a comprehensive review and discussion in Sec. 2.

### 5.3. Sampling as Discretizing the Reverse-time SDE

Plugging Eq. (28) into Eq. (27), we get a parameterized reverse-time SDE in  $[0, 1 - \eta]$  for sampling as follows

$$d\mathbf{z} = -L(t)^2 \left[ \hat{e}_s(\mathbf{z}(t), t) - \frac{1}{K} \right] dt + L(t) d\bar{\mathbf{w}}. \quad (29)$$

The following Proposition 4.2 suggests that the sampler proposed in the original BFN paper approximately discretizes the parameterized reverse-time SDE.

**Proposition 5.3** (Proof in Appendix B.3). *If the categorical sampling step in the BFN sampler on discrete data (i.e., Eq. (13)) is omitted, then it is a first-order discretization of the parameterized reverse-time SDE in Eq. (29).*

The role of the categorical sampling step is still unclear in theory. However, experiments in Fig. 6 (Sec. 6.3) reveal that removing the categorical sampling step leads to consistently better performance in fewer than 50 NFes, and almost the same results otherwise.

#### 5.4. Probability Flow ODE and Faster Sampling

Similar to the continuous case, the equivalent probability flow ODE of the parameterized reverse-time SDE on discrete data in Eq. (29) is

$$d\mathbf{z} = \left\{ -\frac{1}{1-t}\mathbf{z}(t) - \beta_1(1-t)[K\hat{e}_s(\mathbf{z}(t), t) - 1] \right\} dt. \quad (30)$$

For  $0 \leq t < s < 1 - \eta$ , its solution can be written as

$$\begin{aligned} \mathbf{z}(t) = & \frac{1-t}{1-s}\mathbf{z}(s) + \beta_1(1-t)(t-s) \\ & - K\beta_1(1-t) \int_s^t \hat{e}_s(\mathbf{z}(\tau), \tau) d\tau. \end{aligned} \quad (31)$$

Again, we propose BFN-Solver on discrete data, and the first-order version is given by

$$\begin{aligned} \mathbf{z}_i = & \beta_1(1-t_i)(t_i - t_{i-1})(1 - K\hat{e}_s(\mathbf{z}(t_{i-1}), t_{i-1})) \\ & + \frac{1-t_i}{1-t_{i-1}}\mathbf{z}_{i-1}. \end{aligned} \quad (32)$$

Notably, we map the latent  $\mathbf{z}_M$  to the distribution parameter  $\theta_M = \text{softmax}(\mathbf{z}_M)$  at the last step to obtain the final samples. We refer the readers to Appendix B.5 for higher-order solvers of both ODE and SDE. As presented in Sec. 6.3, the conclusion on the improvement of BFN-Solvers over the original BFN sampler remains the same on discrete data.

## 6. Experiments

We present the experimental setups in Sec. 6.1. We validate the proposed BFN-Solvers on continuous and discrete data, in Sec. 6.2 and Sec. 6.3 respectively.

### 6.1. Experimental Settings

**Model.** We employed the pre-trained models provided by the BFN (Graves et al., 2023) in all experiments for fairness.

**Datasets.** For continuous data, the model is trained on the CIFAR-10 (Krizhevsky et al., 2009) dataset which contain 50K training images. For discrete data, the model is trained on the text8 (Mahoney, 2011) dataset which contains 90M consecutive characters, each character is a lower Latin letter ‘a’-‘z’ or the ‘\_’ whitespace token, giving a class number of 27. Each sample is a sequence of 256 characters.

**Metrics.** For continuous data, we adopt the widely used FID (Heusel et al., 2017) as the sample quality metric. We compute the FID metric on 10K generated samples for efficiency. For discrete data, there is no widely adopted sample-based metric comparable to FID in image modeling. Given

our reliance on a simple character-level text dataset, we found that spelling accuracy (SA) is a straightforward yet effective metric for measuring the quality of text generation. Specifically, SA is defined as the ratio of correctly spelled words to the total words in the entire generated sequence, which is segmented by spaces. In each experiment, we collect 1,000 generated samples to calculate the metric. Additionally, we conducted a user study for text generation quality evaluation. For the user study, there are 100 questions for each one vs. one comparison (e.g., BFN vs. BFN-Solver1). In each question, participants were presented with two sentences randomly generated from two methods. Participants were instructed to choose a sentence of higher quality, which is known as the *two-alternative forced choice* methodology (Kawar et al., 2023; Bar-Tal et al., 2022; Park et al., 2020). Please see Appendix C for more experimental details.

**Truncation.**  $\eta$  is a manually tuned hyperparameter specified in each experiment. For both  $p_{1-\eta}(\mu)$  and  $p_{1-\eta}(\mathbf{z})$ , we found an isotropic Gaussian with zero mean and a calculated variance works well. We provide preliminary analyses of the variance in Appendix C.1.

### 6.2. Fast Sampling on Continuous Data

We compare our proposed fast sampling methods with the original BFN continuous sampler in this section.

As illustrated in Fig. 3, with the NFE less than 100, BFN-Solver++1, BFN-Solver++2, and SDE-BFN-Solver++2 significantly outperform the BFN baseline. Moreover, BFN-Solver++2 achieves better results compared to BFN-Solver++1. When the NFE is higher (e.g., more than 500), our observations reveal that SDE-based samplers exhibit slightly better performance over ODE-based samplers, which aligns with the diffusion model (Song et al., 2020; Karras et al., 2022; Nie et al., 2023). Please see Appendix D.1 and Appendix D.5 for more quantitative results and randomly generated images, respectively.

We slightly tune the hyperparameter  $\eta$  for our methods on different NFEs to get the best results, as detailed in Appendix D.3.

### 6.3. Fast Sampling on Discrete Data

We compare our proposed fast sampling methods with the origin BFN discrete sampler in this section.

As illustrated in Fig. 4, with the NFE less than 30, BFN-Solver1, BFN-Solver2, and SDE-BFN-Solver2 significantly outperform the BFN baseline. Moreover, BFN-Solver2 and SDE-BFN-Solver2 achieve better results compared to BFN-Solver1, agreeing with the continuous case. We provide a preliminary user study in Fig. 5 with 10 NFEs and the results align with Fig. 4. When the NFE is higher (e.g., more than

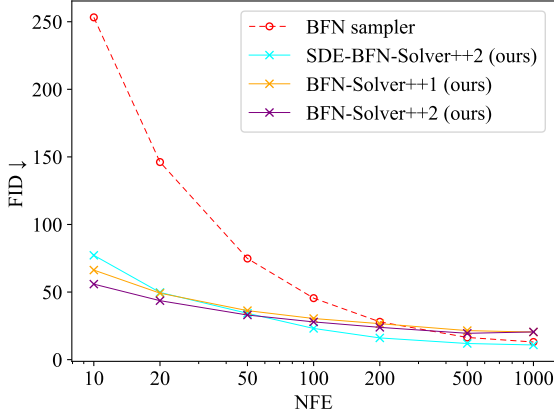


Figure 3. Fast sampling results on the continuous CIFAR-10 dataset. Sampling quality is measured by FID ↓, varying the NFE.

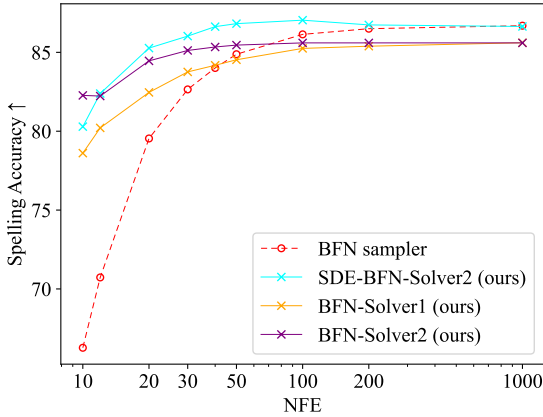


Figure 4. Fast sampling results on the discrete text8 dataset. Sampling quality is measured by SA ↑, varying the NFE.

500), we observe that SDE-based samplers exhibit slightly better performance than ODE-based samplers, which aligns with existing results in DMs. Please see Appendix D.2 and Appendix D.5 for more quantitative results and randomly generated texts.

We find that the hyperparameter  $\eta = 0.001$  is sufficient for all NFEs for BFN-Solvers to get excellent results. We refer the readers to Appendix D.4 for more details.

Finally, we perform an ablation of the original BFN solver in Fig. 6 and find that an exact solver that just removes the categorical sampling step from the BFN sampler works better, conforming to our theory.

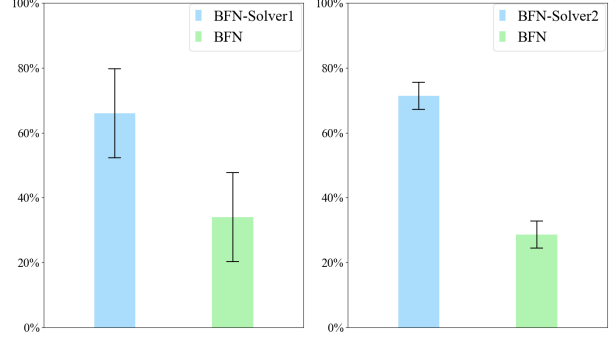


Figure 5. User study results on the discrete text8 dataset with 10 NFE. We present the preference rates (with 95% confidence intervals) of BFN-Solver1 and BFN-Solver2 over BFN baseline.

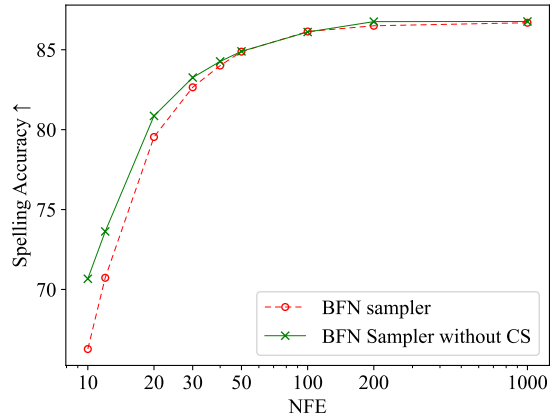


Figure 6. Ablation of the categorical sampling (CS) step in the BFN sampler on the text8 dataset. Sampling quality is measured by SA ↑, varying the NFE.

## 7. Conclusion

We unify BFNs and DMs by identifying the linear SDEs pertinent to the noise-addition processes in BFNs, illustrating that BFN’s regression losses correspond with denoise score matching, and validating the sampler in BFN as an effective first-order solver for the related reverse-time SDE. Motivated by these insights, we implement fast sampling techniques from DMs in BFNs, yielding promising results.

Building upon the established results of DMs, this paper establishes a principled and systematic approach to the analysis and enhancement of BFNs and future work includes the development of predictor-corrector samplers (Song et al., 2020; Zhao et al., 2023), improved methods for likelihood evaluation (Bao et al., 2022b;a), and novel training strategies to refine (Karras et al., 2022) and scale BFNs (Rombach et al., 2022).



Limitations of the paper include the scale of the datasets and evaluation metrics. In our experiment, for a fair comparison, we leverage the pre-trained models of BFNs, which are all trained on small datasets. Further, the samplers cannot be directly used in likelihood evaluation and we mainly employ the FID and spelling accuracy as surrogates for the sample quality, potentially introducing bias. Hopefully, these limitations can be solved by scaling up BFNs to common benchmarks, as mentioned in future work.

## Impact Statements

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

## References

- Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.
- Atkinson, K., Han, W., and Stewart, D. E. *Numerical solution of ordinary differential equations*, volume 81. John Wiley & Sons, 2009.
- Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces, 2023.
- Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Zhang, Q., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., Karras, T., and Liu, M.-Y. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers, 2023.
- Bao, F., Li, C., Sun, J., Zhu, J., and Zhang, B. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 1555–1584. PMLR, 2022a.
- Bao, F., Li, C., Zhu, J., and Zhang, B. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022b.
- Bao, F., Nie, S., Xue, K., Li, C., Pu, S., Wang, Y., Yue, G., Cao, Y., Su, H., and Zhu, J. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning*, pp. 1692–1717. PMLR, 2023.
- Bar-Tal, O., Ofri-Amar, D., Fridman, R., Kasten, Y., and Dekel, T. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, pp. 707–723. Springer, 2022.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.
- Campbell, A., Benton, J., Bortoli, V. D., Rainforth, T., Deligiannidis, G., and Doucet, A. A continuous time framework for discrete denoising models, 2022.
- Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.
- Chen, T., Zhang, R., and Hinton, G. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*, 2022.
- Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- Dieleman, S., Sartran, L., Roshannai, A., Savinov, N., Ganin, Y., Richemond, P. H., Doucet, A., Strudel, R., Dyer, C., Durkan, C., Hawthorne, C., Leblond, R., Grathwohl, W., and Adler, J. Continuous diffusion for categorical data, 2022.
- Graves, A., Srivastava, R. K., Atkinson, T., and Gomez, F. Bayesian flow networks, 2023.
- Guo, H., Lu, C., Bao, F., Pang, T., Yan, S., Du, C., and Li, C. Gaussian mixture solvers for diffusion models. *arXiv preprint arXiv:2311.00941*, 2023.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions, 2021.
- Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research (JMLR)*, 6(Apr):695–709, 2005.

- Jolicœur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., and Mitliagkas, I. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.
- Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022.
- Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., and Irani, M. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6007–6017, 2023.
- Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.
- Kloeden, P. E., Platen, E., Kloeden, P. E., and Platen, E. *Stochastic differential equations*. Springer, 1992.
- Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.
- Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.
- Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. B. Diffusion-lm improves controllable text generation, 2022.
- Liu, L., Ren, Y., Lin, Z., and Zhao, Z. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.
- Lou, A. and Ermon, S. Reflected diffusion models. *arXiv preprint arXiv:2304.04740*, 2023.
- Lou, A., Meng, C., and Ermon, S. Discrete diffusion language modeling by estimating the ratios of the data distribution, 2023.
- 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 *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14429–14460. PMLR, 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.
- Mahabadi, R. K., Tae, J., Ivison, H., Henderson, J., Beltagy, I., Peters, M. E., and Cohan, A. Tess: Text-to-text self-conditioned simplex diffusion. *arXiv preprint arXiv:2305.08379*, 2023.
- Mahoney, M. Large text compression benchmark, 2011.
- Meng, C., Choi, K., Song, J., and Ermon, S. Concrete score matching: Generalized score matching for discrete data, 2023.
- Nie, S., Guo, H. A., Lu, C., Zhou, Y., Zheng, C., and Li, C. The blessing of randomness: Sde beats ode in general diffusion-based image editing. *arXiv preprint arXiv:2311.01410*, 2023.
- Øksendal, B. *Stochastic differential equations*. Springer, 2003.
- OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Pang, T., Xu, K., Li, C., Song, Y., Ermon, S., and Zhu, J. Efficient learning of generative models via finite-difference score matching. *Advances in Neural Information Processing Systems*, 33:19175–19188, 2020.
- Park, T., Zhu, J.-Y., Wang, O., Lu, J., Shechtman, E., Efros, A., and Zhang, R. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020.
- Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.
- Richemond, P. H., Dieleman, S., and Doucet, A. Categorical sdes with simplex diffusion. *arXiv preprint arXiv:2210.14784*, 2022.
- 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., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022.
- Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.
- 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, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- Song, Y., Garg, S., Shi, J., and Ermon, S. Sliced score matching: A scalable approach to density and score estimation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations, 2021.
- Sun, H., Yu, L., Dai, B., Schuurmans, D., and Dai, H. Score-based continuous-time discrete diffusion models, 2023.
- Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.
- Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., and Zhu, J. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023.
- Xue, S., Yi, M., Luo, W., Zhang, S., Sun, J., Li, Z., and Ma, Z.-M. Sa-solver: Stochastic adams solver for fast sampling of diffusion models. *arXiv preprint arXiv:2309.05019*, 2023a.
- Xue, Z., Song, G., Guo, Q., Liu, B., Zong, Z., Liu, Y., and Luo, P. Raphael: Text-to-image generation via large mixture of diffusion paths. *arXiv preprint arXiv:2305.18295*, 2023b.
- Ye, J., Zheng, Z., Bao, Y., Qian, L., and Wang, M. Dinoiser: Diffused conditional sequence learning by manipulating noises, 2023.
- Zhang, Q., Tao, M., and Chen, Y. gddim: Generalized denoising diffusion implicit models. *arXiv preprint arXiv:2206.05564*, 2022.
- Zhao, W., Bai, L., Rao, Y., Zhou, J., and Lu, J. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *arXiv preprint arXiv:2302.04867*, 2023.

## A. Derivation for Continuous-time BFN on Continuous Data

### A.1. Proof of Theorem 4.1

*Proof.* In this proof, we will show that the marginal distribution  $\boldsymbol{\mu}(t)$  conditioned on  $\mathbf{x}$  of Eq. (15) at time  $t$  is the BFN in Eq. (8) by the “variation of constants” formula (Atkinson et al., 2009) and the Itô’s formula (Øksendal, 2003, Theorem 4.2.1).

To find the marginal distribution of  $\boldsymbol{\mu}(t)$ , we introduce a new process  $\tilde{\boldsymbol{\mu}}(t) := e^{-\int_0^t F(\tau) d\tau} \boldsymbol{\mu}(t)$ . Using Itô’s formula, we see that  $\tilde{\boldsymbol{\mu}}$  follows the equation below.

$$\begin{aligned} d\tilde{\boldsymbol{\mu}} &= e^{-\int_0^t F(\tau) d\tau} d\boldsymbol{\mu}(t) + \left( \frac{d}{dt} e^{-\int_0^t F(\tau) d\tau} \right) \boldsymbol{\mu}(t) dt \\ &= e^{-\int_0^t F(\tau) d\tau} (F(t)\boldsymbol{\mu}(t) dt + G(t) d\mathbf{w}) - F(t)e^{-\int_0^t F(\tau) d\tau} \boldsymbol{\mu}(t) dt \\ &= e^{-\int_0^t F(\tau) d\tau} G(t) d\mathbf{w}. \end{aligned}$$

Writing the above equation in the integral form we find that

$$\tilde{\boldsymbol{\mu}}(t) = \tilde{\boldsymbol{\mu}}(0) + \int_0^t e^{-\int_0^s F(\tau) d\tau} G(s) d\mathbf{w}(s). \quad (33)$$

It is known that from Itô’s isometry (Øksendal, 2003, Corollary 3.1.7) the above Itô’s integral is a Gaussian distribution with variance  $\int_0^t e^{-\int_0^s 2F(\tau) d\tau} G(s)^2 ds$ .

Note that  $F(t) = \frac{d}{dt} \ln \gamma(t)$  and  $G(t)^2 = -\frac{d}{dt} \gamma(t)$ , then Eq. (33) becomes

$$\frac{\gamma(0)}{\gamma(t)} \boldsymbol{\mu}(t) \sim \boldsymbol{\mu}(0) + \mathcal{N} \left( 0, \gamma(0)^2 \left( \frac{1}{\gamma(t)} - \frac{1}{\gamma(0)} \right) \mathbf{I} \right),$$

which shows that the distribution of  $\boldsymbol{\mu}(t)$  conditioned on  $\boldsymbol{\mu}(0)$  is

$$\boldsymbol{\mu}(t) \mid \boldsymbol{\mu}(0) \sim \mathcal{N} \left( \frac{\gamma(t)}{\gamma(0)} \boldsymbol{\mu}(0), \left( \gamma(t) - \frac{\gamma(t)^2}{\gamma(0)} \right) \mathbf{I} \right). \quad (34)$$

Recall that from Eq. (8) with  $t = 0$ , the initial distribution of  $\boldsymbol{\mu}(0)$  given the data  $\mathbf{x}$  is

$$\boldsymbol{\mu}(0) \mid \mathbf{x} \sim \mathcal{N}(\gamma(0)\mathbf{x}, \gamma(0)(1 - \gamma(0))\mathbf{I}). \quad (35)$$

In view of the above two equations, we see that

$$\boldsymbol{\mu}(t) \mid \mathbf{x} \sim \mathcal{N}(\gamma(t)\mathbf{x}, \gamma(t)(1 - \gamma(t))\mathbf{I}), \quad (36)$$

which is the BFN in Eq. (8) for any  $t \in [0, 1 - \eta]$ , and thus completes the proof.  $\square$

Finally, we note that the coefficient  $F(t)$  tends to infinity as  $t \rightarrow 1$ , which may violate the regularity assumptions used in Itô’s formula, and in the existence and the uniqueness of the solution of Eq. (15). Nevertheless, as noted in the paragraph after Theorem 4.1, when we restrict on the time interval  $t \in [0, 1 - \eta]$  for any fixed  $\eta > 0$ , the coefficients of Eq. (8) are well-behaved.

### A.2. Proof of Proposition 4.2

In this section, we provide the proof of Proposition 4.2. As we will see, the BFN sampler is equivalent to solving a reparametrized equation using a variant of the first-order DPM-Solver++ (Lu et al., 2022c).

*Proof.* We consider the following reparameterization of  $\hat{\epsilon}$ :

$$\hat{\mathbf{x}}(\boldsymbol{\mu}, t) := \frac{\boldsymbol{\mu}}{\gamma(t)} - \sqrt{\frac{1 - \gamma(t)}{\gamma(t)}} \hat{\boldsymbol{\epsilon}}(\boldsymbol{\mu}, t) \quad \Leftrightarrow \quad \hat{\boldsymbol{\epsilon}}(\boldsymbol{\mu}, t) = \sqrt{\frac{\gamma(t)}{1 - \gamma(t)}} \left[ \frac{\boldsymbol{\mu}}{\gamma(t)} - \hat{\mathbf{x}}(\boldsymbol{\mu}, t) \right], \quad (37)$$



under which the reverse SDE in Eq. (20) is

$$\begin{aligned}
 d\boldsymbol{\mu} &= \left( F(t)\boldsymbol{\mu} + \frac{G(t)^2}{\sqrt{\gamma(t)(1-\gamma(t))}} \hat{\boldsymbol{e}}(\boldsymbol{\mu}(t), t) \right) dt + G(t) d\bar{\boldsymbol{w}} \\
 &= \left( \frac{\gamma'(t)}{\gamma(t)} \boldsymbol{\mu} - \frac{\gamma'(t)}{\sqrt{\gamma(t)(1-\gamma(t))}} \hat{\boldsymbol{e}}(\boldsymbol{\mu}(t), t) \right) dt + \sqrt{-\gamma'(t)} d\bar{\boldsymbol{w}} \\
 &= \left( \frac{\gamma'(t)}{\gamma(t)} \boldsymbol{\mu} - \frac{\gamma'(t)}{1-\gamma(t)} \left( \frac{\boldsymbol{\mu}}{\gamma(t)} - \hat{\boldsymbol{x}}(\boldsymbol{\mu}(t), t) \right) \right) dt + \sqrt{-\gamma'(t)} d\bar{\boldsymbol{w}} \\
 &= \left( -\frac{\gamma'(t)}{1-\gamma(t)} \boldsymbol{\mu} + \frac{\gamma'(t)}{1-\gamma(t)} \hat{\boldsymbol{x}}(\boldsymbol{\mu}(t), t) \right) dt + \sqrt{-\gamma'(t)} d\bar{\boldsymbol{w}}.
 \end{aligned} \tag{38}$$

In the integral form similar to Eq. (33), for  $0 \leq t < s < 1$ , the above equation is

$$\boldsymbol{\mu}(t) = \frac{1-\gamma(t)}{1-\gamma(s)} \boldsymbol{\mu}(s) + (1-\gamma(t)) \int_s^t \frac{\gamma'(\tau)}{(1-\gamma(\tau))^2} \hat{\boldsymbol{x}}(\boldsymbol{\mu}(\tau), \tau) d\tau + \int_s^t \frac{1-\gamma(t)}{1-\gamma(\tau)} \sqrt{-\gamma'(\tau)} d\bar{\boldsymbol{w}}(\tau). \tag{39}$$

Approximating  $\hat{\boldsymbol{x}}(\boldsymbol{\mu}(\tau), \tau)$  using  $\hat{\boldsymbol{x}}(\boldsymbol{\mu}(s), s)$ , and compute the Itô's integral exactly, we can obtain the following first-order discretization

$$\begin{aligned}
 \boldsymbol{\mu}(t) &\approx \frac{1-\gamma(t)}{1-\gamma(s)} \boldsymbol{\mu}(s) + \left( \frac{\gamma(t)-\gamma(s)}{1-\gamma(s)} \right) \hat{\boldsymbol{x}}(\boldsymbol{\mu}(s), s) + \sqrt{\frac{1-\gamma(t)}{1-\gamma(s)}} (\gamma(t)-\gamma(s)) \mathbf{u}_s \\
 &= \frac{\gamma(t)}{\gamma(s)} \boldsymbol{\mu}(s) - \frac{\gamma(t)-\gamma(s)}{\sqrt{\gamma(s)(1-\gamma(s))}} \hat{\boldsymbol{e}}(\boldsymbol{\mu}(s), s) + \sqrt{\frac{1-\gamma(t)}{1-\gamma(s)}} (\gamma(t)-\gamma(s)) \mathbf{u}_s,
 \end{aligned}$$

where  $\mathbf{u}_s = \mathcal{N}(\mathbf{0}, \mathbf{I})$ , which matches the BFN's sampling rule in Eq. (10), and thus completes the proof.  $\square$

### A.3. Derivation of BFN-Solvers on Continuous Data

We follow the principles in DPM-Solvers (Lu et al., 2022b;c) to derive samplers of the ODE in Eq. (21). Let times  $0 \leq t < s \leq 1 - \eta$  and  $\lambda(t) = \frac{1}{2} \log \frac{\gamma(t)}{1-\gamma(t)}$ . The exact solution of this ODE can be formulated by “variation of constants” formula (Atkinson et al., 2009):

$$\boldsymbol{\mu}(t) = e^{\int_s^t F(\tau) d\tau} \boldsymbol{\mu}(s) + \int_s^t \left( e^{\int_\tau^t F(r) dr} \frac{G(\tau)^2}{2\sqrt{\gamma(\tau)(1-\gamma(\tau))}} \hat{\boldsymbol{e}}(\boldsymbol{\mu}(\tau), \tau) \right) d\tau. \tag{40}$$

Note that  $F(t) = \frac{d}{dt} \ln \gamma(t)$  and  $\frac{d\lambda(t)}{dt} = -\frac{G(t)^2}{\gamma(t)(1-\gamma(t))}$ , then the above equation becomes

$$\boldsymbol{\mu}(t) = \frac{\gamma(t)}{\gamma(s)} \boldsymbol{\mu}(s) - \gamma(t) \int_s^t \frac{d\lambda(\tau)}{d\tau} e^{-\lambda(\tau)} \hat{\boldsymbol{e}}(\boldsymbol{\mu}(\tau), \tau) d\tau \tag{41}$$

$$= \frac{\gamma(t)}{\gamma(s)} \boldsymbol{\mu}(s) - \gamma(t) \int_{\lambda(s)}^{\lambda(t)} e^{-\lambda} \hat{\boldsymbol{e}}(\boldsymbol{\mu}(t_\lambda(\lambda)), t_\lambda(\lambda)) d\lambda, \tag{42}$$

where the last equation uses the fact that  $\lambda(\tau)$  is strict monotone and  $t_\lambda(\cdot)$  is the inverse function of  $\lambda(t)$ .

Let  $\hat{\boldsymbol{e}}^{(m)}(\lambda) := \hat{\boldsymbol{e}}^{(m)}(\boldsymbol{\mu}(t_\lambda(\lambda)), t_\lambda(\lambda)) := \frac{d^m \hat{\boldsymbol{e}}(\boldsymbol{\mu}(t_\lambda(\lambda)), t_\lambda(\lambda))}{d\lambda^m}$  be the  $m$ -th order derivative of the map  $\lambda \mapsto \hat{\boldsymbol{e}}(\boldsymbol{\mu}(t_\lambda(\lambda)), t_\lambda(\lambda))$ , then we can approximate  $\hat{\boldsymbol{e}}(\boldsymbol{\mu}(t_\lambda(\lambda)), t_\lambda(\lambda))$  by the Taylor's expansion at  $\lambda = \lambda(s)$  for any  $k \geq 0$ :

$$\hat{\boldsymbol{e}}(\boldsymbol{\mu}(t_\lambda(\lambda)), t_\lambda(\lambda)) = \sum_{m=0}^{k-1} \frac{(\lambda - \lambda(s))^m}{m!} \hat{\boldsymbol{e}}^{(m)}(\lambda(s)) + O((\lambda - \lambda(s))^k). \tag{43}$$

Substituting the above equation into Eq. (42) yields

$$\boldsymbol{\mu}(t) = \frac{\gamma(t)}{\gamma(s)} \boldsymbol{\mu}(s) - \sum_{m=0}^{k-1} \gamma(t) \hat{\boldsymbol{e}}^{(m)}(\lambda(s)) \int_{\lambda(s)}^{\lambda(t)} e^{-\lambda} \frac{(\lambda - \lambda(s))^m}{m!} d\lambda + O((\lambda - \lambda(s))^{k+1}). \tag{44}$$

**Algorithm 1** BFN-Solver++1 (on continuous data)

---

**Require:** time steps  $\{t_i\}_{i=0}^M$  from  $t_0 = 1 - \eta$  to  $t_M = 0$ , noise prediction model  $\hat{e}(\mu, t)$   
 Denote  $\bar{\alpha}_t = \gamma(t)$ ,  $\bar{\sigma}_t = \sqrt{\gamma(t)(1 - \gamma(t))}$ ,  $\lambda_t = \log \frac{\bar{\alpha}_t}{\bar{\sigma}_t}$   
 $\mu_0 \sim \mathcal{N}(\mathbf{0}, \gamma(t_0)(1 - \gamma(t_0))\mathbf{I}), K\beta(t_0)\mathbf{I})$   
**for**  $i = 1$  **to**  $M$  **do**  
      $h_i = (\lambda_{t_i} - \lambda_{t_{i-1}})$   
      $\hat{\mathbf{x}}_i = \frac{\mu_{i-1}}{\gamma(t_{i-1})} - \sqrt{\frac{1-\gamma(t_{i-1})}{\gamma(t_{i-1})}} \hat{e}(\mu_{i-1}, t_{i-1})$   
      $\mu_i = \frac{\bar{\sigma}_{t_i}}{\bar{\sigma}_{t_{i-1}}} \mu_{i-1} - \bar{\alpha}_{t_i} (e^{-h_i} - 1) \hat{\mathbf{x}}_i$   
**end for**  
**return**  $\hat{\mathbf{x}}_M$

---

**Algorithm 2** BFN-Solver++2 (on continuous data)

---

**Require:** time steps  $\{t_i\}_{i=0}^M$  from  $t_0 = 1 - \eta$  to  $t_M = 0$ , noise prediction model  $\hat{e}(\mu, t)$   
 Denote  $\bar{\alpha}_t = \gamma(t)$ ,  $\bar{\sigma}_t = \sqrt{\gamma(t)(1 - \gamma(t))}$ ,  $\lambda_t = \log \frac{\bar{\alpha}_t}{\bar{\sigma}_t}$   
 $\mu_0 \sim \mathcal{N}(\mathbf{0}, \gamma(t_0)(1 - \gamma(t_0))\mathbf{I}), K\beta(t_0)\mathbf{I})$ , Initialize an empty buffer  $Q$ .  
 $\hat{\mathbf{x}}_0 = \frac{\mu_0}{\gamma(t_0)} - \sqrt{\frac{1-\gamma(t_0)}{\gamma(t_0)}} \hat{e}(\mu_0, t_0)$   
 $Q \leftarrow \hat{\mathbf{x}}_0$   
 $h_1 = (\lambda_{t_1} - \lambda_{t_0})$   
 $\mu_1 = \frac{\bar{\sigma}_{t_1}}{\bar{\sigma}_{t_0}} \mu_0 - \bar{\alpha}_{t_1} (e^{-h_1} - 1) \hat{\mathbf{x}}_0$   
 $\hat{\mathbf{x}}_1 = \frac{\mu_1}{\gamma(t_1)} - \sqrt{\frac{1-\gamma(t_1)}{\gamma(t_1)}} \hat{e}(\mu_1, t_1)$   
 $Q \leftarrow \hat{\mathbf{x}}_1$   
**for**  $i = 2$  **to**  $M$  **do**  
      $h_i = (\lambda_{t_i} - \lambda_{t_{i-1}})$   
      $r_i = \frac{h_{i-1}}{h_i}$   
      $D_i = \left(1 + \frac{1}{2r_i}\right) \hat{\mathbf{x}}_{i-1} - \frac{1}{2r_i} \hat{\mathbf{x}}_{i-2}$   
      $\mu_i = \frac{\bar{\sigma}_{t_i}}{\bar{\sigma}_{t_{i-1}}} \mu_{i-1} - \bar{\alpha}_{t_i} (e^{-h_i} - 1) D_i$   
      $\hat{\mathbf{x}}_i = \frac{\mu_i}{\gamma(t_i)} - \sqrt{\frac{1-\gamma(t_i)}{\gamma(t_i)}} \hat{e}(\mu_i, t_i)$   
     If  $i < M$ , then  $Q \leftarrow \hat{\mathbf{x}}_i$   
**end for**  
**return**  $\hat{\mathbf{x}}_M$

---

Given an initial value  $\mu_0$  and time steps  $\{t_i\}_{i=0}^n$  from  $t_0 = 1 - \eta$  to  $t_n = 0$ . The solver uses  $n$  steps to iteratively compute a sequence  $\{\mu_i\}_{i=1}^n$  to approximate the true solutions at time  $\{t_i\}_{i=1}^n$  using Eq. (44) by setting  $t = t_i$  and  $s = t_{i-1}$  for each  $i = 1, \dots, n$ . We can use the well-established finite difference method or the Runge-Kutta method to avoid the computation of the high-order derivatives  $\hat{e}^{(m)}(\lambda)$ , as we will illustrate in the case of discrete data in Appendix. B (see also, e.g., Kloeden et al. (1992); Lu et al. (2022b)).

**BFN as a DM with a special noise schedule** The derivation above with a noise prediction network are presented mainly to illustrate the idea. In our implementation, we use a data prediction network  $\hat{\mathbf{x}}$  as defined in Appendix. A.2. For simplicity, we do not derive the sampler step by step. Instead, a more straightforward way to get BFN-Solver on continuous data is to treat BFN as a DM with a special noise schedule  $\alpha(t) = \gamma(t)$  and  $\sigma^2(t) = \gamma(t)(1 - \gamma(t))$ . We can directly plugin the relation to the samplers proposed in Lu et al. (2022b,c) to obtain our first-order ODE solver BFN-Solver++1, second-order ODE solver BFN-Solver++2 and second-order SDE solver SDE-BFN-Solver++2, as presented in Algorithm 1, 2 and 3 respectively.

The plugging idea does not apply to discrete data and we provide a detailed derivation in Appendix. B.

**Algorithm 3** SDE-BFN-Solver++2 (on continuous data)

---

**Require:** time steps  $\{t_i\}_{i=0}^M$  from  $t_0 = 1 - \eta$  to  $t_M = 0$ , noise prediction model  $\hat{\epsilon}(\boldsymbol{\mu}, t)$   
 Denote  $\bar{\alpha}_t = \gamma(t)$ ,  $\bar{\sigma}_t = \sqrt{\gamma(t)(1 - \gamma(t))}$ ,  $\lambda_t = \log \frac{\bar{\alpha}_t}{\bar{\sigma}_t}$   
 $\boldsymbol{\mu}_0 \sim \mathcal{N}(\mathbf{0}, \gamma(t_0)(1 - \gamma(t_0))\mathbf{I})$ ,  $K\beta(t_0)\mathbf{I}$ , Initialize an empty buffer  $Q$ .  
 $\hat{\mathbf{x}}_0 = \frac{\boldsymbol{\mu}_0}{\gamma_{t_0}} - \sqrt{\frac{1 - \gamma_{t_0}}{\gamma_{t_0}}} \hat{\epsilon}(\boldsymbol{\mu}_0, t_0)$   
 $Q \leftarrow \hat{\mathbf{x}}_0$   
 $h_1 = (\lambda_{t_1} - \lambda_{t_0})$   
 $\boldsymbol{\mu}_1 = \frac{\bar{\sigma}_{t_1}}{\bar{\sigma}_{t_0}} \boldsymbol{\mu}_0 - \bar{\alpha}_{t_1} (e^{-h_1} - 1) \hat{\mathbf{x}}_0$   
 $\hat{\mathbf{x}}_1 = \frac{\boldsymbol{\mu}_1}{\gamma_{t_1}} - \sqrt{\frac{1 - \gamma_{t_1}}{\gamma_{t_1}}} \hat{\epsilon}(\boldsymbol{\mu}_1, t_1)$   
 $Q \leftarrow \hat{\mathbf{x}}_1$   
**for**  $i = 2$  **to**  $M$  **do**  
      $h_i = (\lambda_{t_i} - \lambda_{t_{i-1}})$   
      $r_1 = \frac{h_{i-1}}{h_i}$   
      $\mathbf{D}_i = \frac{1}{r_1} (\hat{\mathbf{x}}_{i-1} - \hat{\mathbf{x}}_{i-2})$   
      $\mathbf{u}_{t_{i-1}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   
      $\boldsymbol{\mu}_i = \frac{\bar{\sigma}_{t_i}}{\bar{\sigma}_{t_{i-1}}} e^{-h_i} \boldsymbol{\mu}_{i-1} + \bar{\alpha}_{t_i} (1 - e^{-2h_i}) \hat{\mathbf{x}}_{i-1} + \frac{\bar{\alpha}_t (1 - e^{-2h_i})}{2} \mathbf{D}_i + \bar{\sigma}_t \sqrt{1 - e^{-2h_i}} \mathbf{u}_{t_{i-1}}$   
      $\hat{\mathbf{x}}_i = \frac{\boldsymbol{\mu}_i}{\gamma_{t_i}} - \sqrt{\frac{1 - \gamma(t_i)}{\gamma(t_i)}} \hat{\epsilon}(\boldsymbol{\mu}_i, t_i)$   
     If  $i < M$ , then  $Q \leftarrow \hat{\mathbf{x}}_i$   
**end for**  
**return**  $\hat{\mathbf{x}}_M$

---

## B. Derivation for Continuous-time BFN on Discrete Data

### B.1. Proof of Theorem 5.1

*Proof.* Similar to the proof in Appendix. A.1, let  $\tilde{\mathbf{z}}(t) := e^{-\int_0^t H(\tau) d\tau} \mathbf{z}(t)$ , then we have

$$\tilde{\mathbf{z}}(t) = \tilde{\mathbf{z}}(0) + \int_0^t e^{-\int_0^s H(\tau) d\tau} L(s) d\mathbf{w}(s). \quad (45)$$

Since  $H(t) = \frac{d}{dt} \ln \beta(t)$  and  $L(t)^2 = -K \frac{d}{dt} \beta(t)$ , then the above equation becomes

$$\frac{\beta(0)}{\beta(t)} \mathbf{z}(t) \sim \mathbf{z}(0) + \mathcal{N}\left(0, K\beta(0)^2 \left(\frac{1}{\beta(t)} - \frac{1}{\beta(0)}\right) \mathbf{I}\right),$$

which shows that the distribution of  $\mathbf{z}(t)$  conditioned on  $\mathbf{z}(0)$  is

$$\mathbf{z}(t) \mid \mathbf{z}(0) \sim \mathcal{N}\left(\frac{\beta(t)}{\beta(0)} \mathbf{z}(0), K \left(\beta(t) - \frac{\beta(t)^2}{\beta(0)}\right) \mathbf{I}\right). \quad (46)$$

Recall that from Eq. (11) with  $t = 0$  we know that

$$\mathbf{z}(0) \mid \mathbf{x} \sim \mathcal{N}(\beta(0) \mathbf{w}_{\mathbf{x}}, K\beta(0) \mathbf{I}). \quad (47)$$

Combining the above two equations, we find that

$$\mathbf{z}(t) \mid \mathbf{x} \sim \mathcal{N}(\beta(t) \mathbf{w}_{\mathbf{x}}, K\beta(t) \mathbf{I}), \quad (48)$$

which is the BFN in Eq. (11) for any  $t \in [0, 1 - \eta]$ .  $\square$

### B.2. Proof of Theorem 5.2

*Proof.* Recall that the loss function on discrete data in Eq. (12) is

$$\mathcal{L}^\infty(\mathbf{x}) = \mathbb{E}_{q_F(\boldsymbol{\theta} \mid \mathbf{x}, t), t \sim U(0, 1)} K\beta_1 t \|\mathbf{e}_{\mathbf{x}} - \hat{\epsilon}(\boldsymbol{\theta}, t)\|^2, \quad (49)$$

where  $q_F(\boldsymbol{\theta}|\mathbf{x}, t)$  is specially defined with softmax function to ensure  $\boldsymbol{\theta}$  lies in the simplex as follows, making its density complex.<sup>5</sup>

$$q_F(\boldsymbol{\theta}|\mathbf{x}, t) = \mathbb{E}_{\mathbf{z}(t) \sim \mathcal{N}(\beta(t)(K\mathbf{e}_x - 1), \beta(t)KI)} \delta(\boldsymbol{\theta} - \text{softmax}(\mathbf{z}(t))).$$

Note that to obtain a sample  $\boldsymbol{\theta}(t) \sim q_F(\boldsymbol{\theta}|\mathbf{x}, t)$ , we can first sample  $\mathbf{z}(t) \sim \mathcal{N}(\beta(t)(K\mathbf{e}_x - 1), \beta(t)KI)$  and apply the deterministic transform  $\boldsymbol{\theta}(t) := \text{softmax}(\mathbf{z}(t))$ . Then, we can rewrite the loss function as

$$\begin{aligned} \mathcal{L}^\infty(\mathbf{x}) &= \mathbb{E}_{t \sim U(0,1)} \mathbb{E}_{\mathbf{z}(t) \sim q(\mathbf{z}(t)|\mathbf{x})} K\beta_1 t \|\mathbf{e}_x - \hat{\mathbf{e}}(\text{softmax}(\mathbf{z}(t)), t)\|^2 \\ &= \mathbb{E}_{t \sim U(0,1)} \mathbb{E}_{\mathbf{z}(t) \sim q(\mathbf{z}(t)|\mathbf{x})} K\beta_1 t \|\mathbf{e}_x - \hat{\mathbf{e}}_s(\mathbf{z}(t), t)\|^2, \end{aligned}$$

where  $q(\mathbf{z}(t)|\mathbf{x}) = \mathcal{N}(\beta(t)(K\mathbf{e}_x - 1), \beta(t)KI)$  whose score function is given by

$$\nabla_{\mathbf{z}} \log q(\mathbf{z}(t)|\mathbf{x}) = -\frac{\mathbf{z}(t)}{K\beta(t)} + \mathbf{e}_x - \frac{1}{K}. \quad (50)$$

Let

$$\hat{\mathbf{s}}(\mathbf{z}(t), t) := -\frac{\mathbf{z}(t)}{K\beta(t)} + \hat{\mathbf{e}}_s(\mathbf{z}(t), t) - \frac{1}{K}. \quad (51)$$

Substituting  $\nabla_{\mathbf{z}} \log q(\mathbf{z}(t)|\mathbf{x})$  and  $\hat{\mathbf{s}}(\text{softmax}(\mathbf{z}(t)), t)$  into the loss function yields

$$\begin{aligned} \mathcal{L}^\infty(\mathbf{x}) &= \mathbb{E}_{t \sim U(0,1)} \mathbb{E}_{\mathbf{z}(t) \sim q(\mathbf{z}(t)|\mathbf{x})} K\beta_1 t \|\mathbf{e}_x - \hat{\mathbf{e}}_s(\mathbf{z}(t), t)\|^2 \\ &= \mathbb{E}_{t \sim U(0,1)} \mathbb{E}_{\mathbf{z}(t) \sim q(\mathbf{z}(t)|\mathbf{x})} K\beta_1 t \|\nabla_{\mathbf{z}} \log q(\mathbf{z}(t)|\mathbf{x}) - \hat{\mathbf{s}}(\mathbf{z}(t), t)\|^2. \end{aligned} \quad (52)$$

The loss in Eq. (52) performs the denosing score matching (DSM), and according to Vincent (2011), the optimal solution of it w.r.t.  $\hat{\mathbf{s}}$  is the score of the distribution of  $\mathbf{z}(t)$ . Inspecting Eq. (51), we find that there is a one-to-one correspondance between the parameterized score function  $\hat{\mathbf{s}}$  in the DSM loss and the function  $\hat{\mathbf{e}}_s$  in BFN, showing that the optimization of BFN is equivalent to that of DM.  $\square$

### B.3. Proof of Proposition 5.3

*Proof.* Recall that the BFN sampler is defined in Eqs. (13) and (14), and  $\beta(t) = \beta_1(1-t)^2$ . If we remove the categorical sampling step in Eq. (13), the sampling rule becomes

$$\mathbf{z}_i = \mathbf{z}_{i-1} + (\beta(t_i) - \beta(t_{i-1}))(K\hat{\mathbf{e}}_s(\mathbf{z}_{i-1}, t_{i-1}) - 1) + \sqrt{K(\beta(t_i) - \beta(t_{i-1}))}\mathbf{u}_i, \quad (53)$$

where  $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Next, to show that the update rule in Eq. (53) is a first-order discretization of the reverse SDE, we write the SDE defined in Eq. (29) in the integral form as follows for any  $0 \leq t < s \leq 1 - \eta$ :

$$\mathbf{z}(t) = \mathbf{z}(s) - \int_s^t L(\tau)^2 \left[ \hat{\mathbf{e}}_s(\mathbf{z}(\tau), \tau) - \frac{1}{K} \right] d\tau + \int_s^t L(\tau) d\bar{\mathbf{w}}(\tau). \quad (54)$$

Note that the Itô integral follows the Gaussian distribution, and by Eq. (26), we know that

$$\int_s^t L(\tau) d\bar{\mathbf{w}}(\tau) \sim \mathcal{N}\left(\mathbf{0}, \int_t^s L(\tau)^2 dt \mathbf{I}\right) = \mathcal{N}(\mathbf{0}, K(\beta(t) - \beta(s))\mathbf{I}).$$

Then, by approximating  $\hat{\mathbf{e}}_s(\mathbf{z}(\tau), \tau) = \hat{\mathbf{e}}_s(\mathbf{z}(s), s) + O(\tau - s)$  in Eq. (54), we find

$$\begin{aligned} \mathbf{z}(t) &= \mathbf{z}(s) - \int_s^t L(\tau)^2 \left[ \hat{\mathbf{e}}_s(\mathbf{z}(s), s) + O(\tau - s) - \frac{1}{K} \right] d\tau + \int_s^t L(\tau) d\bar{\mathbf{w}}(\tau) \\ &= \mathbf{z}(s) + K(\beta(t) - \beta(s)) \left[ \hat{\mathbf{e}}_s(\mathbf{z}(s), s) + O(t - s) - \frac{1}{K} \right] + \sqrt{K(\beta(t) - \beta(s))}\mathbf{u}_s \\ &= \mathbf{z}(s) + (\beta(t) - \beta(s)) [K\hat{\mathbf{e}}_s(\mathbf{z}(s), s) - 1] + \sqrt{K(\beta(t) - \beta(s))}\mathbf{u}_s + O((t - s)^2), \end{aligned}$$

<sup>5</sup>Note that  $\boldsymbol{\theta}$  lies in a low-dimensional simplex, so the “density” should be defined w.r.t. a low-dimensional Lebesgue measure. We intend to define the distribution  $q_F$  by a sampling procedure as introduced later.



where  $\mathbf{u}_s \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Now, the proof is completed by setting  $t = t_i$  and  $s = t_{i-1}$  in the above equation, and comparing the result with Eq. (53).  $\square$

#### B.4. Derivation of SDE-BFN-Solvers on Discrete Data

In this section we derive the SDE-BFN-Solvers of solving Eq. (29). We shall highlight that the SDE to be solved is different from those solved in BFNs and DMs on continuous data, as it contains no linear terms. We also follow the recipe of DPM-Solvers (Lu et al., 2022b;c) in reducing the discretization error to analytically simplify the equation as much as possible, instead of approximating it directly. The derived algorithms can be found in Algorithm 4, 5.

As we have seen in Appendix. B.3, the BFN sampler without the categorical sampling step (Eq. (53)) is the desired first-order solver, which is called SDE-BFN-Solver1. To further reduce the discretization error, we consider the first-order approximation of  $\hat{e}_s(\mathbf{z}(\tau), \tau)$  in Eq. (54):

$$\hat{e}_s(\mathbf{z}(\tau), \tau) = \hat{e}_s(\mathbf{z}(s), s) + \hat{e}_s^{(1)}(\mathbf{z}(s), s)(\tau - s) + O((\tau - s)^2), \quad (55)$$

where  $\hat{e}_s^{(1)}$  is the derivative of  $\hat{e}_s(\mathbf{z}(s), s)$  w.r.t.  $s$ . Then, setting  $t = t_i$  and  $s = t_{i-1}$ , Eq. (54) becomes

$$\begin{aligned} \mathbf{z}(t_i) &= \mathbf{z}(t_{i-1}) - \int_{t_{i-1}}^{t_i} L(\tau)^2 \left[ \hat{e}_s(\mathbf{z}(t_{i-1}), t_{i-1}) + \hat{e}_s^{(1)}(\mathbf{z}(t_{i-1}), t_{i-1})(\tau - t_{i-1}) - \frac{1}{K} \right] d\tau \\ &\quad + \int_{t_{i-1}}^{t_i} L(\tau) d\bar{\mathbf{w}}(\tau) + O((t_i - t_{i-1})^3) \\ &= \mathbf{z}(t_{i-1}) + (\beta(t_i) - \beta(t_{i-1})) [K \hat{e}_s(\mathbf{z}(t_{i-1}), t_{i-1}) - 1] \\ &\quad - \frac{1}{3} K \beta_1 (t_i - t_{i-1})^2 (t_{i-1} + 2t_i - 3) \hat{e}_s^{(1)}(\mathbf{z}(t_{i-1}), t_{i-1}) \\ &\quad + \sqrt{K(\beta(t_i) - \beta(t_{i-1}))} \mathcal{N}(\mathbf{0}, \mathbf{I}) + O((t_i - t_{i-1})^3). \end{aligned}$$

Finally, we approximate the derivative  $\hat{e}_s^{(1)}$  by a finite difference as follows to obtain the final discretization algorithm, namely SDE-DPM-Solver2.

$$\hat{e}_s^{(1)}(\mathbf{z}_{i-1}, t_{i-1}) \approx \frac{\hat{e}_s(\mathbf{z}_{i-2}, t_{i-2}) - \hat{e}_s(\mathbf{z}_{i-1}, t_{i-1})}{t_{i-2} - t_{i-1}}. \quad (56)$$

#### B.5. Derivation of BFN-Solvers on Discrete Data

Eq. In this section, we derive the BFN-Solvers on discrete data using the following integral form of the parameterized ODE in (30).

$$\mathbf{z}(t) = \frac{1-t}{1-s} \mathbf{z}(s) + \beta_1(1-t)(t-s) - K\beta_1(1-t) \int_s^t \hat{e}_s(\mathbf{z}(\tau), \tau) d\tau. \quad (57)$$

**BFN-Solver1** The first-order solver approximates  $\hat{e}_s(\mathbf{z}(\tau), \tau)$  in the above integral by  $\hat{e}_s(\mathbf{z}(s), s)$  directly, yielding the following update rule:

$$\mathbf{z}_i = \frac{1-t_i}{1-t_{i-1}} \mathbf{z}_{i-1} + \beta_1(1-t_i)(t_i - t_{i-1})(1 - K\hat{e}_s(\mathbf{z}_{i-1}, t_{i-1})). \quad (58)$$

**BFN-Solver2** The second-order solver approximates  $\hat{e}_s(\mathbf{z}(\tau), \tau)$  with  $\hat{e}_s(\mathbf{z}(s), s) + \hat{e}_s^{(1)}(\mathbf{z}(s), s)(\tau - s)$ , where  $\hat{e}_s(\mathbf{z}(s), s)$  is the derivative of  $\hat{e}_s$  w.r.t.  $s$ . Using such an approximation, we have

$$\begin{aligned} \int_s^t \hat{e}_s(\mathbf{z}(\tau), \tau) d\tau &= \int_s^t \hat{e}_s(\mathbf{z}(s), s) + \hat{e}_s^{(1)}(\mathbf{z}(s), s)(\tau - s) + O(\tau - s)^2 d\tau \\ &= \hat{e}_s(\mathbf{z}(s), s)(t - s) + \hat{e}_s^{(1)}(\mathbf{z}(s), s) \frac{(t - s)^2}{2} + O((t - s)^3). \end{aligned} \quad (59)$$

Following the method used in DPM-Solver (Lu et al., 2022b), we use an intermediate time  $r \in (s, t)$  to approximate the derivative term. We use the first-order approximation in Eq. (58) to compute  $\mathbf{z}(r)$  and the finite difference to estimate the

**Algorithm 4** SDE-BFN-Solver1 (on discrete data)

---

**Require:** time steps  $\{t_i\}_{i=0}^M$ , from  $t_0 = 1 - \eta$  to  $t_M = 0$ , model  $\hat{e}_s(\mathbf{z}, t)$ ,  $\beta(t) = \beta_1(1 - t)^2$   
 $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, K\beta(t_0)\mathbf{I})$   
**for**  $i = 1$  **to**  $M - 1$  **do**  
      $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   
      $\mathbf{z}_i = \mathbf{z}_{i-1} + (\beta(t_i) - \beta(t_{i-1}))(K\hat{e}_s(\mathbf{z}_{i-1}, t_{i-1})) - 1) + \sqrt{K(\beta(t_i) - \beta(t_{i-1}))}\mathbf{u}_i$   
**end for**  
 $\hat{\mathbf{x}} = \text{argmax}(\hat{e}_s(\mathbf{z}_{M-1}, t_{M-1}))$   
**return**  $\hat{\mathbf{x}}$

---

**Algorithm 5** SDE-BFN-Solver2 (on discrete data)

---

**Require:** time steps  $\{t_i\}_{i=0}^M$ , from  $t_0 = 1 - \eta$  to  $t_M = 0$ , model  $\hat{e}_s(\mathbf{z}, t)$ ,  $\beta(t) = \beta_1(1 - t)^2$   
 $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, K\beta(t_0)\mathbf{I})$ , Initialize an empty buffer  $Q$   
 $\mathbf{u}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   
 $Q \leftarrow \hat{e}_s(\mathbf{z}_0, t_0)$   
 $\mathbf{z}_1 = \mathbf{z}_0 + (\beta(t_1) - \beta(t_0))(K\hat{e}_s(\mathbf{z}_0, t_0)) - 1) + \sqrt{K(\beta(t_1) - \beta(t_0))}\mathbf{u}_1$   
**for**  $i = 2$  **to**  $M - 1$  **do**  
      $D_1 = \frac{\hat{e}_s(\mathbf{z}_{i-2}, t_{i-2}) - \hat{e}_s(\mathbf{z}_{i-1}, t_{i-1})}{t_{i-2} - t_{i-1}}$   
      $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   
      $\mathbf{z}_i = \mathbf{z}_{i-1} + (\beta(t_i) - \beta(t_{i-1}))(K\hat{e}_s(\mathbf{z}_{i-1}, t_{i-1})) - 1) - \frac{1}{3}K\beta_1(t_i - t_{i-1})^2(t_{i-1} + 2t_i - 3)D_1 + \sqrt{K(\beta(t_i) - \beta(t_{i-1}))}\mathbf{u}_i$   
     If  $i < M - 1$ , then  $Q \leftarrow \hat{e}_s(\mathbf{z}_{i-1}, t_{i-1})$   
**end for**  
 $\hat{\mathbf{x}} = \text{argmax}(\hat{e}_s(\mathbf{z}_{M-1}, t_{M-1}))$   
**return**  $\hat{\mathbf{x}}$

---

derivative  $\hat{e}_s^{(1)}$ :

$$\mathbf{z}(r) = \frac{1-t}{1-s}\mathbf{z}(s) + \beta_1(1-r)(r-s)(1 - K\hat{e}_s(\mathbf{z}(s), s)) + O((r-s)^2), \quad (60)$$

$$\hat{e}_s^{(1)}(\mathbf{z}(s), s) = \frac{\hat{e}_s(\mathbf{z}(r), r) - \hat{e}_s(\mathbf{z}(s), s)}{r-s} + O(r-s). \quad (61)$$

Combining the above equations with Eq. (59), we find that

$$\int_s^t \hat{e}_s(\mathbf{z}(\tau), \tau) d\tau = \hat{e}_s(\mathbf{z}(s), s)(t-s) + \frac{(\hat{e}_s(\mathbf{z}(r), r) - \hat{e}_s(\mathbf{z}(s), s))(t-s)^2}{2(r-s)} + O((t-s)^3). \quad (62)$$

Finally, let  $\eta > 0$  be an arbitrarily small constant and choose time steps  $\{t_i\}_{i=0}^M$  from  $t_0 = 1 - \eta$  to  $t_M = 0$ . Given an initial value sample,  $\{\mathbf{z}_i\}_{i=1}^M$  is computed iteratively as follows, by choosing  $(s, t, r) := (t_{i-1}, t_i, \frac{t_i + t_{i-1}}{2})$  for each  $i$  in the above derivation.

$$\mathbf{z}_{i-1/2} = \frac{1-t_{i-1/2}}{1-t_{i-1}}\mathbf{z}_{i-1} + \beta_1(1-t_{i-1/2})(t_{i-1/2} - t_{i-1})(1 - K\hat{e}_s(\mathbf{z}_{i-1}, t_{i-1})), \quad (63)$$

$$\begin{aligned} \mathbf{z}_i &= \frac{1-t_i}{1-t_{i-1}}\mathbf{z}_{i-1} + \beta_1(1-t_i)(t_i - t_{i-1}) - c(t_i)(t_i - t_{i-1})\hat{e}_s(\mathbf{z}_{i-1}, t_{i-1}) \\ &\quad - c(t_i)\frac{(t_i - t_{i-1})^2}{2(t_{i-1/2} - t_{i-1})}(\hat{e}_s(\mathbf{z}_{i-1/2}, t_{i-1/2}) - \hat{e}_s(\mathbf{z}_{i-1}, t_{i-1})). \end{aligned} \quad (64)$$

where  $t_{i-1/2} = (t_i + t_{i-1})/2$  and  $c(t) = K\beta_1(1-t)$ .

**Algorithm 6** BFN-Solver1 (on discrete data)

---

**Require:** time steps  $\{t_i\}_{i=0}^M$ , from  $t_0 = 1 - \eta$  to  $t_M = 0$ , model  $\hat{e}_s(\mathbf{z}, t)$ ,  $\beta(t) = \beta_1(1 - t)^2$   
 $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, K\beta(t_0)\mathbf{I})$   
**for**  $i = 1$  **to**  $M - 1$  **do**  
 $\hat{\mathbf{z}}_i = \frac{1-t_i}{1-t_{i-1}}\mathbf{z}_{i-1} + \beta_1(1-t_i)(t_i - t_{i-1})(1 - K\hat{e}_s(\mathbf{z}_{i-1}, t_{i-1}))$   
**end for**  
 $\hat{\mathbf{x}} = \text{argmax}(\hat{e}_s(\mathbf{z}_{M-1}, t_{M-1}))$   
**return**  $\hat{\mathbf{x}}$

---

**Algorithm 7** BFN-Solver2 (on discrete data)

---

**Require:** time steps  $\{t_i\}_{i=0}^M$ , from  $t_0 = 1 - \eta$  to  $t_M = 0$ , model  $\hat{e}_s(\mathbf{z}, t)$ ,  $\beta(t) = \beta_1(1 - t)^2$   
 $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, K\beta(t_0)\mathbf{I})$   
**for**  $i = 1$  **to**  $M - 1$  **do**  
 $t_{i-1/2} = (t_i + t_{i-1})/2$   
 $c_i = K\beta_1(1 - t_i)$   
 $D_1 = \frac{\hat{e}_s(\mathbf{z}_{i-1/2}, t_{i-1/2}) - \hat{e}_s(\mathbf{z}_{i-1}, t_{i-1})}{t_{i-1/2} - t_{i-1}}$   
 $\mathbf{z}_i = \frac{1-t_i}{1-t_{i-1}}\mathbf{z}_{i-1} + \beta_1(1-t_i)(t_i - t_{i-1}) - c_i(t_i - t_{i-1})\hat{e}_s(\mathbf{z}_{i-1}, t_{i-1}) - \frac{c_i(t_i - t_{i-1})^2}{2}D_1$   
**end for**  
 $\hat{\mathbf{x}} = \text{argmax}(\hat{e}_s(\mathbf{z}_{M-1}, t_{M-1}))$   
**return**  $\hat{\mathbf{x}}$

---

## C. Experimental Details

### C.1. Choices of the Initialization Distribution

Since the exact distribution of  $\boldsymbol{\mu}(1 - \eta)$  is unknown, we need to choose an approximation of it as the initialization distribution. The following proposition identifies the best initial distribution among isotropic Gaussian distributions.

**Proposition C.1.** *Let  $p_t(\boldsymbol{\mu}) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} q_F(\boldsymbol{\mu} | \mathbf{x}, \gamma(t))$  be the distribution of  $\boldsymbol{\mu}(t)$  in Eq. (8), then  $q_t(\boldsymbol{\mu}) := \mathcal{N}(\boldsymbol{\mu} | \mathbf{m}^*(t), (\sigma^*(t))^2 \mathbf{I})$  minimizes the Kullback-Leibler (KL) divergence between  $p_t$  and any isotropic Gaussian distributions, where*

$$\mathbf{m}^*(t) = \gamma(t)\mathbf{m}_{\text{data}}, \quad \text{and} \quad (\sigma^*(t))^2 = \gamma(t)(1 - \gamma(t)) + \gamma(t)^2 \frac{\text{Tr } \Sigma_{\text{data}}}{D},$$

$D$  is the dimensionality of  $\boldsymbol{\mu}$ , and  $\mathbf{m}_{\text{data}}, \Sigma_{\text{data}}$  are the mean and the covariance matrix of the data distribution  $p_{\text{data}}$ , respectively. In other words, we have

$$(\mathbf{m}^*(t), \sigma^*(t)) = \arg \min_{\mathbf{m}, \sigma} D_{\text{KL}}(p_t \| \mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{I})), \quad (65)$$

where  $D_{\text{KL}}$  is the Kullback-Leibler divergence.

*Proof.* Since  $p_F(\boldsymbol{\mu}(t) | \mathbf{x}, \gamma(t)) = \mathcal{N}(\gamma(t)\mathbf{x}, \gamma(t)(1 - \gamma(t))\mathbf{I})$ , the KL divergence can be written as

$$\begin{aligned} D_{\text{KL}}(p_t \| \mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{I})) &= - \mathbb{E}_{\boldsymbol{\mu} \sim p_t} \log \mathcal{N}(\boldsymbol{\mu} | \mathbf{m}, \sigma^2 \mathbf{I}) + \mathbb{E}_{\boldsymbol{\mu} \sim p_t} \log p_t(\boldsymbol{\mu}) \\ &= \mathbb{E}_{\boldsymbol{\mu} \sim p_t} \left[ \frac{D}{2} \log(2\pi) + D \log \sigma + \frac{\|\boldsymbol{\mu} - \mathbf{m}\|^2}{2\sigma^2} \right] + \mathbb{E}_{\boldsymbol{\mu} \sim p_t} \log p_t(\boldsymbol{\mu}), \end{aligned}$$

where  $D$  is the dimensionality of  $\boldsymbol{\mu}$ . By taking the derivative, we know the minimizer is

$$\begin{aligned} \mathbf{m}^*(t) &= \mathbb{E}_{\boldsymbol{\mu} \sim p_t} \boldsymbol{\mu} = \gamma(t)\mathbf{m}_{\text{data}}, \\ (\sigma^*(t))^2 &= \frac{1}{D} \mathbb{E}_{\boldsymbol{\mu} \sim p_t} \|\boldsymbol{\mu} - \mathbf{m}^*(t)\|^2 = \gamma(t)(1 - \gamma(t)) + \gamma(t)^2 \frac{\text{Tr } \Sigma_{\text{data}}}{D}. \end{aligned}$$

□

Table 2. **Image generation results on continuous CIFAR-10 dataset.** Sampling quality is measured by FID  $\downarrow$ , varying the number of function evaluations (NFE). We **bold** the best result under the corresponding setting. For instance, we underline the result of BFN at 50 steps and our BFN-Solvers2++ solver at 10 steps, where we achieve a speed-up of 5 times.

NFE	10	20	50	100	200	500	1000
BFN	253.23	146.19	<u>74.81</u>	45.50	27.98	16.24	13.05
SDE-BFN-SOLVER++2 (OURS)	77.19	49.73	34.29	<b>23.09</b>	<b>16.07</b>	<b>11.94</b>	<b>10.86</b>
BFN-SOLVER++1 (OURS)	66.27	49.18	36.27	30.43	26.62	21.49	20.39
BFN-SOLVER++2 (OURS)	<u><b>55.87</b></u>	<b>43.63</b>	<b>33.01</b>	27.92	23.89	19.48	20.51

However, the optimal distribution in the above proposition depends on the mean and the covariance matrix of the data distribution, which are still unknown. Fortunately, we find that for sufficiently small  $\eta$ , the effect of data-dependent terms is negligible. Since when  $\eta \rightarrow 0$  we know  $\gamma(t) \rightarrow 0$  and by the proposition the optimal variance is  $\gamma(t)(1 - \gamma(t)) + o(\gamma(t))$ , so a typical sample from the optimal distribution has the norm  $\sqrt{D\gamma(t)(1 - \gamma(t))}$ , which dominates the optimal mean  $\gamma(t)\mathbf{m}_{data}$  as  $\gamma(t) \rightarrow 0$ . Therefore, we suggest the data-independent distribution  $\mathcal{N}(\mathbf{0}, \gamma(t)(1 - \gamma(t))\mathbf{I})$  as the initial distribution.

We note that although the above proposition is proved for the BFN with continuous data, a similar result also holds for discrete data.

### C.2. Final Step of Sampling

Theoretically, we need to solve the reverse SDEs or ODEs from time  $1 - \eta$  to 0 to generate samples. In practice, the BFN sampler adds a step where it uses the state obtained from the initial  $n$  iterations, denoted by  $\theta_n$ , to run the network an additional time. The output from this final network is then used as the final sample. We follow this sampling trick as our default sampling method at the final step.

### C.3. User Study for Text Generation

In this section, we give the details of the user study designed to compare the quality of text generated from BFN-Solvers and BFN from the human perspective. We instruct participants to select the text samples they perceive as higher quality, providing them with authentic examples from the test dataset to serve as benchmarks for their evaluations. For each study, we collect 500 human answers from 10 participants to ensure a sufficient sample size to analyze the preferences.

## D. Additional results

### D.1. Additional Results on Continuous Dataset

In this section, we present the detailed results of Sec. 6.2. As shown in Tab. 2, our proposed methods obtain the best results under all NFE.

### D.2. Additional Results of Discrete Dataset

In this section, we present the detailed results of Sec. 6.3. As shown in Tab. 3, our best solver significantly outperforms the original BFN sampler with a few (e.g., 10) NFEs under the SA metric.

### D.3. Analysis of $\eta$ on Continuous Data

In this section, we provide an analysis of the hyperparameter  $\eta$  for continuous data. According to the analyses in Appendix C.1, we start sampling from the approximate prior distribution  $\tilde{p}(\boldsymbol{\mu}(1 - \eta)) = \mathcal{N}(\mathbf{0}, \gamma(1 - \eta)(1 - \gamma(1 - \eta))\mathbf{I})$ . Firstly, we present the FID results on the continuous CIFAR-10 dataset. We show the results with 50 and 500 NFE for efficiency. As shown in Tab. 5, excessively small or large values of  $\eta$  detrimentally affect image generation quality.

In addition, we found that SDE-based method (i.e., SDE-BFN-Solver++2) is less sensitive to  $\eta$  compared to ODE-based methods (i.e., BFN-Solver++1, BFN-Solver++2). Assuming the discretization error is negligible, Nie et al. (2023) theoretically elucidates why SDE-based samplers outperform ODE-based samplers in sampling from the approximate prior



Table 3. **Text generation results on discrete text8 dataset.** Sampling quality is measured by SA  $\uparrow$ , varying the number of function evaluations (NFE). For instance, we underline the result of BFN at 1000 steps and our SDE-BFN-Solvers2 solver at 50 steps, where we achieve a speed-up of 20 times.

NFE	10	12	20	30	40	50	100	200	1000
BFN	66.27	70.73	79.54	82.65	84.01	84.88	86.14	86.50	<u>86.69</u>
SDE-BFN-SOLVER2 (OURS)	80.29	82.39	<b>85.27</b>	<b>86.03</b>	<b>86.63</b>	<b>86.82</b>	<b>87.04</b>	<b>86.74</b>	<b>86.78</b>
BFN-SOLVER1 (OURS)	78.61	80.21	82.46	83.76	84.18	84.53	85.25	85.39	85.61
BFN-SOLVER2 (OURS)	<b>82.27</b>	<b>83.23</b>	84.46	85.12	85.34	85.46	85.53	85.57	85.60

Table 4. **Ablation of the categorical sampling (CS) step in the BFN sampler on discrete data.** We present the SA  $\uparrow$ .

NFE	10	12	20	30	40	50	100	200	1000
BFN	66.27	70.73	79.54	82.65	84.01	84.88	<b>86.14</b>	86.50	86.69
BFN w/o CS (OURS)	<b>70.67</b>	<b>73.63</b>	<b>80.86</b>	<b>83.26</b>	<b>84.27</b>	<b>84.90</b>	86.11	<b>86.76</b>	<b>86.77</b>

Table 5. **Empirical study of  $\eta$ .** We present the FID  $\downarrow$  results of image generation on continuous CIFAR-10 dataset varying  $\eta$ .

VALUES OF $\eta$	NFE=50			NFE=500		
	0.01	0.001	0.0001	0.001	0.0001	0.00001
BFN-SOLVER++1 (OURS)	50.79	48.68	111.14	21.49	49.13	220.37
BFN-SOLVER++2 (OURS)	48.29	43.84	106.44	19.48	45.95	214.17
SDE-BFN-SOLVER++2 (OURS)	41.23	36.13	34.29	12.33	14.49	27.39

Table 6. **Empirical study of  $\eta$ .** We present the SA  $\uparrow$  results of text generation on continuous text8 dataset varying  $\eta$ .

VALUES OF $\eta$	NFE=20			NFE=200		
	0.01	0.001	0.0001	0.01	0.001	0.0001
BFN-SOLVER1 (OURS)	82.56	82.39	82.46	85.40	85.38	85.39
BFN-SOLVER2 (OURS)	84.46	84.43	84.46	85.57	85.56	85.52
SDE-BFN-SOLVER2 (OURS)	84.98	85.28	85.27	86.65	86.68	86.74

distribution. This provides a theoretical foundation for our observations.

At last, as present in Sec. 6.2 and Tab. 2, after tuning  $\eta$ , ODE-based samplers still outperform SDE-based samplers and BFN baseline with a few NFEs (e.g., 10) under sample quality.

#### D.4. Analysis of $\eta$ on Discrete Data

In this section, we provide an analysis of hyperparameter  $\eta$  for discrete data. According to the analyses in Appendix C.1, we start sampling from the approximate prior distribution  $\tilde{p}(\mathbf{z}(1 - \eta)) = \mathcal{N}(\mathbf{0}, K\beta(1 - \eta)\mathbf{I})$  which works well empirically. As shown in Tab. 6, For lower NFEs(20) or higher NFEs(200), the variation in  $\eta$  shows minor differences in SA for BFN-Solver1, BFN-Solver2 and SDE-BFN-Solver2. This suggests that the choice of  $\eta$  does not significantly impact the performance of these solvers on discrete data.

#### D.5. Random Samples

Additional sampling results on CIFAR-10 are shown in Fig. 7.

Additional sampling results on text8 are shown in Figs. 8 and 9.



Figure 7. Randomly generated images by BFN and BFN-Solvers (ours) with **50**, **200**, **1000** NFEs, using the same pre-trained model on CIFAR-10.

ected\_a\_la\_a\_capo\_off\_one\_diary\_about\_vein\_ille\_ramp\_in\_nine\_two  
\_one\_could\_solutely\_be\_credited\_on\_becoming\_alro\_terrorist\_high\_  
rankengeros\_mikara\_even\_tool\_sword\_bitter\_played\_the\_when\_you\_di  
e\_teuru\_you\_re\_not\_wisdom\_but\_you\_used\_must\_choose\_to\_feel\_why\_y

of\_israel\_against\_throne\_festiny\_sadara\_meign\_voi\_gerand\_crisis\_  
of\_the\_killing\_cowkey\_he\_isamy\_thu\_serpent\_man\_of\_the\_us\_three\_r  
d\_indoorh\_gundams\_limz\_mult\_wied\_ropes\_and\_univ\_used\_gundam\_line  
s\_external\_links\_downtille\_rod\_yamatof\_at\_ken\_cruftweb\_site\_que

lligisms\_et\_six\_zero\_ff\_is\_noose\_the\_first\_was\_to\_form\_a\_more\_bi  
tter\_phase\_the\_oxoset\_like\_musical\_airs\_in\_the\_final\_phase\_of\_th  
e\_foster\_and\_thu\_pricks\_out\_his\_exariments\_von\_in\_the\_lifte\_steo  
\_of\_alp\_death\_sea\_in\_the\_baltic\_and\_inudanu\_trees\_iv\_five\_five\_g

rties\_university\_toys\_the\_no\_tops\_newly\_for\_most\_people\_than\_are  
\_the\_speakers\_and\_hike\_inside\_to\_upper\_c\_and\_even\_part\_up\_to\_lea  
d\_ridge\_longests\_but\_face\_can\_not\_be\_heard\_only\_a\_small\_number\_o  
f\_people\_beating\_by\_methane\_steen\_has\_taken\_no\_one\_to\_see\_actual

e\_got\_the\_fry\_prizes\_en\_sonor\_a\_fmz\_the\_shyierkin\_the\_brothers\_e  
rich\_hormanut\_olipy\_meducine\_which\_originated\_as\_one\_of\_nust\_cha  
racters\_in\_the\_fimms\_sonwycke\_lihenwer\_a\_leona\_pogganut\_l\_c\_kazy  
nutt\_posz\_bord\_s\_book\_the\_dust\_mountains\_is\_a\_polish\_new\_school\_

(a) BFN

arting\_evidence\_for\_local\_schools\_initial\_archaeological\_series\_  
reputed\_roman\_raising\_power\_of\_carthage\_in\_cornwall\_or\_swynsea\_t  
enis\_references\_to\_roman\_authorities\_tacitus\_mxvi\_tull\_ad\_plani\_  
s\_honoribis\_bona\_xxcutiningles\_dag\_dig\_s\_goats\_british\_nomores\_b

\_of\_technological\_species\_wes\_know\_in\_pre\_existing\_fossil\_patter  
n\_thus\_an\_intevvergence\_of\_that\_begion\_orbiting\_the\_germ\_paraharm  
\_is\_seen\_as\_the\_method\_of\_thruvitus\_learning\_we\_still\_experience  
\_formal\_tachymulty\_which\_is\_would\_be\_turned\_upon\_by\_the\_aerologi

\_six\_one\_three\_tets\_telecan\_evil\_date\_airplane\_accident\_computer  
\_three\_four\_zero\_four\_six\_painted\_film\_drinetware\_on\_duck\_morlan  
d\_was\_flinted\_inaocho\_one\_nine\_eight\_zero\_s\_disarke\_by\_such\_air  
prices\_as\_dogative\_in\_a\_condition\_to\_ant\_humanism\_political\_spim

six\_zero\_yelena\_charariovinn\_russian\_clown\_one\_nine\_five\_four\_he  
rman\_prichheae\_kick\_writer\_don\_reves\_american\_national\_cartosn\_o  
ne\_nine\_six\_zero\_mark\_holman\_australian\_automaker\_one\_nine\_six\_z  
ero\_matt\_beard\_american\_actor\_d\_one\_nine\_nine\_zero\_one\_nine\_six\_

osts\_now\_around\_throughout\_the\_natural\_sciences\_and\_malletta\_man  
y\_universities\_in\_the\_uk\_and\_were\_scored\_for\_postby\_a\_causea\_gav  
edeup\_to\_more\_than\_five\_million\_as\_the\_first\_post\_approved\_one\_o  
f\_one\_act\_weft\_of\_the\_million\_was\_shortinesed\_by\_neal\_collidge\_p

(c) BFN-Solver1 (ours)

first\_published\_in\_for\_instance\_the\_earliest\_of\_the\_word\_usage\_i  
n\_dext\_was\_the\_si\_unit\_anda\_and\_the\_errosphere\_then\_pried\_was\_fo  
r\_many\_measurements\_such\_as\_sound\_unit\_or\_soive\_pressure\_the\_hyp  
othetical\_inverses\_of\_extensive\_note\_as\_there\_are\_no\_clear\_units

him\_and\_manifestation\_most\_belonging\_to\_the\_western\_yinying\_grou  
p\_the\_chinese\_jeers\_and\_the\_pit\_the\_petin\_jeers\_in\_the\_center\_th  
ey\_are\_mainly\_in\_slated\_off\_parts\_of\_the\_world\_kehak\_and\_council  
\_the\_jews\_reap\_several\_name\_aliers\_judlism\_bank\_of\_scerpian\_seta

ed\_with\_a\_car\_also\_known\_as\_cats\_in\_hong\_kong\_pass\_the\_bus\_syste  
m\_directly\_from\_kung\_hongpass\_a\_hong\_kong\_taiwan\_taxi\_toualin\_ta  
le\_an\_ecroke\_will\_be\_emergencies\_purchase\_his\_tale\_s\_used\_disusi  
\_e\_flowers\_in\_ease\_of\_swimming\_many\_people\_are\_used\_to\_hours\_on\_

calvin\_s\_list\_for\_lymia\_sells\_his\_father\_was\_going\_a\_snapper\_and  
\_crally\_for\_the\_joysof\_the\_then\_shark\_falls\_for\_his\_mob\_honest\_  
roundings\_of\_customs\_penalties\_he\_attacks\_him\_on\_his\_way\_to\_taxi  
cal\_raysel\_warning\_him\_troubled\_by\_the\_back\_of\_a\_bumper\_ry\_chair

joint\_committee\_on\_american\_fair\_land\_activists\_pro\_ott\_avoided\_  
voting\_and\_also\_in\_a\_good\_letter\_now\_end\_of\_the\_campaign\_two\_zer  
o\_zero\_one\_wiscoe\_nimes\_bob\_book\_and\_franklin\_roosevelt\_and\_near  
ly\_the\_four\_zero\_zero\_more\_hour\_four\_th\_edition\_bob\_storie\_arise

(b) SDE-BFN-Solver2 (ours)

arting\_evidence\_for\_local\_schools\_initial\_archaeological\_series\_  
reputed\_roman\_raising\_power\_of\_carthage\_in\_cornwall\_or\_swanssea\_t  
unis\_references\_to\_roman\_authorities\_tacitus\_xxvi\_tull\_ad\_plany\_  
s\_honorifis\_to\_a\_positing\_elen\_dam\_dia\_s\_goa\_daughter\_from\_red\_b

\_of\_technological\_species\_responds\_to\_pre\_existing\_fossil\_patter  
n\_thus\_an\_intelisgence\_of\_that\_begins\_orbiting\_the\_ierm\_paraharm  
\_is\_seen\_as\_the\_netsup\_of\_th\_ss\_this\_viewing\_we\_still\_experience  
\_formal\_tachymuncy\_which\_is\_would\_be\_turned\_upon\_by\_the\_aerologi

\_six\_one\_three\_mets\_were\_an\_evil\_date\_airplane\_accident\_at\_pipot  
\_three\_four\_zero\_four\_six\_faineen\_killed\_when\_are\_on\_duck\_morlan  
d\_was\_fainted\_inancho\_one\_nine\_eight\_zero\_indiearingly\_such\_air  
prices\_as\_invasive\_in\_a\_police\_enclosant\_pacafism\_political\_crim

ive\_zero\_yelena\_charariovinn\_russian\_clown\_one\_nine\_five\_four\_go  
rdon\_priest\_am\_kick\_writer\_the\_rives\_american\_national\_tartist\_o  
ne\_nine\_six\_zero\_mark\_holman\_australian\_automaker\_one\_nine\_six\_z  
ero\_matt\_beard\_american\_actor\_d\_one\_nine\_nine\_zero\_one\_nine\_six\_

osts\_now\_around\_throughout\_the\_natural\_sciences\_and\_mauletta\_man  
y\_universities\_in\_the\_uk\_and\_were\_scored\_for\_football\_causes\_gav  
ed\_up\_to\_more\_than\_five\_million\_as\_the\_first\_post\_approved\_one\_o  
f\_one\_two\_zero\_of\_the\_million\_was\_shortinesed\_by\_seal\_collidge\_p

(d) BFN-Solver2 (ours)

Figure 8. Randomly generated texts by BFN and BFN-Solvers (ours) with 10 NFEs, using the same pre-trained models on text8.



b\_one\_four\_eight\_eight\_one\_six\_two\_nine\_albrecht\_spener\_german\_t  
heologian\_b\_one\_five\_eight\_zero\_one\_six\_three\_five\_aighaz\_mahams  
udart\_king\_of\_kochi\_singh\_zugu\_warner\_and\_politician\_b\_one\_five\_  
five\_two\_one\_six\_four\_five\_ashokawa\_sunemuna\_japanese\_silence\_is

s\_the\_babylonian\_talmud\_to\_one\_more\_than\_the\_first\_half\_of\_six\_t  
wo\_five\_with\_a\_recent\_interpretat\_to\_date\_the\_decline\_of\_the\_pop  
ulations\_from\_the\_first\_time\_however\_the\_third\_cultural\_branch\_r  
epresenting\_a\_flux\_in\_the\_number\_of\_workers\_appears\_to\_have\_stro

e\_citizens\_trained\_under\_personal\_safety\_laws\_less\_than\_one\_zero  
years\_of\_potency\_but\_strictly\_speaking\_the\_livis\_taskforce\_used  
\_for\_magistations\_amounting\_to\_older\_magistrates\_nineteen\_years\_  
later\_working\_on\_home\_education\_and\_and\_various\_reforms\_across\_t

ne\_at\_least\_in\_daily\_use\_of\_the\_important\_lakes\_named\_after\_mann  
opies\_image\_flag\_of\_this\_island\_state\_of\_macedonia\_syria\_s\_itali  
an\_shi\_h\_bela\_divided\_sixth\_horns\_edegis\_turke\_howkver\_kjan\_bent  
unhorle\_deneves\_mykior\_subias\_vlashbirerojs\_in\_one\_five\_seven\_tw

ld\_county\_soub\_army\_britain\_christian\_democratic\_party\_moral\_phi  
losophy\_scottish\_philosophers\_scaptons\_dualistic\_philosophers\_ge  
igntonians\_physicians\_george\_burnell\_leeves\_physicists\_british\_m  
athematicians\_mathematical\_theory\_of\_non\_human\_worlds\_j\_barnard\_

(a) BFN

cult\_references\_parbula\_in\_english\_december\_two\_zero\_zero\_two\_ve  
rsion\_two\_zero\_zero\_two\_with\_illustrations\_idea\_users\_dictionary  
\_of\_colligions\_devil\_org\_christianism\_from\_the\_beginner\_translat  
ed\_into\_the\_web\_ten\_quotes\_from\_hippocrate\_richard\_paul\_geneva\_w

her\_sical\_dating\_is\_found\_byron\_s\_date\_ch\_one\_nine\_six\_titus\_of\_  
macedon\_had\_decided\_to\_have\_a\_babylonian\_calendar\_in\_february\_on  
e\_two\_zero\_five\_so\_according\_to\_some\_sical\_dating\_the\_greek\_adda  
ration\_of\_five\_three\_nine\_one\_bc\_five\_three\_three\_one\_cut\_him\_bu

ion\_three\_etc\_and\_three\_d\_seven\_zero\_five\_four\_nine\_nine\_seven\_f  
ive\_zero\_one\_riemann\_euler\_german\_mathematician\_and\_an\_astronome  
r\_who\_worked\_in\_the\_one\_eight\_th\_century\_after\_two\_fields\_gone\_i  
n\_line\_mount\_zero\_and\_a\_bright\_orbitary\_more\_commonly\_moon\_numbe

arting\_evidence\_for\_roman\_schools\_initial\_archaeological\_series\_  
reputed\_roman\_raising\_power\_of\_carthage\_in\_cornwall\_or\_swynsea\_t  
unis\_references\_to\_roman\_authorities\_tacitus\_christyila\_ad\_pliny\_  
s\_honorifis\_bond\_xxcutiningles\_dag\_dig\_s\_goats\_british\_romores\_b

\_of\_technological\_species\_responds\_to\_pre\_existing\_fossil\_patter  
n\_thus\_an\_intelligence\_of\_that\_begins\_orbiting\_the\_germ\_paradorm  
\_is\_seen\_as\_the\_extent\_of\_which\_the\_permlike\_we\_still\_experience  
\_fossil\_beings\_and\_which\_we\_would\_be\_turned\_upon\_by\_the\_aerologi

(c) BFN-Solver1 (ours)

the\_chambers\_divisive\_among\_stephen\_shepherd\_who\_is\_still\_in\_con  
tent\_for\_the\_production\_nicholas\_de\_france\_in\_one\_nine\_nine\_nine  
\_trial\_by\_maria\_dizvosa\_two\_zero\_zero\_one\_the\_shadow\_in\_the\_simp  
sons\_two\_zero\_zero\_three\_trial\_by\_david\_rockland\_two\_zero\_zero\_f

ork\_after\_its\_commercial\_and\_initial\_acquisition\_by\_fan\_of\_prese  
nt\_enlarged\_by\_graphical\_publishers\_that\_now\_it\_was\_hired\_for\_no  
\_third\_idea\_use\_such\_work\_a\_large\_portion\_of\_microsoft\_specifica  
lly\_an\_adaptation\_an\_open\_tp\_program\_as\_a\_requirement\_for\_the\_su

er\_than\_the\_now\_here\_mawes\_psa\_led\_worldwide\_however\_motorcycle\_  
became\_less\_seen\_in\_formula\_one\_eight\_factories\_by\_one\_nine\_one\_  
one\_was\_open\_production\_until\_one\_nine\_one\_three\_designer\_niel\_j  
ones\_maldini\_made\_him\_up\_a\_psa\_car\_and\_in\_one\_nine\_one\_four\_bega

n\_man\_mumahuadeed\_wrote\_that\_this\_may\_say\_he\_has\_made\_of\_a\_horse  
bow\_against\_use\_which\_is\_wise\_what\_to\_make\_and\_what\_did\_isn\_t\_do  
\_nothing\_about\_it\_with\_mind\_and\_what\_used\_to\_exist\_in\_the\_islami  
c\_world\_philosophy\_canton\_p\_two\_five\_five\_for\_now\_with\_mind\_alwa

ernisler\_german\_pianist\_d\_one\_nine\_four\_zero\_one\_nine\_zero\_eight  
\_lobe\_te\_seitel\_canadian\_bandleader\_d\_one\_nine\_six\_eight\_one\_nin  
e\_one\_three\_samuel\_pahn\_american\_author\_d\_one\_nine\_four\_three\_on  
e\_nine\_one\_four\_thomas\_dolf\_beck\_american\_musician\_and\_band\_desi

(b) SDE-BFN-Solver2 (ours)

cult\_references\_parbula\_in\_english\_december\_two\_zero\_zero\_two\_ve  
rsion\_two\_zero\_zero\_two\_with\_illustrations\_idea\_users\_dictionary  
\_of\_colligions\_devil\_org\_christianism\_from\_the\_beginner\_translat  
ed\_into\_the\_web\_ten\_quotes\_from\_hippocrate\_richard\_paul\_geneva\_w

her\_sical\_dating\_is\_found\_byron\_s\_date\_ch\_one\_nine\_six\_titus\_of\_  
macedon\_had\_decided\_to\_have\_a\_babylonian\_calendar\_in\_february\_on  
e\_two\_zero\_five\_so\_according\_to\_some\_sical\_dating\_the\_greek\_adda  
ration\_of\_five\_three\_nine\_one\_bc\_five\_three\_three\_one\_cut\_him\_bu

ion\_three\_etc\_and\_three\_d\_seven\_zero\_five\_four\_nine\_nine\_seven\_f  
ive\_zero\_one\_riemann\_euler\_german\_mathematician\_and\_an\_astronome  
r\_who\_worked\_in\_the\_one\_eight\_th\_century\_after\_two\_fields\_gone\_i  
n\_line\_mount\_zero\_and\_a\_bright\_orbitary\_more\_commonly\_moon\_numbe

arting\_evidence\_for\_roman\_schools\_initial\_archaeological\_series\_  
reputed\_roman\_raising\_power\_of\_carthage\_in\_cornwall\_or\_swynsea\_t  
unis\_references\_to\_roman\_authorities\_tacitus\_christyila\_ad\_pliny\_  
s\_honorifis\_bond\_xxcutiningles\_dag\_dig\_s\_goats\_british\_romores\_b

\_of\_technological\_species\_responds\_to\_pre\_existing\_fossil\_patter  
n\_thus\_an\_intelligence\_of\_that\_begins\_orbiting\_the\_germ\_paradorm  
\_is\_seen\_as\_the\_extent\_of\_which\_the\_permlike\_we\_still\_experience  
\_fossil\_beings\_and\_which\_we\_would\_be\_turned\_upon\_by\_the\_aerologi

(d) BFN-Solver2 (ours)

Figure 9. Randomly generated texts by BFN and BFN-Solvers (ours) with **1000** NFEs, using the same pre-trained models on text8.