Provably Convergent Schrödinger Bridge with Applications to Probabilistic Time Series Imputation

Yu Chen [* 1] **Wei Deng** [* 1] **Shikai Fang** [* 2] **Fengpei Li** [* 1] **Nicole Tianjiao Yang** [3]
Yikai Zhang [1] **Kashif Rasul** [1] **Shandian Zhe** [2] **Anderson Schneider** [1] **Yuriy Nevmyvaka** [1]

Abstract

The Schrödinger bridge problem (SBP) is gaining increasing attention in generative modeling and showing promising potential even in comparison with the score-based generative models (SGMs). SBP can be interpreted as an entropy-regularized optimal transport problem, which conducts projections onto every other marginal alternatingly. However, in practice, only approximated projections are accessible and their convergence is not well understood. To fill this gap, we present a first convergence analysis of the Schrödinger bridge algorithm based on approximated projections. As for its practical applications, we apply SBP to probabilistic time series imputation by generating missing values conditioned on observed data. We show that optimizing the transport cost improves the performance and the proposed algorithm achieves the state-of-the-art result in healthcare and environmental data while exhibiting the advantage of exploring both temporal and feature patterns in probabilistic time series imputation.

1. Introduction

Time series data is extensively studied in various fields such as finance (Xiong & Pelger, 2023), healthcare (Silva et al., 2012), and meteorology. However, incomplete or partial observations, equipment failures, and human errors may inevitably lead to the missing value problem, severely limiting the interpretation of the time series. For instance, the inherent illiquidity of certain assets can result in the occurrence of missing values, which in turn impacts our ability to devise reliable trading strategies (Christoffersen et al., 2017). In populations of Intensive Care Units (ICUs), predicting mortality rates based on time-series observations of vital signs is essential (Silva et al., 2012). However, the presence of missing data has greatly limited the efficacy of medications and surgical treatments.

One standard approach to tackle such a problem is to leverage score-based generative models (SGMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021a;b), which propose to recover the data distribution through a backward process that estimates the scores of posterior distributions conditioned on the observed data. This conditional nature directly motivates the study of Conditional Score-based Diffusion models for Imputation (CSDI) (Tashiro et al., 2021). CSDI is able to learn the temporal-feature patterns well and achieves state-of-the-art performance in probabilistic time series imputation. However, transporting between terminal distributions is often quite expensive for SGMs and CSDI, which requires extensive computations and hyperparameter tuning. As such, a more efficient algorithm is needed to reduce the transport cost.

The Schrödinger bridge problem (SBP) was initially proposed to solve problems in quantum mechanics and can be transformed into the entropy-regularized optimal transport (EOT) (De Bortoli et al., 2021; Léonard, 2014a; Chen et al., 2021b; Nutz & Wiesel, 2022). Solving the EOT formulation gives rise to the iterative proportional fitting (IPF) algorithm (Kullback, 1968; Ruschendorf, 1995), which provides a principled paradigm to minimize the transport cost and facilitates the estimation of score functions to generate samples of higher quality; SBP further enables the generalization of linear Gaussian priors to non-linear families with more acceleration potential (Chen et al., 2021a; Bunne et al., 2023; Pavon et al., 2021; Deng et al., 2020; 2022).

Despite the theoretical potential, the existing SBP-based generative models assume we can obtain the exact projections for the IPF algorithm, but in practice, it is often only approximated by deep neural networks (De Bortoli et al., 2021) or Gaussian processes (Vargas et al., 2021). In order to fill the gap, we extend the IPF algorithm by allowing for the approximated projections and refer to it as the approxi-

mate IPF (aIPF) algorithm; we further conduct theoretical analysis for aIPF based on the optimal transport theory, which deepens the understanding of training budgets in score approximations. Empirically, we apply the SBP-based generative models to probabilistic time series imputation and demonstrate that minimizing the transport cost improves performance. We summarize our contributions as follows:

- We show a first convergence analysis for Schrödinger bridge with approximated projections and characterize the relation between training errors and the number of iterations. Our theory motivates future research for devising provably convergent Schrödinger bridge (SB) algorithms and paves the way for understanding when SB is faster than SGMs. To bridge the gap between theoretical understanding and practical algorithms, we also draw connections between the aIPF algorithm and the divergence-based likelihood training of forward and backward stochastic differential equations (FB-SDEs).

- We apply the Schrödinger bridge algorithm to probabilistic time series imputation. We show that optimizing the transport cost visibly improves the performance on synthetic data and achieves the state-of-the-art performance on real-world datasets.

2. Related Work

Schrödinger Bridge   Schrödinger bridge problem (SBP) is known for the quantum mechanics formulation and is closely related to stochastic optimal control (SOC) (Chen & Georgiou, 2016; Pavon et al., 2021; Caluya & Halder, 2022) and optimal transport (Peyré & Cuturi, 2019). Recent works leverage SBP for generative modeling (De Bortoli et al., 2021; Wang et al., 2021) and explore theoretical properties (Nutz & Wiesel, 2022; Ghosal et al., 2022; Khrulkov & Oseledets, 2023; Lavenant & Santambrogio, 2022). Shi et al. (2022) apply the amortized formulation to model the conditional SBP for images and state-space models. Chen et al. (2022) propose likelihood training for SBP approximation based on divergence objectives (Hutchinson, 1989; Grathwohl et al., 2019) and forward-backward stochastic differential equations (FB-SDEs) (Ma & Yong, 2007); similar results are shown in Vargas et al. (2021).

Time-series Imputations via Generative Models   Multivariate time series imputation is challenging because of the temporal-feature dependencies and the irregular locations of missing values. To handle these issues, several recent works use deep generative learning and conditional sampling to achieve competitive performance. Generative techniques in these works include Gaussian processes (Dürichen et al., 2014), VAEs (Fortuin et al., 2020), neural ODEs (Rubanova et al., 2019; de Bézenac et al., 2020; Kidger et al., 2020),

neural SDEs (Li et al., 2020; Deng et al., 2021), and GANs (Luo et al., 2019). Imputation methods based on recurrent networks or attention networks can be found in (Che et al., 2018; Cao et al., 2018; Shukla & Marlin, 2021). Curvature flow methods are potentially applicable (Malladi & Sethian, 1995).

3. Preliminaries

3.1. Likelihood Training of SGMs

The score-based generative models (SGMs (Song et al., 2021b)) have become the go-to framework for generative models. SGMs first inject noise into the data and then recover it from a backward process (Anderson, 1982)

$$\mathrm{d}\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t, \tag{1a}$$
$$\mathrm{d}\mathbf{x}_t = \left[\boldsymbol{f}(\mathbf{x}_t, t) - g(t)^2 \nabla \log p_t(\mathbf{x}_t)\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}_t, \tag{1b}$$

where $\{\mathbf{x}_t\}_{t=0}^T \in \mathbb{R}^d$ [§], $\mathbf{x}_0 \sim p_{\text{data}}$, and $\mathbf{x}_T \sim p_{\text{prior}}$; $\boldsymbol{f} \equiv \boldsymbol{f}(\mathbf{x}_t, t)$ is the vector field; $g \equiv g(t)$ is the diffusion term; $\mathbf{w}_t$ is the standard Brownian motion; $\bar{\mathbf{w}}_t$ is a Brownian motion with time moving backward from T to 0; p_t is the marginal density of the forward process (1a) at time t. The score function $\nabla \log p_t(\cdot)$ is approximated via a model $s_\theta(\cdot, t)$; p_{data} is simulated via the backward process (1b) starting at $\mathbf{x}_T$. SGMs (Ho et al., 2020) aim to train $s_\theta(\cdot, t)$ by minimizing the mean squared error between the ground-truth score and estimator such that $\mathbb{E}\left[\lambda(t)\|s_\theta(\mathbf{x}_t, t) - \nabla \log p_t(\mathbf{x}_t \mid \mathbf{x}_0)\|_2^2\right]$, where the weight $\lambda(t)$ is set manually. Song et al. (2021a) proposes to maximize the likelihood to learn $s_\theta(\mathbf{x}, t)$ such that

$$\log p_0^{\text{SDE}}(\mathbf{x}_0) \geq \mathbb{E}_{p_{0T}(\cdot|\mathbf{x}_0))}\left[\log p_T(\mathbf{x}_T)\right]$$
$$- \frac{1}{2}\int_0^T \mathbb{E}_{p_{0t}(\cdot|\mathbf{x}_0)}\left[g^2\|\mathbf{s}_t\|_2^2 + 2\nabla \cdot (g^2\mathbf{s}_t - \boldsymbol{f})\right]\mathrm{d}t,$$

where $s_t = s_\theta(\mathbf{x}_t, t)$, $p_{0t}(\cdot|\mathbf{x}_0) = p_{0t}(\mathbf{x}_t|\mathbf{x}_0)$ stands for the conditional density of $\mathbf{x}_t$, which evolves with the trajectory of (1a). The inequality becomes an equality if the estimator s_t exactly matches the score function. Thus, optimizing the lower bound provides an efficient scheme to maximize the data likelihood.

3.2. Schrödinger Bridge Problem

Even though SGMs have demonstrated success in generative models, they still suffer from transport inefficiency. A *long evolving time T* of the forward process (1a) is required to facilitate the score estimation and guarantee that $\mathbf{x}_t$ will converge close to a prior distribution. Besides, the choice of priors is repeatedly *constrained to Gaussian* and further limits the acceleration potential. To tackle this issue, the

---

[§]d is the data dimension and can be reshaped to other formats.

dynamical Schrödinger Bridge problem (SBP) aims to solve

$$\inf_{\mathbb{P} \in \mathcal{D}(\mu_\star, \nu_\star)} \mathrm{KL}(\mathbb{P}|\mathbb{Q}), \quad (2)$$

where $\mathcal{D}(\mu_\star, \nu_\star)$ denotes the space of *path measures* with marginal probability measures $\mu_\star$ and $\nu_\star$ at time $t = 0$ and $t = T$, respectively; $\mathbb{Q}$ is the prior measure, usually induced by Brownian motion or Ornstein-Uhlenbeck process; $\mathrm{KL}(\cdot|\mathbb{Q})$ denotes the KL divergence with respect to the measure $\mathbb{Q}$.

The dynamical SBP can be interpreted from stochastic optimal control (SOC) (see section 4.4 in Chen et al. (2021a))

$$\inf_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}\left\{ \int_0^T \frac{1}{2}\|\boldsymbol{u}(\mathbf{x}_t, t)\|_2^2 \mathrm{d}t \right\} \quad (3)$$
$$\text{s.t. } \mathrm{d}\mathbf{x}_t = [\boldsymbol{f}(\mathbf{x}_t, t) + g(t)\boldsymbol{u}(\mathbf{x}_t, t)]\,\mathrm{d}t + \sqrt{2\varepsilon}g(t)\mathrm{d}\mathbf{w}_t$$
$$\mathbf{x}_0 \sim \mu_\star(\cdot), \quad \mathbf{x}_T \sim \nu_\star(\cdot),$$

where $\mathcal{U}$ is the control set $\boldsymbol{u} : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$; the state-space is $\mathbb{R}^d$ and is sometimes omitted; the expectation is taken w.r.t the joint state PDF $\rho(\mathbf{x}, t)$; ε is a regularizer.

4. Provably Convergent Schrödinger Bridge

Diffusion models have shown superiority in generative models and time series imputation, which motivate interesting theoretical works (Lee et al., 2022; Chen et al., 2023; De Bortoli et al., 2021; Koehler et al., 2023). As a theoretical ideal candidate, Schrödinger bridge also has gained tremendous attention (De Bortoli et al., 2021; Vargas et al., 2021; Wang et al., 2021; Chen et al., 2022), however, the practical theory has not been studied in the literature.

To bridge the gap between theory and practice, we initiate the convergence study of the practical Schrödinger bridge algorithm based on general cost functions and highlight the connections between SBP, EOT, and FB-SDEs.

4.1. Schrödinger Bridge: from Dynamic to Static

By applying the disintegration of measures (Léonard, 2014b), the chain rule (De Bortoli et al., 2021) for the KL divergence for the dynamical SBP (2) follows

$$\mathrm{KL}(\mathbb{P}|\mathbb{Q}) = \mathrm{KL}(\pi|\mathcal{G}) + \iint \mathrm{KL}(\mathbb{P}_\pi|\mathbb{Q}_\mathcal{G})\mathrm{d}\pi(\mathbf{x}_0, \mathbf{x}_T).$$

where $\pi := (\mu_\star, \nu_\star)$ is a coupling with marginals $\mu_\star$ and $\nu_\star$; $\mathcal{G}$ is a Gibbs measure: $\mathrm{d}\mathcal{G} \propto e^{-c_\varepsilon}\mathrm{d}(\mu_\star \otimes \nu_\star)$; c_ε is a cost function in Eq.(26); $\otimes$ is the product measure; the marginals of $\mathbb{P}$ (or $\mathbb{Q}$) at $t = 0$ and T follow from $\mu_\star$ and $\nu_\star$; $\mathbb{P}_\pi$ (or $\mathbb{Q}_\mathcal{G}$) denotes the conditional probability of the rest information of $\mathbb{P}$ (or $\mathbb{Q}$) given π (or $\mathcal{G}$) (De Bortoli, 2022).

Setting $\mathbb{P}_\pi = \mathbb{Q}_\mathcal{G}$, we have the *static* SBP with a unique

coupling $\pi_\star$ (see Lemma 1 in Appendix A.2):

$$\pi_\star = \arg\min_{\pi \in \Pi(\mu_\star, \nu_\star)} \mathrm{KL}(\pi|\mathcal{G}), \quad (4)$$

where $\Pi(\mu_\star, \nu_\star)$ is the set of couplings with marginals $\mu_\star$ and $\nu_\star$. Moreover, the static SBP yields a structural representation for (4) (Peyré & Cuturi, 2019; Nutz, 2022):

$$\mathrm{d}\pi_\star(\mathbf{x}, \mathbf{y}) = e^{\varphi_\star(\mathbf{x}) + \psi_\star(\mathbf{y}) - c_\varepsilon(\mathbf{x},\mathbf{y})}\mathrm{d}(\mu_\star \otimes \nu_\star),$$

where $\varphi_\star$ and $\psi_\star$ are the Schrödinger potential functions.

4.2. From Static SBP to Entropic Optimal Transport

Next, the equivalence between the static SBP and entropic optimal transport (EOT) follows that:

$$\mathrm{KL}(\pi|\mathcal{G}) = \iint \log\left(\frac{d\pi}{d(\mu_\star \otimes \nu_\star)} \frac{d(\mu_\star \otimes \nu_\star)}{d\mathcal{G}} \right) d\pi$$
$$\doteq \mathrm{KL}(\pi|\mu_\star \otimes \nu_\star) + \iint \log e^{c_\varepsilon}\,d\pi$$
$$= \iint c_\varepsilon\,d\pi + \mathrm{KL}(\pi|\mu_\star \otimes \nu_\star),$$

where $\doteq$ denotes an equality that's up to a constant. Problem (4) is equivalent to the EOT with a 1-regularizer:

$$\inf_{\pi \in \Pi(\mu_\star, \nu_\star)} \iint c_\varepsilon(\mathbf{x}, \mathbf{y})\pi(\mathrm{d}\mathbf{x}, \mathrm{d}\mathbf{y}) + 1 \cdot \mathrm{KL}(\pi|\mu_\star \otimes \nu_\star). \quad (5)$$

4.3. Approximate Iterative Proportional Fitting (aIPF)

Note that Schrödinger bridges are often computationally intractable; to tackle this issue, we resort to the stochastic optimal control formulation (3) and obtain score functions $\nabla \log \overleftarrow{\varphi}$ and $\nabla \log \overrightarrow{\psi}$ (see Eq.(25) in Appendix A.1)

$$\overleftarrow{\varphi}(\mathbf{x}) \int e^{-c_\varepsilon(\mathbf{x},\mathbf{y})}\overrightarrow{\psi}(\mathbf{y})\mathrm{d}\mathbf{y} = \frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}},$$
$$\overrightarrow{\psi}(\mathbf{y}) \int e^{-c_\varepsilon(\mathbf{x},\mathbf{y})}\overleftarrow{\varphi}(\mathbf{x})\mathrm{d}\mathbf{x} = \frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}. \quad (6)$$

Consider the following transformation

$$\overleftarrow{\varphi}(\mathbf{x}) = e^{\varphi_\star(\mathbf{x})}\frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}}, \quad \overrightarrow{\psi}(\mathbf{y}) = e^{\psi_\star(\mathbf{y})}\frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}. \quad (7)$$

Combining Eq.(6) and (7) yields *Schrödinger equations*

$$\int e^{\varphi_\star(\mathbf{x}) + \psi_\star(\mathbf{y}) - c_\varepsilon(\mathbf{x},\mathbf{y})}\mu_\star(\mathrm{d}\mathbf{x}) = 1 \quad \nu_\star\text{-}a.s. \quad (8)$$

$$\int e^{\varphi_\star(\mathbf{x}) + \psi_\star(\mathbf{y}) - c_\varepsilon(\mathbf{x},\mathbf{y})}\nu_\star(\mathrm{d}\mathbf{y}) = 1 \quad \mu_\star\text{-}a.s.. \quad (9)$$

In view of Eq.(32) in Appendix A.3, the above implies that

Eq.(8) $\Longleftrightarrow$ the first marginal of $\pi_\star$ is $\mu_\star$,

Eq.(9) $\Longleftrightarrow$ the second marginal of $\pi_\star$ is $\nu_\star$.
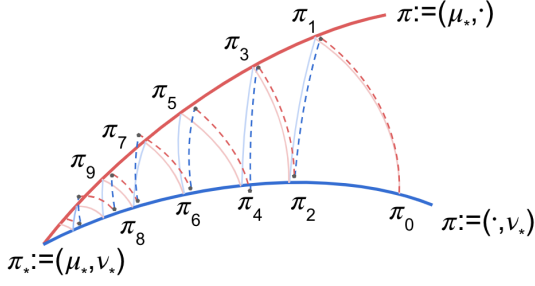
Figure 1. IPF vs aIPF. The light solid lines (IPF) show the iterates of exact projections; the dotted lines (aIPF) present the approximate projections towards $\pi_\star := (\mu_\star, \nu_\star)$.

To obtain the desired $\pi_\star := (\mu_\star, \nu_\star)$, a standard tool is the iterative proportional fitting (IPF) (also known as Sinkhorn algorithm) (Ruschendorf, 1995). The exact IPF algorithm alternatingly projects the coupling $\pi_k := (\mu_k, \nu_k)$ at iteration k to every other marginal such that for any $k \in \mathbb{N}$:

$$\mu_{2k+1} = \mu_\star, \quad \nu_{2k} = \nu_\star.$$

To wit, we solve every other marginal alternatingly and show the convergence of the marginals to the correct distribution

$$\mu_{2k} \xRightarrow{\text{convergence}} \mu_\star, \quad \nu_{2k+1} \xRightarrow{\text{convergence}} \nu_\star.$$

However, it is too expensive in practice to obtain the exact marginals $\mu_\star$ and $\nu_\star$ via Eq.(8) and (9). To solve this problem, it is inevitable to approximate the projections (numerically solved via FB-SDE in Eq.(11)) through specific tools, such as deep neural networks (De Bortoli et al., 2021; Chen et al., 2022) or Gaussian process (Vargas et al., 2021)

$$\mu_{2k+1} = \mu_{\star,k+1} \approx \mu_\star, \quad \nu_{2k} = \nu_{\star,k} \approx \nu_\star, \quad (10)$$

where $\mu_{\star,k+1}$ (or $\nu_{\star,k}$) is an approximate measure at iteration $k+1$ (or k) that is close to $\mu_\star$ (or $\nu_\star$). The approximate IPF (aIPF) is presented in Algorithm 1 and the comparison to the exact IPF is illustrated in Figure.1.

Algorithm 1 One iteration of approximate IPF (aIPF). ψ_k and φ_k denote the estimates of potential functions at iteration k. Empirically, the integral is estimated via divergence-based likelihood training of FB-SDEs in section 5.1.

$$\psi_k(\mathbf{y}) \approx -\log \int_{\mathbb{R}^d} e^{\varphi_k(\mathbf{x}) - c_\varepsilon(\mathbf{x},\mathbf{y})} \mu_\star(\mathrm{d}\mathbf{x})$$

$$\varphi_{k+1}(\mathbf{x}) \approx -\log \int_{\mathbb{R}^d} e^{\psi_k(\mathbf{y}) - c_\varepsilon(\mathbf{x},\mathbf{y})} \nu_\star(\mathrm{d}\mathbf{y}).$$

4.4. Convergence Analysis

For the convergence study, the existing (heuristic) finite sample bound (Vargas et al., 2021) conjectured that

$$\|\mu_{2k} - \mu_\star\| \lesssim \frac{1}{k} + \sum_{i=0}^{k} \epsilon K^i$$

$$\|\nu_{2k-1} - \nu_\star\| \lesssim \frac{1}{k} + \sum_{i=0}^{k} \epsilon K^i,$$

where $\|\cdot\|$ is some metric and the aIPF projection operator is assumed to be K-Lipchitz. However, ensuring $K < 1$ based on general cost functions is not trivial (Greco et al., 2023). As such, a *fundamental question* remains open

Can we ensure the approximation error is less dependent on the number of iterations k?

Answering this question yields concrete guidance on the computational complexity and tells us when the approximation error doesn't get arbitrarily worse. To achieve this target, we first lay out the following standard assumptions.

Assumption A1 (Dissipativity). *$\mu_\star$ and $\nu_\star$ satisfy the dissipative condition for some constants $m_{ds} > 0$ and $b_{ds} \geq 0$.*

$$\left\langle \mathbf{x}, -\nabla \log \frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}}(\mathbf{x}) \right\rangle \geq m_{ds}\|\mathbf{x}\|_2^2 - b_{ds}$$

$$\left\langle \mathbf{y}, -\nabla \log \frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}(\mathbf{y}) \right\rangle \geq m_{ds}\|\mathbf{y}\|_2^2 - b_{ds},$$

where $\frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}}$ and $\frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}$ are the probability densities for the probability measure $\mu_\star$ and $\nu_\star$, respectively; $\nabla \log \frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}}(\mathbf{x})$ and $\nabla \log \frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}(\mathbf{y})$ are the score functions.

Remark: The above assumption is standard and has been used in Raginsky et al. (2017), which allows the densities to be non-convex in a ball with a radius depending on b_{ds}. Notably, it also implies the log-Sobolev inequality (LSI) with a bounded constant C_{LS} (Lee et al., 2022).

Assumption A2 (ϵ-approximation). *$\nabla \log \frac{\mathrm{d}\mu_{\star,k}}{\mathrm{d}\mathbf{x}}(\mathbf{x})$ and $\nabla \log \frac{\mathrm{d}\nu_{\star,k}}{\mathrm{d}\mathbf{y}}(\mathbf{y})$ are the ϵ approximation of score functions $\nabla \log \frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}}(\mathbf{x})$ and $\nabla \log \frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}(\mathbf{y})$ at the k-th iteration, respectively*

$$\left\| \nabla \log \frac{\mathrm{d}\mu_{\star,k}}{\mathrm{d}\mathbf{x}} - \nabla \log \frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}} \right\|_\infty \leq \epsilon$$

$$\left\| \nabla \log \frac{\mathrm{d}\nu_{\star,k}}{\mathrm{d}\mathbf{y}} - \nabla \log \frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}} \right\|_\infty \leq \epsilon,$$

Such an assumption is closely related to the ϵ-accurate score approximation in De Bortoli et al. (2021); Lee et al. (2022) except that our focus is the marginals on $\mu_\star$ and $\nu_\star$ while theirs is the marginal density along the forward SDE (1a).

To further extend the score approximation assumption from L^∞ (uniformly accurate) to L^2 (in expectation), we can leverage the "bad set" idea (Lee et al., 2022) or the Girsanov theorem (Chen et al., 2023) to match the likelihood training framework better. Moreover, the errors in the two marginals do not need to be identical, and a unified ϵ is employed mainly for the sake of analytical convenience.

Assumption A3 (Lipschitz smoothness). *The score functions of marginal densities* $\nabla \log \frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}}$ *are* $\nabla \log \frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}$ *are both* L-*Lipschitz smooth.*

To sketch the proof, we first show a summation property of $\sum_{k\geq 1}^n \mathrm{KL}(\pi_k|\pi_{k-1})$ without breaking the cyclical invariance property (Ghosal et al., 2022) in Lemma 3 such that

$$\sum_{k\geq 1}^n \mathrm{KL}(\pi_k|\pi_{k-1}) \leq \mathrm{KL}(\pi_\star|\mathcal{G}) - \mathrm{KL}(\pi_0|\mathcal{G}) + O(n\epsilon).$$

Next, we prove $\mathrm{KL}(\mu_{2k}|\mu_{\star,k}) \leq \mathrm{KL}(\pi_{2k}|\pi_{2k-1})$ and $\mathrm{KL}(\nu_{\star,k}|\nu_{2k-1}) \leq \mathrm{KL}(\pi_{2k}|\pi_{2k-1}) + O(\epsilon)$, which yields

$$\sum_{k\geq 1}^n \mathrm{KL}(\mu_{2k}|\mu_{\star,k}) \leq \mathrm{KL}(\pi_\star|\mathcal{G}) - \mathrm{KL}(\pi_0|\mathcal{G}) + O(n\epsilon)$$

$$\sum_{k\geq 1}^n \mathrm{KL}(\nu_{\star,k}|\nu_{2k-1}) \leq \mathrm{KL}(\pi_\star|\mathcal{G}) - \mathrm{KL}(\pi_0|\mathcal{G}) + O(n\epsilon).$$

Moreover, we obtain an approximately monotone-decreasing property in proposition 4 as follows

$$\mathrm{KL}(\mu_{2k}|\mu_{\star,k}) \leq \mathrm{KL}(\mu_{2t+2}|\mu_{\star,k+1}) + O(\epsilon)$$

$$\mathrm{KL}(\nu_{\star,k}|\nu_{2k-1}) \leq \mathrm{KL}(\nu_{\star,k+1}|\nu_{2k+1}) + O(\epsilon).$$

Finally, combining Lemma 8 and the fact that $\mu_{\star,k}$ (or $\nu_{\star,k}$) is ϵ-close to $\mu_\star$ (or $\nu_\star$), our main theorem follows that:

Theorem 1 (Approximately Sublinear Rate for Marginals). *Given dissipative assumption A1, ϵ-approximate score assumption A2, and smoothness assumption A3, we have*

$$KL(\mu_{2k}|\mu_\star) \leq \frac{KL(\pi_\star|\mathcal{G}) - KL(\pi_0|\mathcal{G})}{k} + O(\epsilon^{\frac{1}{2}} + k^{\frac{1}{2}}\epsilon)$$

$$KL(\nu_\star|\nu_{2k-1}) \leq \frac{KL(\pi_\star|\mathcal{G}) - KL(\pi_0|\mathcal{G})}{k} + O(\epsilon^{\frac{1}{2}} + k^{\frac{1}{2}}\epsilon),$$

where the big-O notations are independent of k.

The proof is presented in Appendix B.1, which provides the first-ever evidence of the convergence of the aIPF algorithm with approximate projections. Our analysis suggests that to achieve an ϵ-accurate target, the iteration should be greater than $\Omega(\frac{1}{\epsilon})$, although early stopping may be necessary to avoid excessive perturbations. It is worth noting that *order $O(k^{\frac{1}{2}})$ is more preferable than linear-order or expansive upper bounds* in Vargas et al. (2021) (when $K \geq 1$). However, we acknowledge this result is not entirely practical

without the bounded-cost-function assumption. We believe the square-root order can be further refined by obtaining a tighter convergence rate (Ghosal & Nutz, 2022; Greco et al., 2023). This refinement can be left as future work.

Moreover, the convergence of the minimized cost (3) potentially facilitates the estimation of score functions. However, it involves a trade-off between computation and accuracy. Such a trade-off can be used to establish instances where Schrödinger bridge is faster than SGMs.

4.5. Connections between aIPF and FB-SDE

The complexity analysis of SBP hinges on the convergence of aIPF based on entropic optimal transport; however, solving the exact integrals in Algorithm 1 is far from trivial and heavily relies on score-based sampling techniques. In the next section, we present the likelihood training of SBP to connect with aIPF and build the conditional variant for applications in probabilistic time series imputations. To facilitate reading, we visualize the connections between the convergence analysis of aIPF and the likelihood training of FB-SDE in Figure 2 below.
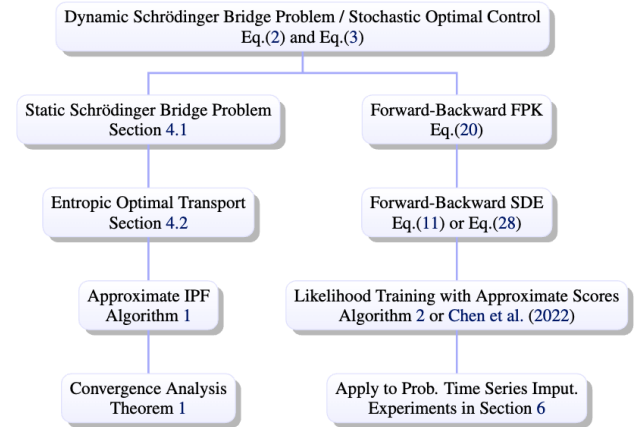

Figure 2. Relation between IPF, SBP, SOC and FB-SDE.

5. Likelihood Training with Applications to Probabilistic Time Series Imputation

In this section, we solve SBP via likelihood training of FB-SDEs and briefly present the conditional Schrödinger bridge method for probabilistic time series imputation (CSBI).

5.1. Likelihood Training via Schrödinger Bridge

Solving SBP is often intractable. We show it could be transformed into computation-friendly FB-SDEs. We sketch the main results here and detail derivations in appendix A.1.

Rewrite the SOC perspective of SBP (3) with $\varepsilon = \frac{1}{2}$ and
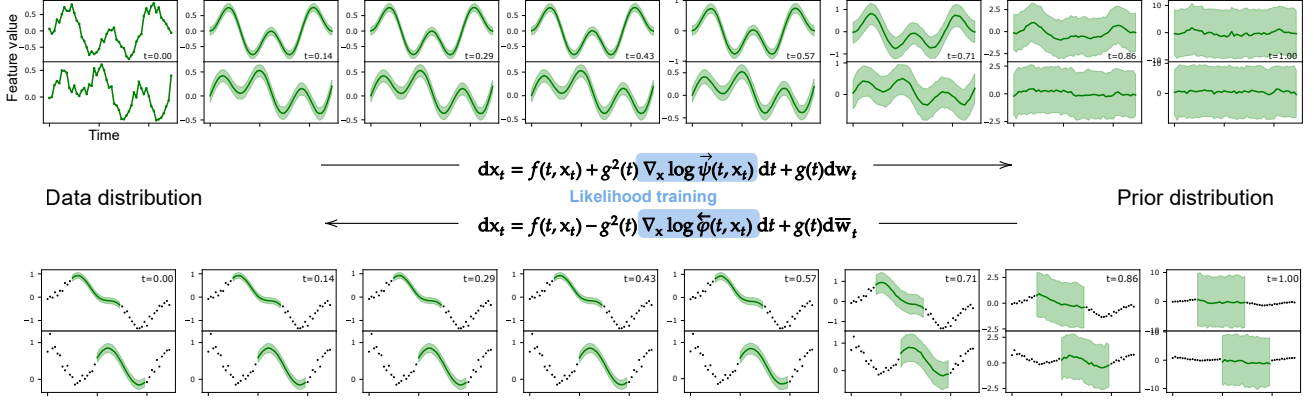
Figure 3. Demo of conditional Schrödinger bridge for imputation (CSBI). The example shows two correlated time-series features, the dark dots are condition values, and the green band represents an 80% confidence interval of the imputed missing values, which starts from the prior distribution with very large uncertainty on the right to narrow and smooth band on the left matching the data distribution very well. The score functions $\nabla \log \overrightarrow{\psi}$ and $\nabla \log \overleftarrow{\varphi}$ in FB-SDEs (11) are trained via divergence-based likelihood.

constraints (16) into a Lagrangian (Chen et al., 2021a):

$$\mathcal{L}(\rho, \boldsymbol{u}, \phi) := \int_0^T \int_{\mathbb{R}^d} \frac{1}{2} \|\boldsymbol{u}(\mathbf{x}_t, t)\|_2^2 \rho(\mathbf{x}_t, t)$$
$$+ \phi(\mathbf{x}_t, t) \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho(\boldsymbol{f} + g\boldsymbol{u})) - \frac{g^2 \Delta \rho}{2} \right) d\mathbf{x}_t dt,$$

where $\phi(\mathbf{x}_t, t)$ denotes a Lagrangian multiplier (Chen et al., 2021a). Further, consider the log transformation based on score functions $(\nabla \log \overrightarrow{\psi}, \nabla \log \overleftarrow{\varphi})$

$$\overrightarrow{\psi}(\mathbf{x}_t, t) = \exp(\phi(\mathbf{x}_t, t)/g^2)$$
$$\overleftarrow{\varphi}(\mathbf{x}_t, t) = \rho^\star(\mathbf{x}_t, t)/\overrightarrow{\psi}(\mathbf{x}_t, t),$$

where $\rho^\star(\mathbf{x}, t)$ is the optimal density conditional on the optimal control $\boldsymbol{u}^\star := \nabla \phi(\mathbf{x}, t)$. Now we obtain the FB-SDEs (Chen et al., 2022):

Proposition 1. *The forward-backward SDE (FB-SDE) associated with the problem* (3) *with* $\epsilon = \frac{1}{2}$ *and conditional constraints follows*

$$d\mathbf{x}_t = \left[\boldsymbol{f}(\mathbf{x}_t, t) + g(t)^2 \nabla \log \overrightarrow{\psi}(\mathbf{x}_t, t) \right] dt + g(t) d\mathbf{w}_t,$$
$$(11a)$$
$$d\mathbf{x}_t = \left[\boldsymbol{f}(\mathbf{x}_t, t) - g(t)^2 \nabla \log \overleftarrow{\varphi}(\mathbf{x}_t, t) \right] dt + g(t) d\overline{\mathbf{w}}_t.$$
$$(11b)$$

where $\mathbf{x}_0 \sim \mu_\star$ *and* $\mathbf{x}_T \sim \nu_\star$.

Define $\overrightarrow{\mathbf{z}}_t = g(t) \nabla \log \overrightarrow{\psi}(\mathbf{x}_t, t)$ and $\overleftarrow{\mathbf{z}}_t = g(t) \nabla \log \overleftarrow{\varphi}(\mathbf{x}, t)$. Itô's lemma (see Theorem 3 (Chen et al., 2022)) leads to the diffusion of $\overrightarrow{\mathbf{z}}_t$ and $\overleftarrow{\mathbf{z}}_t$ in (11).

We use models $\overrightarrow{\mathbf{z}}_t^\theta$ and $\overleftarrow{\mathbf{z}}_t^\omega$ to learn the forward policy $\overrightarrow{\mathbf{z}}_t$ and backward policy $\overleftarrow{\mathbf{z}}_t$ and refer to the objective of data likelihood as $\mathcal{L}_{\theta,\omega}^{\text{SBP}}$. In the context of imputation problems

with conditional and target entries, maximizing the likelihood is equivalent to optimizing the backward policy $\overleftarrow{\mathbf{z}}_t^\omega$ and forward policy $\overrightarrow{\mathbf{z}}_t^\theta$ as follows (Chen et al., 2022):

$$\mathcal{L}_\omega^{\text{SBP}}(\mathbf{x}_0) = -\widehat{\mathbb{E}}_{\mathbf{x}_t \sim (11a)} \left[\frac{1}{2} \| \overleftarrow{\mathbf{z}}_t^\omega \circ \boldsymbol{M}_{\text{target}} \|_2^2 + \quad (12a) \right.$$
$$\left. g\nabla \cdot \left(\overleftarrow{\mathbf{z}}_t^\omega \circ \boldsymbol{M}_{\text{target}} \right) + (\overrightarrow{\mathbf{z}}_t \circ \boldsymbol{M}_{\text{target}})^\intercal (\overleftarrow{\mathbf{z}}_t^\omega \circ \boldsymbol{M}_{\text{target}}) \right]$$

$$\mathcal{L}_\theta^{\text{SBP}}(\mathbf{x}_T) = -\widehat{\mathbb{E}}_{\mathbf{x}_t \sim (11b)} \left[\frac{1}{2} \| \overrightarrow{\mathbf{z}}_t^\theta \|_2^2 + g\nabla \cdot \overrightarrow{\mathbf{z}}_t^\theta + \overleftarrow{\mathbf{z}}_t^\intercal \overrightarrow{\mathbf{z}}_t^\theta \right],$$
$$(12b)$$

where $\widehat{\mathbb{E}}$ denotes the empirical expectations of the sampled trajectories according to the FB-SDEs (11); $\boldsymbol{M}_{\text{target}}$ is the conditional mask to be clarified in section 5.2; $\nabla \cdot$ denotes the divergence (for clarity, ∇ is the gradient.). The masks and conditions are not required in Eq.(12b), because the backward SDEs start from the known prior. Since simulating the full sample trajectory is costly, we apply the caching-trajectory strategy (De Bortoli et al., 2021; Chen et al., 2022) to improve the efficiency. Now, we present the practical method in Algorithm 2 and refer to the conditional Schrödinger bridge for imputation as CSBI.

During the inference, conditional sampling follows the joint distribution learning by applying the backward policy (Song et al., 2021b). The Langevin corrector (Song et al., 2021b; Chen et al., 2022) can also be used to improve performance. See details in Appendix C.6.

Algorithm 2 Likelihood training of conditional Schrödinger bridge for imputation (CSBI)

1: **Input:** samplers of joint space $p_{\text{prior}}(\mathbf{x})$, $p_{\text{obs}}(\mathbf{x})$, and masks.
2: Set output of $\overrightarrow{\mathbf{z}}_t^\theta$ as zero and warmup train $\overleftarrow{\mathbf{z}}_t^\omega$ using score matching.
3: **repeat**
4:    Update cached forward trajectories following (11a) with certain frequency.
5:    Compute $\mathcal{L}_\omega^{\text{SBP}}$ (12a) with $\mathbf{x}_0 \sim p_{\text{obs}}(\mathbf{x})$ together with M_{cond}, M_{target}.
6:    Take gradient step $\nabla \mathcal{L}_\omega^{\text{SBP}}$ and update $\overleftarrow{\mathbf{z}}_t^\omega$.
7:    Update cached backward trajectories of (11b) with certain frequency.
8:    Compute $\mathcal{L}_\theta^{\text{SBP}}$ (12b), with $\mathbf{x}_T \sim p_{\text{prior}}(\mathbf{x})$, $M_{\text{cond}} = \mathbf{0}$, $M_{\text{target}} = \mathbf{1}$.
9:    Take gradient step $\nabla \mathcal{L}_\theta^{\text{SBP}}$ and update $\overrightarrow{\mathbf{z}}_t^\theta$.
10: **until a stopping criterion**

5.2. Joint Space of SBP for Time Series Imputation

Time series imputation task requires filling missing values in arbitrary entries given partial observations in random positions, where the condition-target relation usually varies from sample to sample. This requires the model to capture both temporal and feature-wise dependency at the same time. Next, we present our framework based on divergence objectives. The joint distribution learning of $\mathbf{x} := (\mathbf{x}_{\text{target}}, \mathbf{x}_{\text{cond}})$ is the following.

$$\begin{cases} \mathbf{x}_0 \sim \mu_\star = p_{\text{target|cond}}(\mathbf{x}) p_{\text{cond}}(\mathbf{x}) \\ \mathbf{x}_T \sim \nu_\star = p_{\text{prior-target}}(\mathbf{x}) p_{\text{cond}}(\mathbf{x}), \end{cases} \quad (13)$$

where $p_{\text{target|cond}}(\mathbf{x}) = p(\mathbf{x}_{\text{target}}|\mathbf{x}_{\text{cond}})$ is the target conditional distribution, $p_{\text{prior-target}}(\mathbf{x})$ is the prior distribution of target values, and $p_{\text{cond}}(\mathbf{x}) = p(\mathbf{x}_{\text{cond}})$ is the data-dependent distribution of observations being conditioned on.
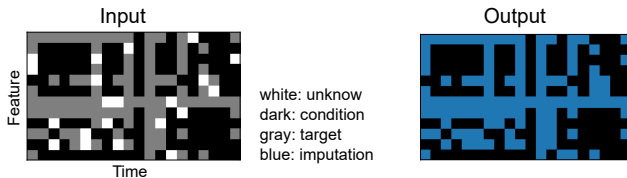

Figure 4. Demonstration of entry types and masks. The usage for SBP is described in section 5.2

Masking for conditional inference The irregular condition-target relation is indicated by observation mask M_{obs}, condition mask M_{cond}, and target mask M_{target} (Figure 4). M_{obs} covers all ground true values, unknown entries is complementary to M_{obs} without ground truths, M_{target} is for the imputation target, M_{cond} indicates the input condition for the model, which is a subset of M_{obs}. When the model is trained or evaluated, M_{target} is usually set as part of M_{obs} so the performance can be calculated by comparing the imputed values and the ground truths. When the model is deployed, M_{target} can also cover the unknown entries. See more details on masks in Appendix C.1.

6. Experiments

In this section, we evaluate the performance of CSBI through one synthetic data and two real datasets. [†]

6.1. Datasets

Synthetic Data We first test our algorithm on a simple synthetic dataset. The time series data has $K = 8$ features and $L = 50$ time steps. The data is created by adding signals, sinusoidal curves of various frequencies, and random noise. Next, data entries are removed randomly mimicking the missed observed values (unknown entries). The observed entries are split into conditions and artificial targets. 20 consecutive time points of each feature are selected as the artificial targets. More details can be found in Appendix C.2. Examples are shown in Figure 5.
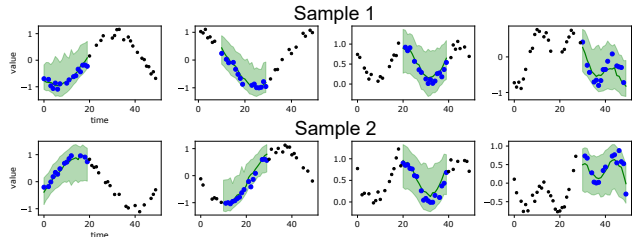

Figure 5. A demo with 4 features and 50 time steps. Dark dots are conditions, blue dots are the ground truth of the missing values, and the error bar shows the 80% confidence. The imputation is conducted for the targets (blue dots) and the unknowns (gaps between blue dots). The target mask of the last feature is at the end of the time window which is equivalent to the prediction task.

Environmental Data It is composed of the hourly sampled PM2.5 air quality index (with unit $\mu g/m^3$) from 36 monitoring stations for 12 months (Zheng et al., 2013). The time window has $K = 36$ features and $L = 36$ time points. The raw data has 13% missing values (the portion of un-

---

[†] https://github.com/morganstanley/MSML/tree/main/papers/Conditional_Schrodinger_Bridge_Imputation.

Table 1. Model evaluation: The metrics include root mean square error (RMSE), mean absolute error (MAE), and continuous ranked probability score (CRPS). A smaller metric is better. Results are obtained from 5 folds.

Metrics	PM2.5			PhysioNet 0.1			PhysioNet 0.5		
	RMSE	MAE	CRPS	RMSE	MAE	CRPS	RMSE	MAE	CRPS
V-RIN	40.1	25.4	0.53	0.63	0.27	0.81	0.69	0.37	0.83
Multitask GP	42.9	34.7	0.27	0.80	0.46	0.49	0.84	0.51	0.56
GP-VAE	43.1	26.4	0.41	0.73	0.42	0.58	0.76	0.47	0.66
CSDI	19.3	9.86	**0.11**	0.57	0.24	0.26	0.65	0.32	**0.35**
CSBI (ours)	**19.0**	**9.80**	**0.11**	**0.55**	**0.23**	**0.25**	**0.63**	**0.31**	**0.35**

known entries). The target entries only come from the observed entries. See demonstrations in Appendix C.10.

Healthcare Data Another dataset widely used in time series imputation literature is the PhysioNet Challenge 2012 (Silva et al., 2012). It has 4000 clinical time series with $K = 35$ features and $L = 48$ time points for 48 hours from the intensive care unit (ICU). The raw data is sparse with 80% missing values (the portion of unknown entries) making the imputation very challenging. We further randomly select 10% and 50% out of observed values as the targets. Demonstrations are shown in Appendix C.10.

6.2. Model Pipeline

In this section, we briefly describe the pipeline of the framework. More details about the neural networks, training procedure, inference, baseline models, and evaluation can be found in Appendix C.

As described in section 5.1 and algorithm 2, we use two separate neural networks to model the forward or backward policy $\overrightarrow{\mathbf{z}}_t$ and $\overleftarrow{\mathbf{z}}_t$. The backward network needs to handle partially observed input and conduct conditional inference. More specifically, the backward policy has format $\overleftarrow{\mathbf{z}}_t(t, \mathbf{X}_{t,\text{cond}}, \mathbf{M}_{\text{cond}})$ which takes in diffusion time, conditions, and outputs the policy of the whole time window (its outputs at condition positions are usually ignored). While the forward network, as an assistant for training the backward policy, does not need to process partial input, and we use a modified U-Net as the neural network (Ronneberger et al., 2015). In both networks, the diffusion time is incorporated through embedding. Similar to the design (Tashiro et al., 2021), the backward policy handles the input with irregular conditions based on the transformer, where the condition information is encoded through channel concatenation, feature index embedding, and time index embeddings (using the time point index of the time window, not the actual time of the time series to have a fair comparison with baseline models).

Our baseline models include V-RIN (Mulyadi et al., 2021), multitask Gaussian process (multitask GP) (Dürichen et al.,

2014), GP-VAE (Fortuin et al., 2020), and the state-of-the-art model CSDI (Tashiro et al., 2021). Our model is evaluated using 100 samples. We report the mean absolute error (MAE) and root mean square error (RMSE); in addition, we include the continuous ranked probability score (CRPS) to measure the quality of the imputed distribution that is calculated using all samples.

6.3. Evaluation

Synthetic Data We try to answer a key question here:

Does lower transport costs facilitate estimations of score functions and yield samples of higher quality?

To answer the question in a consistent framework, we compare CSBI with a CSBI variant by forcing $\nabla \log \overrightarrow{\psi} \equiv 0$ in Eq.(11), where the latter (denoted by CSBI_0) is, in theory, equivalent to the SGM-based CSDI. We observe in Figure 6 that all criteria of CSBI converge to small errors, however, CSBI_0 yields a rather crude performance when the terminal time T in Eq.(11) is small, which helps answer the question affirmatively. See details in Appendix C.8.
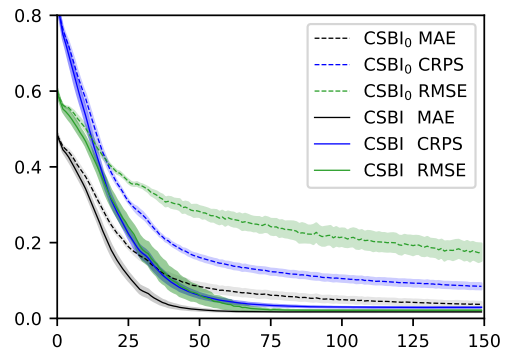

Figure 6. Comparison between CSBI_0 (equivalent to the SGM-based CSDI in theory) and CSBI in a consistent framework. Both models are trained with 150 iterations under the same settings.

Healthcare and Environmental Data We observe in Table 1 that the recurrent imputation networks in V-RIN

achieve the worst performance; the GP-based methods, such as multitask GP and GP-VAE, can model the uncertainty accurately, however, it is still inferior to CSDI in these two datasets; we suspect one reason is that GP-based methods rely more on distributional assumptions; in addition, the VAE structure needs to take the whole window (conditions and dummy missing values) as the input to generate the latent states, as such, the dummy values at the target entries may affect the performance. By contrast, our CSBI model achieves state-of-the-art performance and slightly outperforms the CSDI model, although the latter is already highly optimized. Despite the optimized transport cost, we didn't achieve a significant improvement. One reason is due to the large variance issue via the trace estimator (Hutchinson, 1989). See discussions in appendix C.7 for details.

6.4. Time Series Forecasting

Time series prediction is a natural application of our current framework, where the condition mask is substituted with the context window and the target mask with the future window. Our method accommodates missing values in the context window during both training and inference, eliminating the need to fill these gaps with dummy values. The training and inference remain the same as the imputation task. Table 2 shows the results using two public datasets: Solar and Exchange (Lai et al., 2018). Baseline models include GP-copula (Salinas et al., 2019), Vec-LSTM-low-rank-Copula (Vec) (Salinas et al., 2019), TransMAF (Rasul et al., 2021). Our method achieves competitive performance compared to other baseline models. For a comprehensive understanding, we provide detailed information in Appendix C.9.

Table 2. Time series prediction task evaluated by CRPS.

Methods	GP-copula	Vec	TransMAF	**CSBI (ours)**
Exchange	**0.008**	0.009	0.012	**0.008**
Solar	0.371	0.384	0.368	**0.365**

7. Conclusion

Schrödinger bridge algorithm is gaining popularity in generative models, however, to the best of our knowledge, there is no prior work studying the convergence based on the IPF algorithm with approximate projections. To bridge the gap between theoretical analysis and empirical training, we provide the first approximation analysis and motivate future research to obtain a tighter upper bound. We also draw connections to demystify the connections between IPF and FB-SDEs, which sheds light on the complexity analysis for future algorithm development.

For applications to probabilistic time series imputation, we propose a conditional Schrödinger bridge algorithm based on divergence-based likelihood training and the method is able to tackle missing values in random positions. Empiri-

cally, the proposed algorithm is tested on multiple datasets; the great performance indicates the effectiveness of our algorithm in time series imputation. The flexible formulation further shows a potential to extend the linear Gaussian prior to more general priors (Deng et al., 2020; 2022) to yield more efficient algorithms.

References

Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and Their Applications*, 12(3):313–326, 1982.

Bunne, C., Hsieh, Y.-P., Cuturi, M., and Krause, A. The Schrödinger Bridge between Gaussian Measures has a Closed Form. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Caluya, K. and Halder, A. Wasserstein Proximal Algorithms for the Schrödinger Bridge Problem: Density Control with Nonlinear Drift. *IEEE Transactions on Automatic Control*, 67(3):1163–1178, 2022.

Cao, W., Wang, D., Li, J., Zhou, H., Li, L., and Li, Y. BRITS: Bidirectional Recurrent Imputation for Time Series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific reports*, 8(1):1–12, 2018.

Chen, S., Chewi, S., Li, J., Li, Y., Salim, A., and Zhang, A. R. Sampling is as Easy as Learning the Score: Theory for Diffusion Models with Minimal Data Assumptions. In *Proc. of the International Conference on Learning Representation (ICLR)*, 2023.

Chen, T., Liu, G.-H., and Theodorou, E. A. Likelihood Training of Schrödinger Bridge using Forward-Backward SDEs Theory. In *Proc. of the International Conference on Learning Representation (ICLR)*, 2022.

Chen, Y. and Georgiou, T. Stochastic Bridges of Linear Systems. *IEEE Transactions on Automatic Control*, 61 (2), 2016.

Chen, Y., Georgiou, T. T., and Pavon, M. Stochastic Control Liaisons: Richard Sinkhorn Meets Gaspard Monge on a Schrödinger Bridge. *SIAM Review*, 63(2):249–313, 2021a.

Chen, Y., Georgiou, T. T., and Pavon, M. Optimal Transport in Systems and Control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:89–113, 2021b.

Christoffersen, P., Goyenko, R., Jacobs, K., and Karoui, M. Illiquidity Premia in the Equity Options Market. *The Review of Financial Studies*, 2017.

de Bézenac, E., Rangapuram, S. S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., Hasson, H., Gallinari, P., and Januschowski, T. Normalizing Kalman Filters for Multivariate Time series Analysis. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.

De Bortoli, V. Generative Modeling via Schrödinger bridge (basics on Schrödinger bridge). *Slides*, 2022.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Deng, R., , Brubaker, M. A., Mori, G., and Lehrmann, A. M. Continuous Latent Process Flows. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Deng, W., Lin, G., and Liang, F. A Contour Stochastic Gradient Langevin Dynamics Algorithm for Simulations of Multi-modal Distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Deng, W., Lin, G., and Liang, F. An Adaptively Weighted Stochastic Gradient MCMC Algorithm for Monte Carlo Simulation and Global Optimization. *Statistics and Computing*, pp. 32–58, 2022.

Dürichen, R., Pimentel, M. A., Clifton, L., Schweikard, A., and Clifton, D. A. Multitask Gaussian Processes for Multivariate Physiological Time-Series Analysis. *IEEE Transactions on Biomedical Engineering*, 62(1):314–322, 2014.

Fortuin, V., Baranchuk, D., Rätsch, G., and Mandt, S. GP-VAE: Deep Probabilistic Time Series Imputation. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Ghosal, P. and Nutz, M. On the Convergence Rate of Sinkhorn's Algorithm. *arXiv:2212.06000*, 2022.

Ghosal, P., Nutz, M., and Bernton, E. Stability of Entropic Optimal Transport and Schrödinger Bridges. *Journal of Functional Analysis*, 283, 2022.

Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models. In *Proc. of the International Conference on Learning Representation (ICLR)*, 2019.

Greco, G., Noble, M., Conforti, G., and Durmus, A. Non-asymptotic Convergence Bounds for Sinkhorn Iterates and Their Gradients: a Coupling Approach. *arXiv:2304.06549*, 2023.

Ho, J., Jain, A., and Abbeel, P. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Hutchinson, M. F. A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

Khrulkov, V. and Oseledets, I. Understanding DDPM Latent Codes through Optimal Transport. In *Proc. of the International Conference on Learning Representation (ICLR)*, 2023.

Kidger, P., Morrill, J., Foster, J., and Lyons, T. Neural Controlled Differential Equations for Irregular Time Series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Koehler, F., Heckett, A., and Risteski, A. Statistical Efficiency of Score Matching: The View from Isoperimetry. In *Proc. of the International Conference on Learning Representation (ICLR)*, 2023.

Kullback, S. Probability Densities with Given Marginals. *Ann. Math. Statist.*, 1968.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling Long-and Short-term Temporal Patterns with Deep Neural Networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Lavenant, H. and Santambrogio, F. The Flow Map of the Fokker–Planck Equation Does Not Provide Optimal Transport. *Applied Mathematics Letters*, 133, 2022.

Lee, H., Lu, J., and Tan, Y. Convergence for Score-based Generative Modeling with Polynomial Complexity. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Léonard, C. A Survey of the Schrödinger Problem and Some of its Connections with Optimal Transport. *Discrete & Continuous Dynamical Systems-A*, 34(4):1533–1574, 2014a.

Léonard, C. Some Properties of Path Measures. *Séminaire de Probabilités XLVI*, pp. 207–230, 2014b.

Li, X., Wong, T.-K. L., Chen, R. T. Q., and Duvenaud, D. Scalable Gradients for Stochastic Differential Equations. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Luo, Y., Zhang, Y., Cai, X., and Yuan, X. E2GAN: End-to-end Generative Adversarial Network for Multivariate time Series Imputation. In *Proceedings of the 28th international joint conference on artificial intelligence*, pp. 3094–3100. AAAI Press, 2019.

Ma, J. and Yong, J. *Forward-Backward Stochastic Differential Equations and their Applications*. Springer, 2007.

Malladi, R. and Sethian, J. A. Image Processing via Level Set Curvature Flow. *Proc. Natl. Acad. Sci.*, 92:7046–7050, 1995.

Mulyadi, A. W., Jun, E., and Suk, H.-I. Uncertainty-aware Variational-recurrent Imputation Network for Clinical Time Series. *IEEE Transactions on Cybernetics*, 2021.

Nutz, M. Introduction to Entropic Optimal Transport. *Lecture Notes*, 2022.

Nutz, M. and Wiesel, J. Stability of Schrödinger Potentials and Convergence of Sinkhorn's Algorithm. *Annals of Probability*, 2022.

Pavon, M., Tabak, E. G., and Trigila, G. The Data-driven Schrödinger Bridge. *Communications on Pure and Applied Mathematics*, 74:1545–1573, 2021.

Peyré, G. and Cuturi, M. *Computational Optimal Transport: With Applications to Data Science*. Foundations and Trends in Machine Learning, 2019.

Raginsky, M., Rakhlin, A., and Telgarsky, M. Non-convex Learning via Stochastic Gradient Langevin Dynamics: a Nonasymptotic Analysis. In *Proc. of Conference on Learning Theory (COLT)*, June 2017.

Rasul, K., Seward, C., Schuster, I., and Vollgraf, R. Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting. In *International Conference on Machine Learning*, pp. 8857–8868. PMLR, 2021.

Richter-Powell, J., Lipman, Y., and Chen, R. T. Q. Neural Conservation Laws: A Divergence-Free Perspective. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015.

Rubanova, Y., Chen, R. T. Q., and Duvenaud, D. Latent ODEs for Irregularly-Sampled Time Series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Ruschendorf, L. Convergence of the Iterative Proportional Fitting Procedure. *Annals of Statistics*, 1995.

Salinas, D., Bohlke-Schneider, M., Callot, L., Medico, R., and Gasthaus, J. High-dimensional Multivariate Forecasting with Low Rank Gaussian Copula Processes. *Advances in neural information processing systems*, 32, 2019.

Shi, Y., De Bortoli, V., Deligiannidis, G., and Doucet, A. Conditional Simulation Using Diffusion Schrödinger Bridges. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2022.

Shukla, S. N. and Marlin, B. M. Multi-time Attention Networks for Irregularly Sampled Time Series. In *Proc. of the International Conference on Learning Representation (ICLR)*, 2021.

Silva, I., Moody, G., Scott, D. J., Celi, L. A., and Mark, R. G. Predicting Inhospital Mortality of ICU patients: The Physionet/Computing in Cardiology Challenge 2012. In *Computing in Cardiology*, pp. 245–248. IEEE, 2012.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proc. of the International Conference on Machine Learning (ICML)*, 2015.

Song, Y. and Ermon, S. Generative Modeling by Estimating Gradients of The Data Distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum Likelihood Training of Score-Based Diffusion Models . In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations . In *Proc. of the International Conference on Learning Representation (ICLR)*, 2021b.

Tashiro, Y., Song, J., Song, Y., and Ermon, S. CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Vargas, F., Thodoroff, P., Lamacraft, A., and Lawrence, N. Solving Schrödinger Bridges via Maximum Likelihood. *Entropy*, 23(9):1134, 2021.

Wang, G., Jiao, Y., Xu, Q., Wang, Y., and Yang, C. Deep Generative Learning via Schrödinger Bridge. In *Proc. of the International Conference on Machine Learning (ICML)*, 2021.

Xiong, R. and Pelger, M. Large Dimensional Latent Factor Modeling with Missing Observations and Applications to Causal Inference. *Journal of Econometrics*, 233:271–301, 2023.

Zheng, Y., Liu, F., and Hsieh, H.-P. U-Air: When Urban Air Quality Inference Meets Big Data. In *Proceedings of the 19th SIGKDD conference on Knowledge Discovery and Data Mining (KDD'13)*, 2013.

Supplimentary Material for *"Provably Convergent Schrödinger Bridge with Applications to Probabilistic Time Series Imputation"*

In section A, we lay out the preliminary knowledge of Schrödinger Bridge; In section B, we establish the main convergence result for Schrödinger Bridge based on approximated scores; In section C, we provide the experimental details based on a synthetic dataset, PM2.5, and PhysioNet data.

A. Preliminaries

A.1. From Schrödinger Bridge problem (SBP) to FB-SDE

The stochastic-optimal-control perspective of SBP (see section 4.4 in Chen et al. (2021a) and (Pavon et al., 2021; Caluya & Halder, 2022)) proposes to minimize

$$
\inf_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}\left\{ \int_0^T \frac{1}{2}\|\boldsymbol{u}(\mathbf{x},t)\|_2^2 \mathrm{d}t \right\}
$$
$$
\text{s.t.} \ \ \mathrm{d}\mathbf{x}_t = [\boldsymbol{f}(\mathbf{x},t) + g(t)\boldsymbol{u}(\mathbf{x},t)]\,\mathrm{d}t + \sqrt{2\varepsilon}g(t)\mathrm{d}\mathbf{w}_t \tag{14}
$$
$$
\mathbf{x}_0 \sim \mu_\star, \ \ \mathbf{x}_T \sim \nu_\star,
$$

where $\mathcal{U}$ is a set of control variables $\boldsymbol{u} : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$; the state-space of $\mathbf{x}$ is $\mathbb{R}^d$ by default; $\boldsymbol{f} : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is the drift or vector field; $\mathbf{w}_t$ is the standard Brownian motion in $\mathbb{R}^d$. The expectation is taken w.r.t the joint state PDF $\rho(\mathbf{x},t)$ given initial and terminal conditions; ε is a scalar and is also related to the regularizer in the EOT formulation. Rewrite SBP into a variational formulation (Chen et al., 2021a), we have

$$
\inf_{\boldsymbol{u} \in \mathcal{U}, \rho} \int_0^T \int_{\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{u}(\mathbf{x},t)\|_2^2 \rho(\mathbf{x},t)\mathrm{d}\mathbf{x}\mathrm{d}t \tag{15}
$$
$$
\text{s.t.} \ \ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho(\boldsymbol{f} + g\boldsymbol{u})) = \varepsilon g^2 \Delta \rho \tag{16}
$$
$$
\rho(\mathbf{x},0) = \frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}}, \ \ \rho(\mathbf{y},T) = \frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}},
$$

Note that Eq.(16) is the Fokker-Planck equation for the corresponding controlled diffusion process (14) based on decision variables $(\rho, \boldsymbol{u}) \in \mathcal{P}(\mathbb{R}^d) \times \mathcal{U}$ and $\mathcal{P}(\mathbb{R}^d)$ is the set of probability measures on $\mathbb{R}^d$.

Consider the Lagrangian of (15) and introduce $\phi(\mathbf{x},t) : \mathbb{R}^d \times [0,T] \to \mathbb{R}$ as a Lagrangian multiplier (Chen et al., 2021b)

$$
\mathcal{L}(\rho, \boldsymbol{u}, \phi) := \int_0^T \int_{\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{u}(\mathbf{x},t)\|_2^2 \rho(\mathbf{x},t) + \phi(\mathbf{x},t)\left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho(\boldsymbol{f} + g\boldsymbol{u})) - \varepsilon g^2 \Delta \rho\right)\mathrm{d}\mathbf{x}\mathrm{d}t,
$$
$$
= \int_0^T \int_{\mathbb{R}^d} \left(\frac{1}{2}\|\boldsymbol{u}(\mathbf{x},t)\|_2^2 - \frac{\partial \phi}{\partial t} - \langle \nabla \phi, \boldsymbol{f} + g\boldsymbol{u}\rangle - \varepsilon g^2 \Delta \phi\right)\rho(\mathbf{x},t)\mathrm{d}\mathbf{x}\mathrm{d}t, \tag{17}
$$

where the second equation is obtained through integration by parts with respect to t and $\mathbf{x}$, respectively.

Minimizing with respect to $\boldsymbol{u}$, we get the optimal control $\boldsymbol{u}^\star$ as follows

$$
\boldsymbol{u}^\star(\mathbf{x},t) = g(t)\nabla \phi(\mathbf{x},t). \tag{18}
$$

Plugging Eq.(18) into Eq.(17), we have

$$
\mathcal{L}(\rho, \boldsymbol{u}, \phi) = \int_0^T \int_{\mathbb{R}^d} \left(-\frac{\partial \phi}{\partial t} - \frac{1}{2}\|g(t)\nabla \phi(\mathbf{x},t)\|_2^2 - \langle \nabla \phi, \boldsymbol{f}\rangle - \varepsilon g^2 \Delta \phi\right)\rho(\mathbf{x},t)\mathrm{d}\mathbf{x}\mathrm{d}t.
$$

To achieve the minimum control cost, we hope $\mathcal{L}(\rho, \boldsymbol{u}, \phi) = 0$, which means that

$$
\frac{\partial \phi}{\partial t} + \varepsilon g^2 \Delta \phi + \langle \nabla \phi, \boldsymbol{f}\rangle = -\frac{1}{2}\|g(t)\nabla \phi(\mathbf{x},t)\|_2^2.
$$

Given the optimal control $u^\star$, the above PDE is known as the *Hamilton–Jacobi–Bellman* (HJB) PDE. Since the HJB PDE is non-linear due to the presence of $\frac{1}{2}\|g(t)\nabla\phi(\mathbf{x},t)\|_2^2$, we make it linear through the Cole-Hopf transformation

$$\overrightarrow{\psi}(\mathbf{x},t) = \exp\left(\frac{\phi(\mathbf{x},t)}{2\varepsilon}\right) \tag{19a}$$

$$\overleftarrow{\varphi}(\mathbf{x},t) = \rho^\star(\mathbf{x},t)/\overrightarrow{\psi}(\mathbf{x},t), \tag{19b}$$

where $\rho^\star(\mathbf{x},t)$ is the optimal density of (15) conditional on the optimal control $u^\star$. We now can verify that the transformed variables $(\overrightarrow{\psi}, \overleftarrow{\varphi})$ solve the *Schrödinger system* and obtain the forward-backward Fokker-Planck (FPK) equations

$$\frac{\partial\overrightarrow{\psi}}{\partial t} + \langle\nabla\overrightarrow{\psi}, \boldsymbol{f}\rangle + \varepsilon g^2\Delta\overrightarrow{\psi} = 0,$$
$$\frac{\partial\overleftarrow{\varphi}}{\partial t} + \nabla\cdot(\overleftarrow{\varphi}\boldsymbol{f}) - \varepsilon g^2\Delta\overleftarrow{\varphi} = 0, \tag{20}$$

under the constraint that

$$\overrightarrow{\psi}(\mathbf{x},t=0)\overleftarrow{\varphi}(\mathbf{x},t=0) = \frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}}, \quad \overrightarrow{\psi}(\mathbf{y},t=T)\overleftarrow{\varphi}(\mathbf{y},t=T) = \frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}. \tag{21}$$

Next, applying the solution of Eq.(20) leads to the following equations by the Chapman-Kolmogorov equations

$$\overrightarrow{\psi}(\mathbf{x},t) = \int_{\mathbb{R}^d}\mathrm{Ker}_\varepsilon(t,\mathbf{x},T,\mathbf{y})\overrightarrow{\psi}(\mathbf{y},T)\mathrm{d}\mathbf{y}, \quad \overleftarrow{\varphi}(\mathbf{y},t) = \int_{\mathbb{R}^d}\mathrm{Ker}_\varepsilon(0,\mathbf{x},t,\mathbf{y})\overleftarrow{\varphi}(\mathbf{x},0)\mathrm{d}\mathbf{x}, \tag{22}$$

where $\mathrm{Ker}_\varepsilon(s,\mathbf{x},t,\mathbf{y})$ is the Markov kernel associated with the diffusion $\mathrm{d}\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t,t)\mathrm{d}t + \sqrt{2\varepsilon}g(t)\mathrm{d}\mathbf{w}_t$; closed-forms of Ker_ε is in general intractable; some concrete cases follow that

$$\mathrm{Ker}_\varepsilon(s,\mathbf{x}_s,t,\mathbf{x}_t) = \begin{cases} (2\pi\varepsilon(t-s))^{-n/2}\exp\left(-\frac{\left\|\mathbf{x}_t - e^{-\gamma(t-s)}\mathbf{x}_s\right\|_2^2}{2\varepsilon(1-e^{-2\gamma(t-s)})}\right) & \text{VP-SDE: } \boldsymbol{f}(\mathbf{x}_t,t) = -\gamma\mathbf{x}_t, g(t) \equiv 1, \\[4mm] (4\pi\varepsilon(t-s))^{-n/2}\exp\left(-\frac{\|\mathbf{x}_t-\mathbf{x}_s\|_2^2}{4\varepsilon(t-s)}\right) & \text{VE-SDE: } \boldsymbol{f}(\mathbf{x}_t,t) = 0, g(t) \equiv 1. \end{cases} \tag{23}$$

In view of Eq.(19), the optimal decision variables $(\rho, \boldsymbol{u})$ can be obtained as follows

$$\rho^\star(\mathbf{x},t) = \overrightarrow{\psi}(\mathbf{x},t)\overleftarrow{\varphi}(\mathbf{x},t), \qquad \boldsymbol{u}^\star(\mathbf{x},t) = 2\varepsilon g(t)\nabla\log\overrightarrow{\psi}(\mathbf{x},t). \tag{24}$$

Combining Eq.(21) and Eq.(22), we solve a variant of *Schrödinger equations* as follows

$$\overleftarrow{\varphi}(\mathbf{x})\int_{\mathbb{R}^d}e^{-c_\varepsilon(\mathbf{x},\mathbf{y})}\overrightarrow{\psi}(\mathbf{y})\mathrm{d}\mathbf{y} = \frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}}, \quad \overrightarrow{\psi}(\mathbf{y})\int_{\mathbb{R}^d}e^{-c_\varepsilon(\mathbf{x},\mathbf{y})}\overleftarrow{\varphi}(\mathbf{x})\mathrm{d}\mathbf{x} = \frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}, \tag{25}$$

where $c_\varepsilon(\cdot,\cdot)$ is the potential energy of the kernel $K_\varepsilon(0,\mathbf{x},T,\mathbf{y})$. $c_\varepsilon(\cdot,\cdot)$, $\overleftarrow{\varphi}(\cdot)$ and $\overrightarrow{\psi}(\cdot)$ are defined as follows

$$c_\varepsilon(\mathbf{x},\mathbf{y}) = -\log\mathrm{Ker}_\varepsilon(0,\mathbf{x},T,\mathbf{y}), \quad \overleftarrow{\varphi}(\cdot) \equiv \overleftarrow{\varphi}(\cdot,t=0) \text{ and } \overrightarrow{\psi}(\cdot) \equiv \overrightarrow{\psi}(\cdot,t=T). \tag{26}$$

Combining Eq.(18) and Eq.(19) and replacing $\boldsymbol{u}$ with $\overrightarrow{\psi}$ (Eq.(24)) into the forward diffusion process (14), we have

$$\mathrm{d}\mathbf{x}_t = \left[\boldsymbol{f}(\mathbf{x}_t,t) + 2\varepsilon g(t)^2\nabla\log\overrightarrow{\psi}(\mathbf{x}_t,t)\right]\mathrm{d}t + \sqrt{2\varepsilon}g(t)\mathrm{d}\mathbf{w}_t. \tag{27}$$

Following Anderson (1982); Chen et al. (2022) to reverse the forward diffusion (27), we obtain the backward diffusion:

$$\mathrm{d}\mathbf{x}_t = \left[\boldsymbol{f}(\mathbf{x}_t,t) + 2\varepsilon g(t)^2\nabla\log\overrightarrow{\psi}(\mathbf{x}_t,t) - 2\varepsilon g(t)^2\nabla\log\rho^\star(\mathbf{x}_t,t)\right]\mathrm{d}t + \sqrt{2\varepsilon}g(t)\mathrm{d}\mathbf{w}_t$$
$$= \left[\boldsymbol{f}(\mathbf{x}_t,t) - 2\varepsilon g(t)^2\nabla\log\overleftarrow{\varphi}(\mathbf{x}_t,t)\right]\mathrm{d}t + \sqrt{2\varepsilon}g(t)\mathrm{d}\bar{\mathbf{w}}_t,$$

where the second equality follows since $\log\overrightarrow{\psi}(\cdot,t) + \log\overleftarrow{\varphi}(\cdot,t) = \log\rho^\star(\cdot,t)$ by Eq.(19b).

Proposition 2. *Given the score functions* $(\overrightarrow{\psi}, \overleftarrow{\varphi})$ *that solve the* Schrödinger system

$$\begin{cases} \frac{\partial \overrightarrow{\psi}}{\partial t} + \langle \nabla \overrightarrow{\psi}, \boldsymbol{f} \rangle + \varepsilon g^2 \Delta \overrightarrow{\psi} = 0 \\ \frac{\partial \overleftarrow{\varphi}}{\partial t} + \nabla \cdot (\overleftarrow{\varphi} \boldsymbol{f}) - \varepsilon g^2 \Delta \overleftarrow{\varphi} = 0. \end{cases} \quad s.t. \ \overrightarrow{\psi}(\mathbf{x},0)\overleftarrow{\varphi}(\mathbf{x},0) = \frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}}, \quad \overrightarrow{\psi}(\mathbf{y},T)\overleftarrow{\varphi}(\mathbf{y},T) = \frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}.$$

Schrödinger system *yields the desired forward-backward stochastic differential equation (FB-SDE)*

$$\mathrm{d}\mathbf{x}_t = \left[\boldsymbol{f}(\mathbf{x}_t,t) + 2\varepsilon g(t)^2 \nabla \log \overrightarrow{\psi}(\mathbf{x}_t,t) \right] \mathrm{d}t + \sqrt{2\varepsilon}g(t)\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim \mu_\star \tag{28a}$$

$$\mathrm{d}\mathbf{x}_t = \left[\boldsymbol{f}(\mathbf{x}_t,t) - 2\varepsilon g(t)^2 \nabla \log \overleftarrow{\varphi}(\mathbf{x}_t,t) \right] \mathrm{d}t + \sqrt{2\varepsilon}g(t)\mathrm{d}\bar{\mathbf{w}}_t, \quad \mathbf{x}_T \sim \nu_\star. \tag{28b}$$

Setting $\varepsilon = \frac{1}{2}$ *recovers the FB-SDE* (11). *Part of the above derivation is standard and we present it here for the sake of self-containedness.*

A.2. An Important Property of Static Schrödinger Bridge Problem (SBP)

Lemma 1 (Structure Property of Static SBP (*Peyré & Cuturi, 2019; Nutz, 2022*))**.** *Let* $\mathcal{G} \backsim \mu_\star \otimes \nu_\star$, *where* $\backsim$ *is the cyclical invariant property* (*Ghosal et al., 2022*) *and* $\mathrm{d}\mathcal{G} \propto e^{-c_\varepsilon}\mathrm{d}(\mu_\star \otimes \nu_\star)$. *Suppose there is a unique coupling* $\pi_\star$ *for the static SBP*

$$\pi_\star = \underset{\pi \in \Pi(\mu_\star, \nu_\star)}{\arg\min} \ KL(\cdot|\mathcal{G}).$$

- *There exist measurable functions* $\varphi_\star$ *and* $\psi_\star$ *such that*

$$\frac{\mathrm{d}\pi_\star}{\mathrm{d}\mathcal{G}} = e^{\varphi_\star \oplus \psi_\star}, \tag{29}$$

  where $\varphi_\star, \psi_\star : \mathbb{R}^d \to [-\infty, \infty)$ *are known as the Schrödinger potentials. The* $\oplus$ *operator is defined as* $(\varphi_\star \oplus \psi_\star)(\mathbf{x},\mathbf{y}) = \varphi_\star(\mathbf{x}) + \psi_\star(\mathbf{y})$ *for functions* $\varphi_\star$ *and* $\psi_\star$. *The summation of potentials is unique.*

- *Suppose there is a solution* $\widehat{\pi}_0$ *that admits a density formulation*

$$\frac{\mathrm{d}\widehat{\pi}_0}{\mathrm{d}\mathcal{G}} = e^{\varphi_\star \oplus \psi_\star},$$

  for functions $\varphi_\star : \mathbb{R}^d \to [-\infty, \infty)$ *and* $\psi_\star : \mathbb{R}^d \to [-\infty, \infty)$, *it follows that* $\widehat{\pi}_0$ *is the Schrödinger bridge.*

A.3. Connections between Score Functions in FB-SDE and Potential Functions in EOT

To build connections between optimal transport and the SOC problem, we consider the following transformations

$$\overleftarrow{\varphi}(\mathbf{x}) \equiv \overleftarrow{\varphi}(\cdot, t=0) = e^{\varphi_\star(\mathbf{x})}\frac{\mathrm{d}\mu_\star}{\mathrm{d}\mathbf{x}} \tag{30a}$$

$$\overrightarrow{\psi}(\mathbf{y}) \equiv \overrightarrow{\psi}(\cdot, t=T) = e^{\psi_\star(\mathbf{y})}\frac{\mathrm{d}\nu_\star}{\mathrm{d}\mathbf{y}}, \tag{30b}$$

where $\nabla \log \overrightarrow{\psi}$ and $\nabla \log \overleftarrow{\varphi}$ are the forward and backward score functions and $\psi_\star$ and $\varphi_\star$ are the Schrödinger potentials.

We can obtain the *Schrödinger equations* by combining Eq.(30) and Eq.(25)

$$\int_{\mathbb{R}^d} e^{\varphi_\star(\mathbf{x})+\psi_\star(\mathbf{y})-c_\varepsilon(\mathbf{x},\mathbf{y})}\nu_\star(\mathrm{d}\mathbf{y}) = 1 \quad \mu_\star\text{-}a.s., \tag{31a}$$

$$\int_{\mathbb{R}^d} e^{\varphi_\star(\mathbf{x})+\psi_\star(\mathbf{y})-c_\varepsilon(\mathbf{x},\mathbf{y})}\mu_\star(\mathrm{d}\mathbf{x}) = 1 \quad \nu_\star\text{-}a.s. \tag{31b}$$

For any set $A \subset \mathbb{R}^d$, take the integral for the coupling Π on with respect to the marginals

$$\iint_{A \times \mathbb{R}^d} \pi_\star(\mathrm{d}\mathbf{x}, \mathrm{d}\mathbf{y}) = \iint_{A \times \mathbb{R}^d} \underbrace{e^{\varphi_\star(\mathbf{x}) + \psi_\star(\mathbf{y})} \mathcal{G}(\mathrm{d}\mathbf{x}, \mathrm{d}\mathbf{y})}_{\text{by Eq.(29)}} = \int_A \underbrace{\int_{\mathbb{R}^d} e^{\varphi_\star(\mathbf{x}) + \psi_\star(\mathbf{y}) - c_\varepsilon(\mathbf{x}, \mathbf{y})} \nu_\star(\mathrm{d}\mathbf{y})}_{:=1 \text{ by Eq.(31a)}} \mu_\star(\mathrm{d}\mathbf{x}) = \int_A \mu_\star(\mathrm{d}\mathbf{x})$$

$$\iint_{\mathbb{R}^d \times A} \pi_\star(\mathrm{d}\mathbf{x}, \mathrm{d}\mathbf{y}) = \iint_{\mathbb{R}^d \times A} \underbrace{e^{\varphi_\star(\mathbf{x}) + \psi_\star(\mathbf{y})} \mathcal{G}(\mathrm{d}\mathbf{x}, \mathrm{d}\mathbf{y})}_{\text{by Eq.(29)}} = \int_A \underbrace{\int_{\mathbb{R}^d} e^{\varphi_\star(\mathbf{x}) + \psi_\star(\mathbf{y}) - c_\varepsilon(\mathbf{x}, \mathbf{y})} \mu_\star(\mathrm{d}\mathbf{x})}_{:=1 \text{ by Eq.(31b)}} \nu_\star(\mathrm{d}\mathbf{y}) = \int_A \nu_\star(\mathrm{d}\mathbf{y}).$$

$$(32)$$

This implies that

$$\text{Eq.}(31a) \iff \text{the first marginal of } \pi_\star \text{ is } \mu_\star,$$
$$\text{Eq.}(31b) \iff \text{the second marginal of } \pi_\star \text{ is } \nu_\star.$$

In other words, if $(\varphi_\star, \psi_\star)$ are Schrödinger potentials, then $(\varphi_\star, \psi_\star)$ is a solution of Eq.(31a) and (31b).

In view of the connection between Eq.(22) and Eq.(30), we see the score functions $\overleftarrow{\varphi}(\cdot, t), \overrightarrow{\psi}(\cdot, t)$ in the FB-SDE (11) can be represented as the dynamic function of the potential functions $\psi_\star(\cdot), \varphi_\star(\cdot)$ in EOT

$$\overrightarrow{\psi}(\cdot, t) = \int_{\mathbb{R}^d} \mathrm{Ker}_\varepsilon(t, \cdot, T, \mathbf{y}) \overrightarrow{\psi}(\mathbf{y}, t = T) \mathrm{d}\mathbf{y} = \int_{\mathbb{R}^d} \mathrm{Ker}_\varepsilon(t, \cdot, T, \mathbf{y}) e^{\psi_\star(\mathbf{y})} \nu_\star(\mathrm{d}\mathbf{y})$$

$$\overleftarrow{\varphi}(\cdot, t) = \int_{\mathbb{R}^d} \mathrm{Ker}_\varepsilon(0, \mathbf{x}, t, \cdot) \overleftarrow{\varphi}(\mathbf{x}, t = 0) \mathrm{d}\mathbf{x} = \int_{\mathbb{R}^d} \mathrm{Ker}_\varepsilon(0, \mathbf{x}, t, \cdot) e^{\varphi_\star(\mathbf{x})} \mu_\star(\mathrm{d}\mathbf{x}),$$

where the kernel Ker_ε associated with the diffusion $\mathrm{d}\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t, t)\mathrm{d}t + \sqrt{2\varepsilon}g(t)\mathrm{d}\mathbf{w}_t$ follows from Eq.(23).

B. Convergence Analysis for the Marginals

Notations $\pi_k := (\mu_k, \nu_k)$ is the coupling at the k-th iteration with marginals μ_k and ν_k; $\pi_\star$ is the optimal coupling with target marginals $\mu_\star$ and $\nu_\star$; $\mu_{\star,k}$ and $\nu_{\star,k}$ denote the ϵ-approximation of $\mu_\star$ and $\nu_\star$ via approximations. φ_{k_1} and ψ_{k_2} denote the potential functions and the coupling can be represented as $\pi_{k_1+k_2} := \pi(\varphi_{k_1}, \psi_{k_2})$ by Lemma 1.

We are interested in the convergence of the marginals. However, computing the integrals in Algorithm 1 is too expensive. To handle this issue, we first present the exact formulation of the approximate IPF algorithm in Algorithm 3.

Algorithm 3 One iteration of approximate IPF (aIPF) based on approximate measures $\mu_{\star,k}$ and $\nu_{\star,k}$. In view of Eq.(10), it differs from Algorithm 1 in that $\mu_\star$ (or $\nu_\star$) is changed to $\mu_{\star,k}$ (or $\nu_{\star,k}$) for an exact equality.

$$\psi_k(\mathbf{y}) = -\log \int_{\mathbb{R}^d} e^{\varphi_k(\mathbf{x}) - c_\varepsilon(\mathbf{x}, \mathbf{y})} \mu_{\star,k}(\mathrm{d}\mathbf{x}), \qquad \varphi_{k+1}(\mathbf{x}) = -\log \int_{\mathbb{R}^d} e^{\psi_k(\mathbf{y}) - c_\varepsilon(\mathbf{x}, \mathbf{y})} \nu_{\star,k}(\mathrm{d}\mathbf{y}).$$

In such a case, by Lemma 1, the approximate couplings π_{2k} and π_{2k-1} follows that

$$\pi_{2k} := \pi(\varphi_k, \psi_k), \quad \pi_{2k-1} := \pi(\varphi_k, \psi_{k-1}), \quad \forall k \geq 0,$$

where the approximate potential functions φ_k and ψ_k (and ψ_{k-1}) are associated with the couplings $\pi_{2k} = (\mu_{2k}, \nu_{2k})$ and $\pi_{2k-1} = (\mu_{2k-1}, \nu_{2k-1})$ as follows

$$\mathrm{d}\pi_{2k} = e^{\varphi_k \oplus \psi_k - c_\varepsilon} \mathrm{d}(\mu_{\star,k} \otimes \nu_{\star,k}), \quad \mathrm{d}\pi_{2k-1} = e^{\varphi_k \oplus \psi_{k-1} - c_\varepsilon} \mathrm{d}(\mu_{\star,k} \otimes \nu_{\star,k-1}), \tag{34}$$

where $\varphi_{-1} = \psi_0 := 0$ and thus $\pi_{-1} = \mathcal{G}$.

Lemma 2. *For all $k \geq 0$ and $n \geq 0$, we have*

(i) $KL(\pi_{2k}|\pi_{2k-1})$ and $KL(\pi_{2k+1}|\pi_{2k})$ satisfy the following equations

$$KL(\pi_{2k}|\pi_{2k-1}) = \nu_{\star,k}(\psi_k + \log d\nu_{\star,k} - \psi_{k-1} - \log d\nu_{\star,k-1})$$
$$KL(\pi_{2k+1}|\pi_{2k}) = \mu_{\star,k+1}(\varphi_{k+1} + \log d\mu_{\star,k+1} - \varphi_k - \log d\mu_{\star,k}),$$

(ii) the summation of $KL(\pi_{2k}|\pi_{2k-1})$ and $KL(\pi_{2k+1}|\pi_{2k})$ follows that

$$\sum_{k=0}^{n-1} KL(\pi_{2k+1}|\pi_{2k}) \leq \mu_{\star}(\varphi_n) + O(n\epsilon), \quad \sum_{k=0}^{n} KL(\pi_{2k}|\pi_{2k-1}) \leq \nu_{\star}(\psi_n) + O(n\epsilon). \tag{35}$$

Proof (i) In view of Eq.(34), we have

$$\mathrm{KL}(\pi_{2k}|\pi_{2k-1}) = \iint_{\mathbb{R}^d \times \mathbb{R}^d} \log \frac{\mathrm{d}\pi_{2k}}{\mathrm{d}\pi_{2k-1}} \mathrm{d}\pi_{2k} = \iint_{\mathbb{R}^d \times \mathbb{R}^d} \big(\psi_k + \log d\nu_{\star,k} - \psi_{k-1} - \log d\nu_{\star,k-1}\big)\mathrm{d}\pi_{2k}$$
$$= \int_{\mathbb{R}^d} \big(\psi_k + \log d\nu_{\star,k} - \psi_{k-1} - \log d\nu_{\star,k-1}\big)\mathrm{d}\nu_{\star,k}. \tag{36}$$

Similarly, we can show $\mathrm{KL}(\pi_{2k+1}|\pi_{2k}) = \int_{\mathbb{R}^d} \big(\varphi_{k+1} + \log d\mu_{\star,k+1} - \varphi_k - \log d\mu_{\star,k}\big)\mathrm{d}\mu_{\star,k+1}$.

(ii) The non-negative property is clear; Summing up items in Eq.(36), we have

$$\sum_{k=0}^{n} \mathrm{KL}(\pi_{2k}|\pi_{2k-1}) = \sum_{k=0}^{n} \int_{\mathbb{R}^d} \left(\psi_k + \log d\nu_{\star,k} - \psi_{k-1} - \log d\nu_{\star,k-1}\right)\mathrm{d}\nu_{\star,k}$$
$$= \int_{\mathbb{R}^d} \left(\psi_n + \log d\nu_{\star,n} - \psi_{-1} - \log d\nu_{\star,-1}\right)\mathrm{d}\nu_{\star}+$$
$$+ \sum_{k=0}^{n} \int_{\mathbb{R}^d} \left(\psi_k + \log d\nu_{\star,k} - \psi_{k-1} - \log d\nu_{\star,k-1}\right)\mathrm{d}(\nu_{\star,k} - \nu_{\star})$$
$$\leq \nu_{\star}(\psi_n) + O(n\epsilon),$$

where the $O(\epsilon)$ approximation follows from Lemma 5 and 6 based on Assumption A1, Assumption A2, and Assumption A3. The other half can be shown similarly. □.

In the following, we present an important result for the convergence analysis.

Proposition 3.
$$\sum_{k=0}^{n} KL(\pi_k|\pi_{k-1}) \leq KL(\pi_{\star}|\mathcal{G}) - KL(\pi_{\star}|\pi_n) + O(n\epsilon). \tag{37}$$

Proof By Lemma 2 and Eq.(35), we can easily verify that

$$\sum_{k=0}^{2n} \mathrm{KL}(\pi_k|\pi_{k-1}) \leq \mu_{\star}(\varphi_n) + \nu_{\star}(\psi_n) + O(n\epsilon). \tag{38}$$

From another perspective, we know that

$$\mathrm{KL}(\pi_{\star}|\mathcal{G}) - \mathrm{KL}(\pi_{\star}|\pi_{2n}) = \mathbb{E}^{\pi_{\star}}[\log(\mathrm{d}\pi_{2n}/\mathrm{d}\mathcal{G})] = \mathbb{E}^{\pi_{\star}}[\varphi_n + \psi_n] + O(\epsilon) = \mu_{\star}(\varphi_n) + \nu_{\star}(\psi_n) + O(\epsilon). \tag{39}$$

Combining (39) and (38) concludes the proof for $2n$. Similarly, we can also show the proof for $2n - 1$. □

The above result shows $\mathrm{KL}(\pi_k|\pi_{k-1})$ decays (approximately) fast as $k \to \infty$, which implies a convergence of the marginals.

Lemma 3.

$$KL(\mu_k|\mu_{\star,\lfloor \frac{k+1}{2} \rfloor}) + KL(\nu_k|\nu_{\star,\lfloor \frac{k}{2} \rfloor}) \leq KL(\pi_k|\pi_{k-1}), \tag{40}$$

where $\lfloor \cdot \rfloor$ *is the floor function; the sum of RHS from* 1 *to* n *is upper bounded by a fixed constant*

$$\sum_{k \geq 1}^{n} KL(\pi_k|\pi_{k-1}) \leq KL(\pi_\star|\mathcal{G}) - KL(\pi_0|\mathcal{G}) + O(n\epsilon) \leq KL(\pi_\star|\mathcal{G}) + O(n\epsilon).$$

Proof   The first inequality holds by the data processing inequality in Lemma 7 for both even and odd k.

The second one follows by Eq.(37) and $\pi_{-1} = \mathcal{G}$.   □

The approximate IPF algorithm also yields other important theoretical properties.

Lemma 4.

$$\frac{d\mu_{2k}}{d\mu_{\star,k}} = e^{\varphi_k - \varphi_{k+1}}, \quad \frac{d\nu_{2k-1}}{d\nu_{\star,k-1}} = e^{\psi_{k-1} - \psi_k}, \quad \frac{d\mu_{2k+1}}{d\mu_{\star,k+1}} = 1, \quad \frac{d\nu_{2k}}{d\nu_{\star,k}} = 1.$$

Moreover, we have

$$KL(\mu_{2k}|\mu_{\star,k}) = \mu_{2k}(\varphi_k - \varphi_{k+1}) = KL(\pi_{2k}|\pi_{2k+1}) + O(\epsilon) \tag{41a}$$

$$KL(\nu_{2k-1}|\nu_{\star,k-1}) = \nu_{2k-1}(\varphi_{k-1} - \varphi_k) = KL(\pi_{2k-1}|\pi_{2k}) + O(\epsilon) \tag{41b}$$

$$KL(\mu_{\star,k+1}|\mu_{2k}) = \mu_{\star,k+1}(\varphi_{k+1} - \varphi_k) + O(\epsilon) = KL(\pi_{2k+1}|\pi_{2k}) + O(\epsilon) \tag{41c}$$

$$KL(\nu_{\star,k}|\nu_{2k-1}) = \nu_{\star,k}(\psi_k - \psi_{k-1}) + O(\epsilon) = KL(\pi_{2k}|\pi_{2k-1}) + O(\epsilon) \tag{41d}$$

$$KL(\mu_{2k+2}|\mu_{\star,k+1}) - KL(\mu_{2k+2}|\mu_{2k}) = \mu_{2k+2}(\varphi_k - \varphi_{k+1}) + O(\epsilon) \tag{41e}$$

$$KL(\nu_{2k+1}|\nu_{\star,k+1}) - KL(\nu_{2k+1}|\nu_{2k-1}) = \nu_{2k+1}(\psi_{k-1} - \psi_{k+1}) + O(\epsilon). \tag{41f}$$

Proof   By Eq.(34), we take the integral with respect to the second marginal to obtain the marginal density

$$\frac{d\mu_{2k}}{d\mu_{\star,k}} = \int_{\mathbb{R}^d} \frac{d\pi_{2k}}{d(\mu_{\star,k} \otimes \nu_{\star,k})}(\mathbf{x}, \mathbf{y})\nu_{\star,k}(d\mathbf{y}) = \int_{\mathbb{R}^d} e^{\varphi_k(\mathbf{x}) + \psi_k(\mathbf{y}) - c_\epsilon(\mathbf{x}, \mathbf{y})}\nu_{\star,k}(d\mathbf{y})$$
$$= e^{\varphi_k(\mathbf{x})} \int_{\mathbb{R}^d} e^{\psi_k(\mathbf{y}) - c_\epsilon(\mathbf{x}, \mathbf{y})}\nu_{\star,k}(d\mathbf{y}) = e^{\varphi_k(\mathbf{x})}e^{-\varphi_{k+1}(\mathbf{x})}, \tag{42}$$

where the last equality follows by Algorithm 3. $\frac{d\mu_{2k+1}}{d\mu_{\star,k+1}} = 1$ follows directly due to the definition in Eq.(10). Next, by Eq.(42) and Assumption A2, we show that

$$KL(\mu_{\star,k+1}|\mu_{2k}) = -\int_{\mathbb{R}^d} \log \frac{d\mu_{2k}}{d\mu_{\star,k}} \frac{d\mu_{\star,k}}{d\mu_{\star,k+1}} d\mu_{\star,k+1} = \mu_{\star,k+1}(\varphi_{k+1} - \varphi_k) + O(\epsilon),$$

where the last item follows by Assumption A2 and Lemma 6.

By Eq.(36) and Assumption A2, we can easily show the inequality in Eq.(41d). The rest can be proved similarly.   □

Before we finally present the final result, we provide some elementary entropy calculations.

Proposition 4. *For any* $k \geq 0$,

$$KL(\nu_{\star,k+1}|\nu_{2k+1}) \geq KL(\mu_{2k+2}|\mu_{\star,k}) + KL(\pi_{2k+2}|\pi_{2k}) - KL(\mu_{2k+2}|\mu_{2k}) + O(\epsilon) \geq KL(\mu_{2k+2}|\mu_{\star,k}) + O(\epsilon), \tag{43}$$

$$KL(\nu_{\star,k+1}|\nu_{2k+1}) \leq KL(\mu_{2k}|\mu_{\star,k}) - KL(\pi_{2k}|\pi_{2k+2}) + O(\epsilon) \leq KL(\mu_{2k}|\mu_{\star,k}) + O(\epsilon). \tag{44}$$

Moreover, an approximately monotone-decreasing property is shown as follows

$$KL(\mu_{2k}|\mu_{\star,k}) + O(\epsilon) \geq KL(\nu_{\star,k+1}|\nu_{2k+1})$$
$$KL(\nu_{\star,k+1}|\nu_{2k+1}) + O(\epsilon) \geq KL(\mu_{2k+2}|\mu_{\star,k+1})$$
$$KL(\mu_{2k+2}|\mu_{\star,k+1}) + O(\epsilon) \geq KL(\nu_{\star,k+2}|\nu_{2k+3})$$
$$\cdots$$

which further implies that $KL(\mu_{2k}|\mu_{\star,k})$ *and* $KL(\nu_{\star,k+1}|\nu_{2k+1})$ *are approximately monotone decreasing as* $k \to \infty$.

Proof

To prove Eq.(44), recall the definitions of π_{2k} and π_{2k+2} in Eq.(34), we first observe that

$$
\begin{aligned}
\mathrm{KL}(\pi_{2k}|\pi_{2k+2}) &= \iint_{\mathbb{R}^d \times \mathbb{R}^d} \log \frac{\pi_{2k}}{\pi_{2k+2}} \mathrm{d}\pi_{2k} \\
&= \iint_{\mathbb{R}^d \times \mathbb{R}^d} \log \frac{e^{\varphi_k + \varphi_k - c_\epsilon} \mathrm{d}\mu_{\star,k} \otimes \nu_{\star,k}}{e^{\varphi_{k+1} + \varphi_{k+1} - c_\epsilon} \mathrm{d}\mu_{\star,k+1} \otimes \nu_{\star,k+1}} \mathrm{d}\pi_{2k} \\
&= \iint_{\mathbb{R}^d \times \mathbb{R}^d} (\varphi_k - \varphi_{k+1} + \psi_k - \psi_{k+1}) \mathrm{d}\pi_{2k} + \iint_{\mathbb{R}^d \times \mathbb{R}^d} \log \frac{\mathrm{d}\mu_{\star,k} \otimes \nu_{\star,k}}{\mathrm{d}\mu_{\star,k+1} \otimes \nu_{\star,k+1}} \mathrm{d}\pi_{2k} \\
&\leq \mu_{2k}(\varphi_k - \varphi_{k+1}) + \nu_{\star,k}(\psi_k - \psi_{k+1}) + O(\epsilon) \\
&\leq \mu_{2k}(\varphi_k - \varphi_{k+1}) + \nu_{\star,k+1}(\psi_k - \psi_{k+1}) + O(\epsilon) \\
&\leq \mathrm{KL}(\mu_{2k}|\mu_{\star,k}) - \mathrm{KL}(\nu_{\star,k+1}|\nu_{2k+1}) + O(\epsilon),
\end{aligned}
\tag{45}
$$

where the first inequality follows by Lemma 6; the second inequality follows by Lemma 5; the last inequality follows by Eq.(41a) and Eq.(41d) in Lemma 4 and the approximate error $O(\epsilon)$ follows from Assumption A2.

To show Eq.(43), apply Eq.(41e) and Eq.(41d) in Lemma 4 again

$$
\begin{aligned}
\mathrm{KL}(\pi_{2k+2}|\pi_{2k}) &= \iint_{\mathbb{R}^d \times \mathbb{R}^d} \left(\varphi_{k+1} - \varphi_k + \psi_{k+1} - \psi_k + \log \frac{\mathrm{d}\mu_{\star,k+1} \otimes \nu_{\star,k+1}}{\mathrm{d}\mu_{\star,k} \otimes \nu_{\star,k}} \right) \mathrm{d}\pi_{2k+2} \\
&\leq \mu_{2k+2}(\varphi_{k+1} - \varphi_k) + \nu_{\star,k+1}(\psi_{k+1} - \psi_k) + O(\epsilon) \\
&\leq \mathrm{KL}(\mu_{2k+2}|\mu_{2k}) - \mathrm{KL}(\mu_{2k+2}|\mu_{\star,k+1}) + \mathrm{KL}(\nu_{\star,k+1}|\nu_{2k+1}) + O(\epsilon).
\end{aligned}
\tag{46}
$$

Moreover, by the data processing inequality (Nutz, 2022), we have $\mathrm{KL}(\pi_{2k+2}|\pi_{2k}) \geq \mathrm{KL}(\mu_{2k+2}|\mu_{2k})$. Combining Eq.(46) shows $\mathrm{KL}(\nu_{\star,k+1}|\nu_{2k+1}) + O(\epsilon) \geq \mathrm{KL}(\mu_{2k+2}|\mu_{\star,k+1})$. Eq.(45) naturally leads to $\mathrm{KL}(\mu_{2k}|\mu_{\star,k}) + O(\epsilon) \geq \mathrm{KL}(\nu_{\star,k+1}|\nu_{2k+1})$. The approximate decreasing property and the inequalities in Eq.(43) and Eq.(44) are proved. $\square$

B.1. Proof of Theorem 1

Finally, we are able to prove the main result, that is, the sublinear convergence for the marginals.

Proof   Recall in Lemma 3 that $\sum_{k\geq1}^{n} \mathrm{KL}(\pi_k|\pi_{k-1}) \leq \mathrm{KL}(\pi_\star|\mathcal{G}) - \mathrm{KL}(\pi_0|\mathcal{G}) + O(n\epsilon)$. By Eq.(40), we have $\mathrm{KL}(\mu_{2k}|\mu_{\star,k}) \leq \mathrm{KL}(\pi_{2k}|\pi_{2k-1})$; by Eq.(41d), $\mathrm{KL}(\nu_{\star,k}|\nu_{2k-1}) \leq \mathrm{KL}(\pi_{2k}|\pi_{2k-1}) + O(\epsilon)$. It follows that

$$
\sum_{k\geq1}^{n} \mathrm{KL}(\mu_{2k}|\mu_{\star,k}) \leq \mathrm{KL}(\pi_\star|\mathcal{G}) - \mathrm{KL}(\pi_0|\mathcal{G}) + O(n\epsilon), \quad \sum_{k\geq1}^{n} \mathrm{KL}(\nu_{\star,k}|\nu_{2k-1}) \leq \mathrm{KL}(\pi_\star|\mathcal{G}) - \mathrm{KL}(\pi_0|\mathcal{G}) + O(n\epsilon).
$$

Combining the approximately monotone decreasing property in Proposition 4 and Lemma 9, we have

$$
\mathrm{KL}(\mu_{2k}|\mu_{\star,k}) \leq \frac{\mathrm{KL}(\pi_\star|\mathcal{G}) - \mathrm{KL}(\pi_0|\mathcal{G})}{k} + O(\epsilon^{\frac{1}{2}} + k^{\frac{1}{2}}\epsilon), \quad \mathrm{KL}(\nu_{\star,k}|\nu_{2k-1}) \leq \frac{\mathrm{KL}(\pi_\star|\mathcal{G}) - \mathrm{KL}(\pi_0|\mathcal{G})}{k} + O(\epsilon^{\frac{1}{2}} + k^{\frac{1}{2}}\epsilon).
$$

Similar results hold for $\mathrm{KL}(\mu_{\star,k}|\mu_{2k})$ and $\mathrm{KL}(\nu_{2k-1}|\nu_{\star,k-1})$ by Eq.(41). Further combining Lemma 5 and Lemma 6, we can complete the first half of the proof as follows

$$
\mathrm{KL}(\mu_{2k}|\mu_\star) = \int \log \frac{\mathrm{d}\mu_{2k}}{\mathrm{d}\mu_\star} \mathrm{d}\mu_{2k} \leq \int \log \frac{\mathrm{d}\mu_{2k}}{\mathrm{d}\mu_{\star,k}} + \log \frac{\mathrm{d}\mu_{\star,k}}{\mathrm{d}\mu_\star} \mathrm{d}\mu_{2k} \leq \mathrm{KL}(\mu_{2k}|\mu_{\star,k}) + O(\epsilon).
$$

The rest can be proved similarly. $\square$

B.2. Auxiliary Results

Lemma 5. *Assume we have a probability density $\rho(\mathbf{x}) = C_{Norm} e^{-U(\mathbf{x})}$ defined on $\mathbb{R}^d$ and an approximate density $\widetilde{\rho}(\mathbf{x}) = C_{Norm} e^{-\widetilde{U}(\mathbf{x})}$, where the energy functions U and $\widetilde{U}$ are differentiable and satisfy*

$$
\|\nabla\widetilde{U} - \nabla U\|_\infty \leq \epsilon.
\tag{47}
$$

Moreover, the density ρ satisfies the dissipative assumption A1, then for an Lipschitz smooth function f, we have that

$$\left| \int_{\mathbb{R}^d} f(\mathbf{x})\widetilde{\rho}(\mathbf{x})d\mathbf{x} - \int_{\mathbb{R}^d} f(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} \right| \leq O(\epsilon),$$

where the big-O notation mainly depends on m_{ds} in assumption A1, the smoothness assumption A3, and the dimension d.

Proof Recall that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $U(\mathbf{x}) - U(\mathbf{y}) = \int_0^1 \frac{d}{dt} U(t\mathbf{x} + (1-t)\mathbf{y}) = \int_0^1 \langle \mathbf{x} - \mathbf{y}, \nabla U(t\mathbf{x} + (1-t)\mathbf{y})\rangle dt$. Since ρ and $\widetilde{\rho}$ are probability densities, there is a $\mathbf{x}_0$ such that $U(\mathbf{x}_0) = \widetilde{U}(\mathbf{x}_0)$. By the differentiability of U and $\widetilde{U}$, we have

$$
\begin{aligned}
|\widetilde{U}(\mathbf{x}) - U(\mathbf{x})| &= \left| \int_0^1 \langle \mathbf{x} - \mathbf{x}_0, \nabla \widetilde{U}(\cdot) - \nabla U(\cdot) dt \right| \\
&\leq \int_0^1 \|\mathbf{x} - \mathbf{x}_0\|_2 \cdot \left\| \nabla \widetilde{U}(\cdot) - \nabla U(\cdot) \right\|_2 dt \\
&\leq \epsilon(\|\mathbf{x}\|_2 + \|\mathbf{x}_0\|_2),
\end{aligned}
\tag{48}
$$

where the first inequality follows by Cauchy Schwarz inequality; the second inequality follows by Eq.(47); we use $\nabla U(\cdot)$ and $\nabla \widetilde{U}(\cdot)$ for convenience because the value holds for any t. As such, we have

$$
\begin{aligned}
\left| \int_{\mathbb{R}^d} f(\mathbf{x})\widetilde{\rho}(\mathbf{x})d\mathbf{x} - \int_{\mathbb{R}^d} f(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} \right| &= C_{\text{Norm}} \left| \int_{\mathbb{R}^d} e^{-U(\mathbf{x})} f(\mathbf{x}) \left(e^{U(\mathbf{x}) - \widetilde{U}(\mathbf{x})} - 1\right) d\mathbf{x} \right| \\
&\leq C_{\text{Norm}} \int_{\mathbb{R}^d} e^{-U(\mathbf{x})} |f(\mathbf{x})| \cdot \left| e^{U(\mathbf{x}) - \widetilde{U}(\mathbf{x})} - 1 \right| d\mathbf{x} \\
&\leq C_{\text{Norm}} \int_{\mathbb{R}^d} e^{-U(\mathbf{x})} |f(\mathbf{x})| \cdot \left| e^{\epsilon(\|\mathbf{x}\|_2 + \|\mathbf{x}_0\|_2)} - 1 \right| d\mathbf{x}.
\end{aligned}
$$

Given the dissipativity assumption A1, following Lemma 3.1 in (Raginsky et al., 2017) we have that

$$U(\mathbf{x}) \geq \frac{m_{ds}}{3} \|\mathbf{x}\|_2^2 - \frac{b_{ds}}{2} \log 3. \tag{49}$$

Consider a large enough compact set $\mathbb{C}$ that contains $\mathbf{x}_0$, we have that

$$
\begin{aligned}
&\left| \int_{\mathbb{R}^d} f(\mathbf{x})\widetilde{\rho}(\mathbf{x})d\mathbf{x} - \int_{\mathbb{R}^d} f(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} \right| \\
&\leq C_{\text{Norm}} \int_{\mathbb{R}^d} e^{-U(\mathbf{x})} |f(\mathbf{x})| \cdot \left| e^{\epsilon(\|\mathbf{x}\|_2 + \|\mathbf{x}_0\|_2)} - 1 \right| d\mathbf{x} \\
&= C_{\text{Norm}} \left(\int_{\mathbb{C}} e^{-U(\mathbf{x})} |f(\mathbf{x})| \cdot \left| e^{\epsilon(\|\mathbf{x}\|_2 + \|\mathbf{x}_0\|_2)} - 1 \right| d\mathbf{x} + \int_{\mathbb{R}^d/\mathbb{C}} e^{-U(\mathbf{x})} |f(\mathbf{x})| \cdot \left| e^{\epsilon(\|\mathbf{x}\|_2 + \|\mathbf{x}_0\|_2)} - 1 \right| d\mathbf{x} \right) \\
&\leq C_{\text{Norm}} \Big(\underbrace{\int_{\mathbb{C}} e^{-U(\mathbf{x})} |f(\mathbf{x})| \cdot \left| e^{\epsilon(\|\mathbf{x}\|_2 + \|\mathbf{x}_0\|_2)} - 1 \right| d\mathbf{x}}_{\text{I}} + \underbrace{\int_{\mathbb{R}^d/\mathbb{C}} |f(\mathbf{x})| \cdot e^{-\frac{m_{ds}}{3}\|\mathbf{x}\|_2^2 + 2\epsilon\|\mathbf{x}\|_2 + \frac{b_{ds}}{2}\log 3} d\mathbf{x}}_{\text{II}} \Big),
\end{aligned}
$$

where the last inequality follows by Eq.(49).

Recall that the quadratic growth of a Lipschitz continuous function f (Assumption A3) is much slower than the decay speed of an exponential function. As such, we can first upper bound II by the tail of a Gaussian density:

$$\text{II} = 3e^{\frac{b_{bs}}{2}} \int_{\mathbb{R}^d/\mathbb{C}} |f(\mathbf{x})| \cdot e^{-\frac{m_{ds}}{3}\|\mathbf{x}\|_2^2 + 2\epsilon\|\mathbf{x}\|_2} d\mathbf{x} \leq 3e^{\frac{3d\epsilon^2}{m_{ds}} + \frac{b_{bs}}{2}} \int_{\mathbb{R}^d/\mathbb{C}} |f(\mathbf{x})| e^{2\epsilon\|\mathbf{x}\|_2 - 2\epsilon\mathbf{x}} \cdot e^{-\frac{m_{ds}}{3}\left\|\mathbf{x} - \frac{3\epsilon}{m_{ds}}\mathbf{1}\right\|_2^2} d\mathbf{x} \leq O(\epsilon),$$

where the last inequality holds given a large enough compact set $\mathbb{C}$.

For the first term I with small enough ϵ and a fixed $\mathbb{C}$, applying Taylor expansion completes the proof.

$$\text{I} = \int_{\mathbb{C}} e^{-U(\mathbf{x})} |f(\mathbf{x})| \cdot \left| e^{\epsilon(\|\mathbf{x}\|_2 + \|\mathbf{x}_0\|_2)} - 1 \right| d\mathbf{x} \leq \sup_{\mathbf{x} \in \mathbb{C}} \epsilon(\|\mathbf{x}\|_2 + \|\mathbf{x}_0\|_2) \cdot |f(\mathbf{x})| \cdot \int_{\mathbb{C}} e^{-U(\mathbf{x})} d\mathbf{x} \leq O(\epsilon).$$

$\square$

Lemma 6. *Suppose we have probability densities $\rho(\mathbf{x})$ and $\widetilde{\rho}(\mathbf{x})$ that satisfy $\rho(\mathbf{x}) = e^{-U(\mathbf{x})}/C$ and $\widetilde{\rho}(\mathbf{x}) = e^{-\widetilde{U}(\mathbf{x})}/\widetilde{C}$ with C and $\widetilde{C}$ being the normalizing constants. Moreover, the energy functions U and $\widetilde{U}$ follow dissipative Assumption A1 and the smoothness Assumption A3 and $\|\nabla\widetilde{U} - \nabla U\|_\infty \le \epsilon$. Then we have that*

$$|\log\rho(\mathbf{x}) - \log\widetilde{\rho}(\mathbf{x})| \le O(\epsilon\|\mathbf{x}\|_2 + \epsilon).$$

Further, given a probability density $\widehat{\rho}(\mathbf{x})$ that satisfies the dissipative Assumption A1, we have

$$\int_{\mathbb{R}^d} |\log\rho(\mathbf{x}) - \log\widetilde{\rho}(\mathbf{x})|\widehat{\rho}(\mathbf{x})\mathrm{d}\mathbf{x} \le O(\epsilon).$$

Proof   (i) Similar to Eq.(48), we have $|\widetilde{U}(\mathbf{x}) - U(\mathbf{x})| \le \epsilon(\|\mathbf{x}\|_2 + \|\mathbf{x}_0\|_2)$. Combining $e^a \le 1 + 2a$ for $a \le 1$, it follows

$$|\widetilde{C} - C| \le \int_{\mathbb{R}^d} e^{-U(\mathbf{x})}\big|e^{-\widetilde{U}(\mathbf{x})+U(\mathbf{x})} - 1\big|\mathrm{d}\mathbf{x} \le \epsilon\underbrace{\int_{\mathbb{R}^d} e^{-U(\mathbf{x})}(2\|\mathbf{x}\|_2 + 2\|\mathbf{x}_0\|_2)\mathrm{d}\mathbf{x}}_{\text{integrable}},$$

where the last item is integrable due to the fast tail decay by Assumption A1. We can easily show that $|\frac{\widetilde{C}}{C} - 1| \le O(\epsilon)$ for small enough ϵ. It concludes that

$$|\log\rho(\mathbf{x}) - \log\widetilde{\rho}(\mathbf{x})| \le |\widetilde{U}(\mathbf{x}) - U(\mathbf{x})| + \big|\log C - \log\widetilde{C}\big| \le O(\epsilon\|\mathbf{x}\|_2) + O(\epsilon) = O(\epsilon\|\mathbf{x}\|_2 + \epsilon).$$

(ii) Similar to Lemma 5, the second result holds directly due to the fast tail decay induced by Assumption A1. $\square$

The following lemma is a restatement of Lemma 1.6 in (Nutz, 2022).

Lemma 7 (Data processing inequality)**.** *Let $P, Q \in \mathcal{P}(\Omega)$ and $K : \Omega \to \mathcal{P}(\Omega')$ a Markov kernel. Assume $P' \in \mathcal{P}(\Omega')$ and $Q' \in \mathcal{P}(\Omega')$ are the second marginals of $P \otimes K \in \mathcal{P}(\Omega \otimes \Omega')$ and $Q \otimes K \in \mathcal{P}(\Omega \otimes \Omega')$, respectively. Then we have*

$$KL(P'|Q') \le KL(P \otimes K|Q \otimes K).$$

Lemma 8. *Given a non-negative sequence $\{x_i\}_{0 \le i \le N}$ such that $\sum_{i=0}^{N} x_i \le C$ and $x_{i+1} \le x_i + \epsilon$, we have*

$$x_i \le \frac{C}{i+1} + \sqrt{2C\epsilon} + \epsilon.$$

Proof   Fix $0 \le i^* \le N$, consider the optimization problem

$$
\begin{aligned}
\max_{\boldsymbol{x}} \quad & x_{i^*} \\
\text{s.t.} \quad & \sum_{i=0}^{N} x_i \le C \text{ and } x_i \ge 0 \text{ for } 0 \le i \le N \\
& x_{i+1} \le x_i + \epsilon \text{ for } 0 \le i \le N-1 \,.
\end{aligned}
\tag{50}
$$

The optimal solution exists as this is a linear programming with a bounded feasible region. Let $x_{i^*}^*$ be the optimal value. Then, we must have

  (I) $x_i^* = x_{i-1}^* + \epsilon$ for any $i \le i^*$ where $x_{i-1}^* > 0$.

  (II) $x_i^* = 0$ for $i > i^*$.

To see (I), suppose $x_{i-1}^* > 0$ for some $i \le i^*$ and $x_i^* < x_{i-1}^* + \epsilon$. Then we can decrease x_{i-1}^* and increase each entry of $\{x_j^*\}_{i \le j \le i^*}$. Now the solution is still feasible but the objective value is increased, thus contracting the optimality of (50).

To see (II), if $x_i^* = a > 0$ for some $i > i^*$, we can set $x_i^* = 0$ and increase each element of $\{x_j\}_{0 \le j \le i^*}$ by $\frac{a}{i^*+1}$. Again, this would not violate any constraints.

Define $i_0^* \triangleq \min\{0 \le i \le N : x_i^* > 0\}$, the analysis can be broken down in two scenarios:

Scenario 1 When $i_0^* = 0$: by (I) and (II), we must have $x_0^* = c_0$ for some $c_0 > 0$, $x_i^* = c_0 + \epsilon i$ for all $i \leq i^*$ and $x_i^* = 0$ for all $i > i^*$. It follows that $\sum_{i=0}^N x_i^* = (i^* + 1)c_0 + \frac{i^*(1+i^*)\epsilon}{2} \leq C$, which implies $c_0 \leq \frac{C}{i^*+1} - \frac{i^*\epsilon}{2}$ and $i^* \leq \sqrt{\frac{2C}{\epsilon}}$. As a result, the optimal value of $x_{i^*}^*$ satisfies

$$x_{i^*}^* = (c_0 + i^*\epsilon) \leq \frac{C}{i^* + 1} + \frac{(i^*)\epsilon}{2} \leq \frac{C}{i^* + 1} + \sqrt{C\epsilon}.$$

Scenario 2 When $i_0^* > 0$: by (I) and (II), we have $x_i^* = 0$ for $i \notin [i_0^*, i^*]$ and $x_i^* = c_0 + \epsilon(i - i_0^*)$ for all $i_0^* \leq i \leq i^*$ and some $0 < c_0 \leq \epsilon$. Define $I \triangleq i^* - i_0^* + 1$, we have $\sum_{i=0}^N x_i^* = c_0 I + \frac{(I-1)I\epsilon}{2} \leq C$, which implies $(I - 1) \leq \sqrt{\frac{2C}{\epsilon}}$. The optimal value of $x_{i^*}^*$ satisfies

$$x_{i^*}^* = c_0 + (I - 1)\epsilon \leq \epsilon + \sqrt{2C\epsilon}.$$

Combining the results of Scenario 1 and Scenario 2 completes the proof. □

Lemma 9. *Given a non-negative sequence $\{x_i\}_{i \geq 0}$ such that $\sum_i x_i = C < \infty$ and $x_{i+1} \leq x_i + \epsilon$. We have*

$$x_i \leq \frac{C}{i + 1} + \sqrt{2C\epsilon} + \epsilon.$$

Proof 1) Since $\sum_i x_i = C < \infty$, we have N such that $\sum_{i>N} x_i < \epsilon$ which implies $x_i \leq \epsilon$ for $i > N$; 2) For $0 \leq i \leq N$, we have $\sum_{i=1}^N x_i \leq C$, applying Lemma 8 shows $x_i \leq \frac{C}{i+1} + \sqrt{2C\epsilon} + \epsilon$ for $0 \leq i \leq N$. □

C. Experimental Details

C.1. Conditional-inference framework

Consider an arbitrary window of a *multivariate* time series of some fixed length L and K features (variates): $\mathbf{x}_{\text{data}} \in \mathbb{R}^{K \times L}$ from the full training dataset. The entries of this window are labeled by *observation*, *condition*, *target*, and *unknown* (one entry can have multiple labels). Observations represent all known values from the raw data; in many cases, the raw data has missing values, so the complementary of *observation* is *unknown*; the condition entries are presented to the model as partial information of the window and are part of the observations.

To evaluate the model, the target entries are randomly selected from the observations, but these are hidden from the model as artificial "missing" values. The performance metrics are calculated by comparing the imputed values and the known observations as ground truth. The locations of observation, condition, and target missing values in a time series window are indicated by binary masks $M_{\text{obs}}, M_{\text{cond}}, M_{\text{target}} \in \{0, 1\}^{K \times L}$. Their values can thus be obtained through Hadamard product $\mathbf{x}_{\text{obs}} = \mathbf{x}_{\text{data}} \circ M_{\text{obs}}$, $\mathbf{x}_{\text{cond}} = \mathbf{x}_{\text{data}} \circ M_{\text{cond}}$ and $\mathbf{x}_{\text{target}} = \mathbf{x}_{\text{data}} \circ M_{\text{target}}$, respectively. The masks may change from window to window. Note that $M_{\text{cond}} \circ M_{\text{target}} \equiv \mathbf{0}$ and $M_{\text{cond}} \circ M_{\text{obs}} \equiv M_{\text{cond}}$. $M_{\text{cond}} + M_{\text{target}}$ is not necessarily equal to M_{obs} or $\mathbf{1}^{K \times L}$. The unknown entries do not have ground true values, as shown in Figure. 4. Having formulated the general multivariate time series imputation task as a probabilistic model, the imputation task is treated as a conditional generative model and the goal is to sample according to $p(\mathbf{x}_{\text{target}}|\mathbf{x}_{\text{cond}}, M_{\text{cond}}, M_{\text{target}})$.

C.2. Datasets

Synthetic dataset Each sample has $K = 8$ features and $L = 50$ time points. The signal has a simple temporal and feature structure. The signals are a mixture of sinusoidal curves.

$$\begin{aligned}
&\text{signal}_1(t) = \sin(2\pi t) && \text{signal}_5(t) = \sin^2(2\pi t)\cos(2\pi t) + 0.3t \\
&\text{signal}_2(t) = \cos(2\pi t) && \text{signal}_6(t) = \sin^3(2\pi t) - 0.3t \\
&\text{signal}_3(t) = \sin^2(2\pi t) && \text{signal}_7(t) = \cos^2(2\pi t)e^{-0.1t} - 0.2t \\
&\text{signal}_4(t) = 2\sin^2(2\pi t)\cos(2\pi t) && \text{signal}_8(t) = \cos^2(2\pi t)\sin(2\pi t)e^{0.4t} + 0.2t.
\end{aligned}$$

The noisy data is created by randomly shifting the phases and adding Gaussian noise,

$$\mathbf{x}_{\text{data},i}(k, t) = \text{signal}_k(t + \omega_i) + \sigma_{\text{noise}} \cdot \varepsilon_{i,k,t}, \quad k = 1, ..., K, \quad t = 1, ..., L$$

where i is the sample index, $\varepsilon_{i,k,t} \overset{i.i.d.}{\sim} N(0,1)$, $\omega_i \overset{i.i.d.}{\sim}$ Uniform$(0,1)$. The phase of each sample is random $\omega_i \overset{i.i.d.}{\sim}$ Uniform$[0,1]$, and all features in a sample share the same phase shift. Imputing the missing values requires inferring the phase of the signal based on a partially observed noisy signal which imposes the learning of the dependency between conditions and targets to handle the imputation task. Once raw data is created, some time points are randomly removed, mimicking the missed observed values (unknown entries). The observed entries are split into conditions and artificial missing values (targets). 20 consecutive time points of each feature are selected as artificial missing values (targets).

Real datasets The model is applied to real datasets such as air quality PM2.5 (Zheng et al., 2013) and PhysioNet (Silva et al., 2012). The air quality data has $K = 36$ features and $L = 36$ time points. The raw data has 13% missing values (the portion of the unknown entries). The PhysioNet data has 4000 clinical time series with $K = 35$ features and $L = 48$. The raw data is sparse with 80% missing values. We further randomly select 10% and 50% out of observed values as the targets. The preprocess and time window splitting follow the previous work (Tashiro et al., 2021). Both real datasets have large dimensions (in terms of $K \times L$) than the synthetic data.

C.3. SDEs

VESDE The forward SDE is $d\mathbf{x}_t = \sqrt{\frac{d\sigma^2(t)}{dt}} = g(t)d\mathbf{w}_t = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t \sqrt{2\log\frac{\sigma_{\max}}{\sigma_{\min}}}d\mathbf{w}_t$. The variance term is $\sigma(t) = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t$, $t \in (0,1]$. $\sigma_{\min}$ is usually set as a very small value close to zero. $\sigma_{\max}$ is set as a much larger value than the variance of the data so $p(\mathbf{x}_t|\mathbf{x}_0)$ is closer to normal distribution as t approximates $T = 1$.

VPSDE The forward SDE is $d\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t dt + \sqrt{\beta(t)}d\mathbf{w}_t$, where $\beta(t) = \beta_{\min} + t(\beta_{\max} - \beta_{\min})$, $t \in (0,1]$. A straightforward numerical scheme follows that

$$\mathbf{x}_{i+1} \approx \left(1 - \frac{1}{2}\beta^{\text{VPSDE}}(i/N)\Delta\right)\mathbf{x}_i + \sqrt{\beta^{\text{VPSDE}}(i/N)\Delta} \cdot \boldsymbol{\varepsilon}.$$

which is adopted in Song et al. (2021b).

C.4. Model details

The key structure is the backward policy for generating imputed values; The forward policy aims to reduce the transport cost.

Transformer for the backward policy The diagram below shows the major transformations of the neural network. The tuple (B, C, K, L) represents the shape of a tensor, where B is the batch size, C is the number of channels, K is the number of features, L is the number of time points. The backward policy takes $\mathbf{x}_{\text{cond}}, \boldsymbol{M}_{\text{cond}}$ as the input.

$$\mathbf{x}_{\text{cond}} \xrightarrow{\text{input}} (B, K, L) \xrightarrow{\text{unsqueeze}} (B, 1, K, L) \xrightarrow{\text{stem}} (B, C-1, K, L) \xrightarrow{\text{concatenate } \boldsymbol{M}_{\text{cond}}}$$
$$(B, C, K, L) \xrightarrow[\text{step 1}]{\text{add embedding}} (B, C, K, L) \xrightarrow[\times N_{\text{layer}}]{\text{transformer blocks}} (B, C, K, L) \xrightarrow{\text{output projection}} (B, K, L).$$

In step 1, the embedding is a concatenation of feature index, time index, and the condition mask with shape $(B, C_{\text{feature}} + C_{\text{time}} + 1, K, L)$. The time index is for the time series not for the SDE diffusion time. The feature embedding is the same for all batches and all time point, and the time embedding is the same for all batches and all features. Then the embedding is projected to C channels. The diffusion time embedding is added in the transformer blocks. The model stacks N_{layer} transformer blocks with residual connections. The diagram of the main component is the following,

$$\xrightarrow{\text{input}} (B, C, K, L) \xrightarrow[\text{step 1}]{\text{reshape}} (BK, L, C) \xrightarrow[\text{step 2}]{\text{time transformer}} (BK, L, C) \xrightarrow[\text{step 3}]{\text{reshape}}$$
$$(B, C, K, L) \xrightarrow[\text{step 4}]{\text{reshape}} (BL, K, C) \xrightarrow[\text{step 5}]{\text{feature transformer}} (BL, K, C) \xrightarrow{\text{reshape}} (B, C, K, L)$$

Each block has two transformers, one is for temporal information and the other is for feature information. In step 2, the time transformer encoder performs along the L dimension as the sequence and treats C as the embedding. The size of the attention matrix is $L \times L$. In step 1, the feature dimension is reshaped into batch dimension meaning all features share the same transformer function. Similarly, in step 5, the feature transformer performs along features K as the sequence and

uses C as the embedding. This is the reason why step 4 reshapes time dimension L into batch dimension. The size of the attention matrix is $K \times K$. Our model has $N_{\text{layer}} = 4$ transformers blocks. Each transformer block has 64 channels, 8 attention heads. Totally it has 414 thousand parameters.

U-Net for the forward policy The forward policy does not hand missing values. We use the U-Net for the forward policy (Ronneberger et al., 2015; Song & Ermon, 2019). It has skip connections from the down-scaling branch to the up-scaling branch on each scale. Our model has 3 down-scaling layers and 3 up-scaling layers with 32 channels and 664K parameters.

Activation functions It is important to note that the loss function of the model involves the calculation of the divergence with respect to the data. We use *SiLU* instead of *ReLU* to avoid vanishing gradients.

C.5. Training

Compared to the denoising score matching method, the Schrödinger bridge method involves optimizing both forward and backward SDEs and sampling non-linear forward SDE, which makes it harder to train, perform inference, debug, and tune the model. In both approaches, the inference only requires the backward SDE and the procedure is similar.

Hyperparameters The model is warmed up using SGM for about 6000 iterations with batch size 64. We use AdamW as the optimizer. The alternative training has 40 stages with each stage running 480 iterations. The trajectories are sampled every 80 iterations. The learning rates for the forward and backward steps are 2×10^{-6} and 2×10^{-5}, respectively. The exponential decay scheduler is adopted to improve stability. We use VESDE as the base SDE as introduced in section C.3. $\sigma_{\max} = 20, \sigma_{\min} = 0.001$. We use 100 discretization steps. The prior distribution for VESDE is $N(0, \sigma_{\max}^2 \mathbf{I})$.

C.6. Inference

The imputation task requires conditional sampling $p(\mathbf{x}_{\text{target}} | \mathbf{x}_{\text{cond}}, \boldsymbol{M}_{\text{cond}}, \boldsymbol{M}_{\text{target}})$. The conditional inference model needs to process partially observed information $\mathbf{x}_{\text{obs}}$ and the condition mask $\boldsymbol{M}_{\text{obs}}$.

Conditional inference The conditional inference, more specifically the imputation or inpainting in our case (Song et al., 2021b; Tashiro et al., 2021), is the following,

---

Algorithm 4 Conditional inference based on VESDE

1: **Input:** trained backward policy $\overleftarrow{\mathbf{z}}$, $\mathbf{x}_{\text{cond}}$, $\boldsymbol{M}_{\text{cond}}$, hyperparameters $\sigma_{\min}, \sigma_{\max}$.
2: Draw sample $\mathbf{x}_T \sim N(\mathbf{0}, \sigma_{\max}^2 \mathbf{I})$
3: **for** $i \leftarrow T$ to 1 **do**
4:     $t = i/T, \varepsilon \sim N(0, \mathbf{I}), \Delta = 1/T, g = \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}} \right)^t \sqrt{2 \log \frac{\sigma_{\max}}{\sigma_{\min}}}$
5:     $\mathbf{x}_i = \mathbf{x}_{\text{cond}} \circ \boldsymbol{M}_{\text{cond}} + \mathbf{x}_i \circ (\mathbf{1} - \boldsymbol{M}_{\text{cond}})$
6:     $\mathbf{x}_{i-1} = \mathbf{x}_i + [-\boldsymbol{f}(t) + g\overleftarrow{\mathbf{z}}(t, \mathbf{x}_i, \boldsymbol{M}_{\text{cond}})]\Delta + g\sqrt{\Delta}\varepsilon$
7: **end for**

C.7. Limitations

The marginal improvement doesn't mean minimizing the transport cost via the control variable $\boldsymbol{u}$ is not promising; by contrast, our model is limited by other complications such as the divergence approximations. How to reduce the variance and computation workload of the Hutchinson estimators (Hutchinson, 1989; Grathwohl et al., 2019) will be essential to improve the performance; other interesting updates can be seen in Richter-Powell et al. (2022).

C.8. Empirical verification

In this section, we empirically compare the convergence of CSBI and CSBI_0 using the synthetic data as described in Appendix C.2. To make a fair comparison, CSBI_0 is trained following Eq.(12) by *forcing* $\overrightarrow{\mathbf{z}}_t \equiv 0$, which is equivalent in theory to score matching loss (Chen et al., 2022). Two models share the same settings with a constant learning rate except that our method trains the forward policy in each iteration. Since CSBI_0 needs more forward diffusion steps to converge to the ideal prior distribution, it may have poor performance when the number of diffusion steps is insufficient or the variance

of the diffusion is small. As a comparison, SBP can overcome such an issue by minimizing the transport cost through the forward SDE (De Bortoli et al., 2021). In this experiment, the number of diffusion steps is 20, and the variance of the forward diffusion is small $\sigma_{\max} = 0.3$ using VESDE as described in Appendix C.3.

C.9. Time series prediction

Our model can be easily adopted for time series prediction task by simply manipulating the masks. The condition mask in our model corresponds to the context window in the prediction task, and the target mask is equivalent to the future window. Our method allows missing values in the context window during both training and inference. The training and inference procedures remain the same as the imputation task.

We evaluate the performance of the model using two public datasets: Solar and Exchange (Lai et al., 2018). Details of the datasets are shown in Table 3. Baseline models include GP-copula (Salinas et al., 2019), Vec-LSTM-low-rank-Copula (Vec) (Salinas et al., 2019), TransMAF (Rasul et al., 2021). The performance of the baseline models is from the reference therein.

Table 3. Properties of the datasets.

Datasets	Dimension	Frequency	Total time points	Context length	Prediction length
Exchange	8	Daily	6,071	48	30
Solar	137	Hourly	7,009	80	24

C.10. Imputation examples

In this section, we demonstrate an example of the diffusion process from the prior distribution to the final data distribution using the PM2.5 dataset as shown in Figure 7. Figure 8 and 12 present more examples of the imputed data distributions. Figure 8, 9, and 10 illustrate the irregularity of the time series imputation, where the missing values can location anywhere in the window. In Figure 8, the top left feature has much more missing values than the feature in row 3 column 1. As a comparison, Figure 9 provides a different layout of missing values and conditions. All these imputations are handled by one model not by models trained separately with different masks.
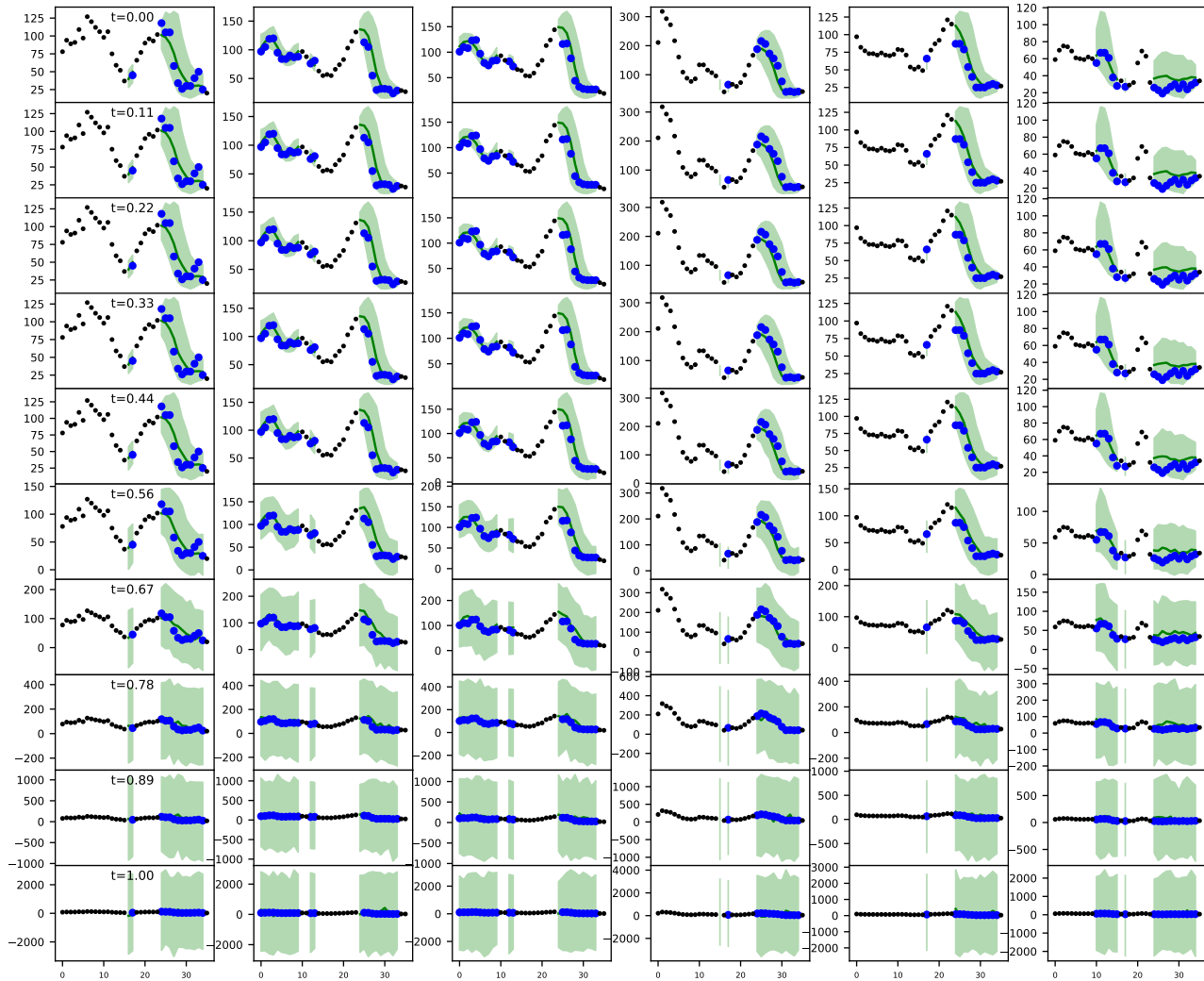
Figure 7. Demonstration of the backward diffusion process for conditional inference using the air quality PM2.5 dataset. The diffusion process starts from the prior distribution at the bottom ($t = 1$), and the backward diffusion will converge to the data distribution at the top ($t = 0$). Each column is one feature, each row is one diffusion time. The dark dots are $\mathbf{x}_{\text{obs}}$, the blue dots are true values of $\mathbf{x}_{\text{target}}$, the green band is 80% confidence interval of imputation. The imputation is performed using the normalized data; the figure shows the time series in the original scale.
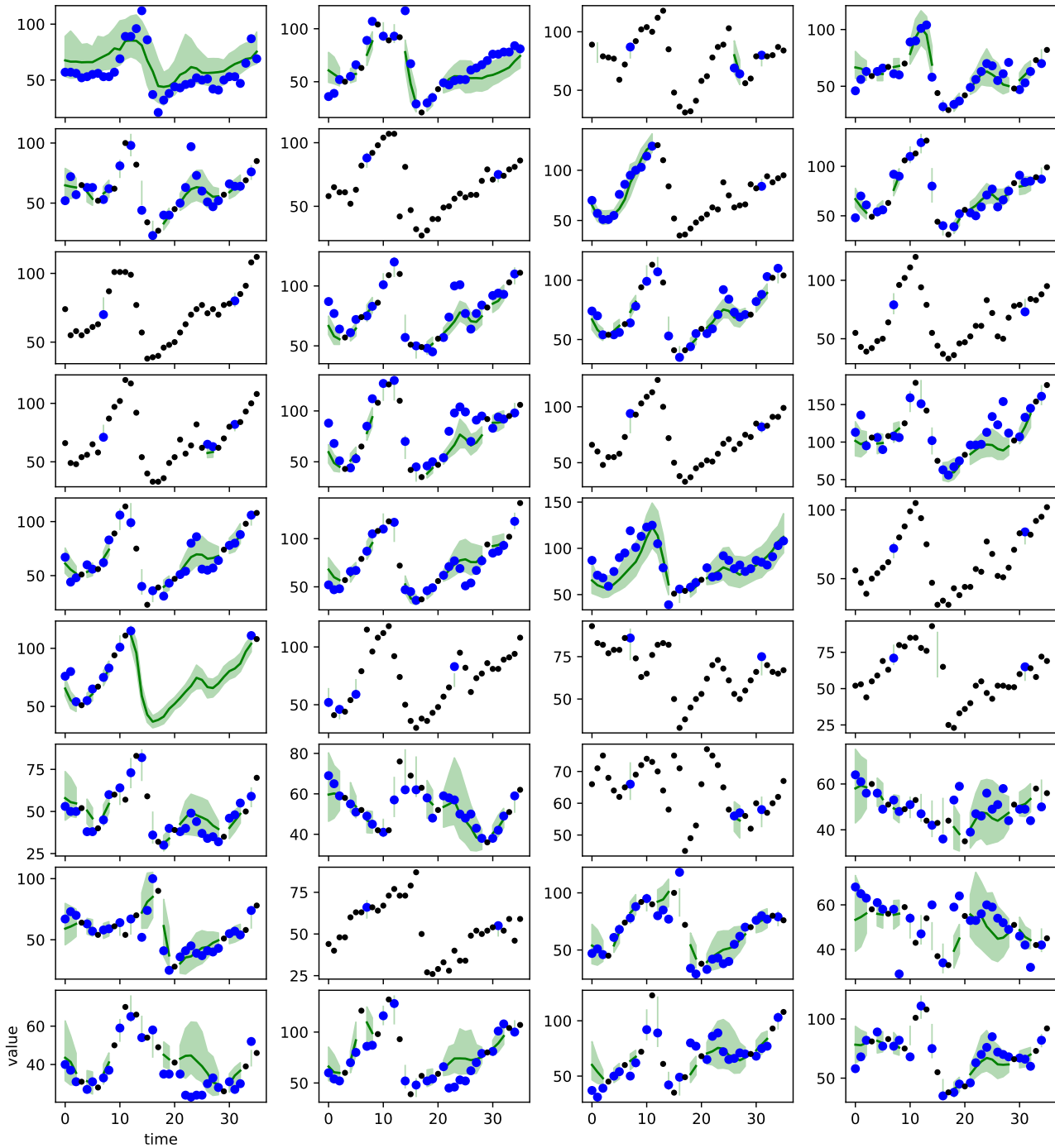
Figure 8. Example of imputation for PM2.5 dataset. The figure shows one sample with 36 features in each subplot and 36 time points. Smaller dark dots are conditions, larger blue dots are ground true values of the artificial missing values, the green belt shows the 80% confidence interval and the median curve of the imputed values. The time series data do not have observations at every time point, the missing ones are the unknowns.
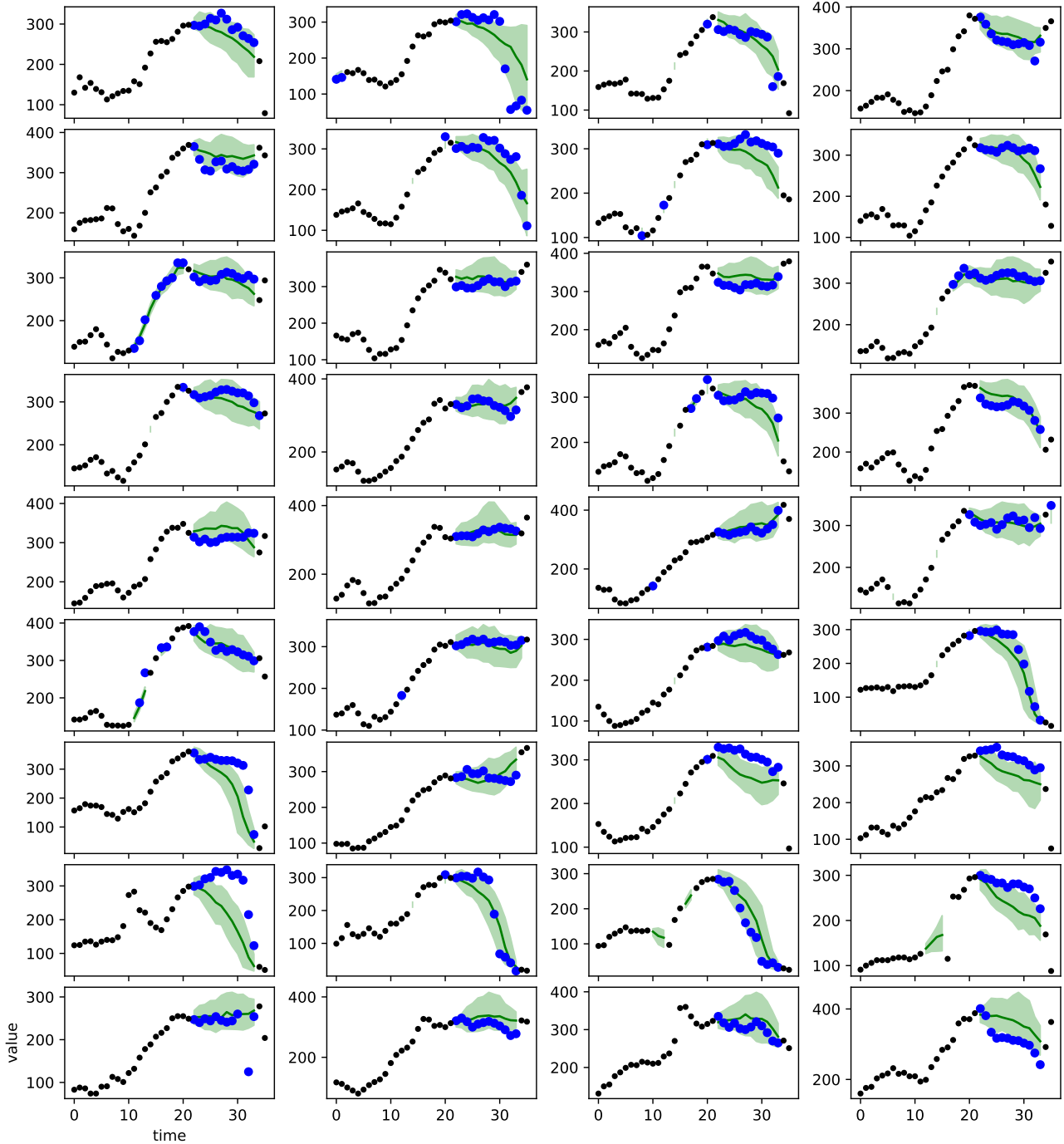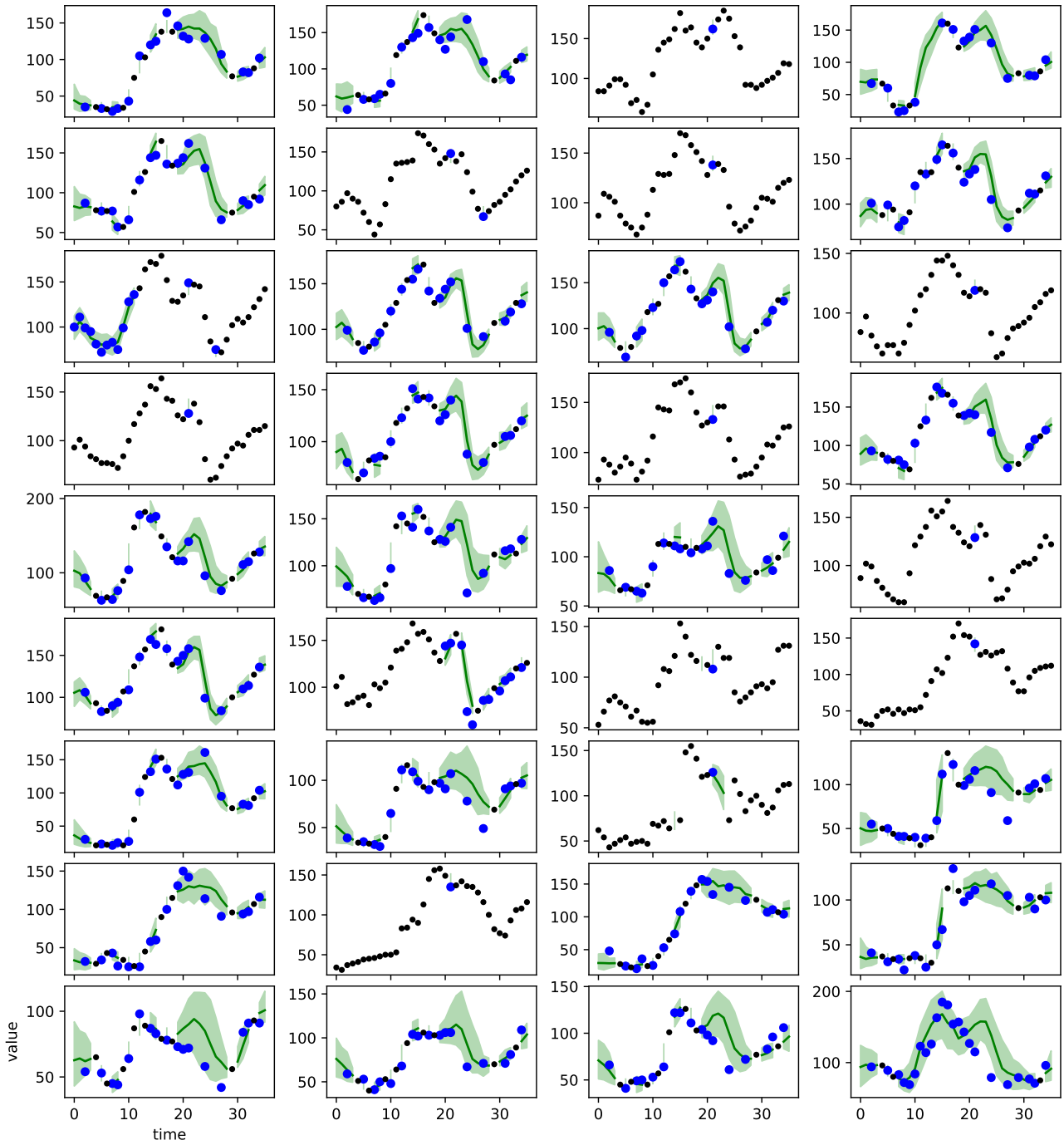
Figure 9. Example of imputation for PM2.5 dataset. The figure shows one sample with 36 features in each subplot and 36 time points. Smaller dark dots are conditions, larger blue dots are ground true values of the artificial missing values, the green belt shows the 80% confidence interval and the median curve of the imputed values. The time series data do not have observations at every time point, the missing ones are the unknowns.

Figure 10. Example of imputation for PM2.5 dataset. The figure shows one sample with 36 features in each subplot and 36 time points. Smaller dark dots are conditions, larger blue dots are ground true values of the artificial missing values, the green belt shows the 80% confidence interval and the median curve of the imputed values. The time series data do not have observations at every time point, the missing ones are the unknowns.
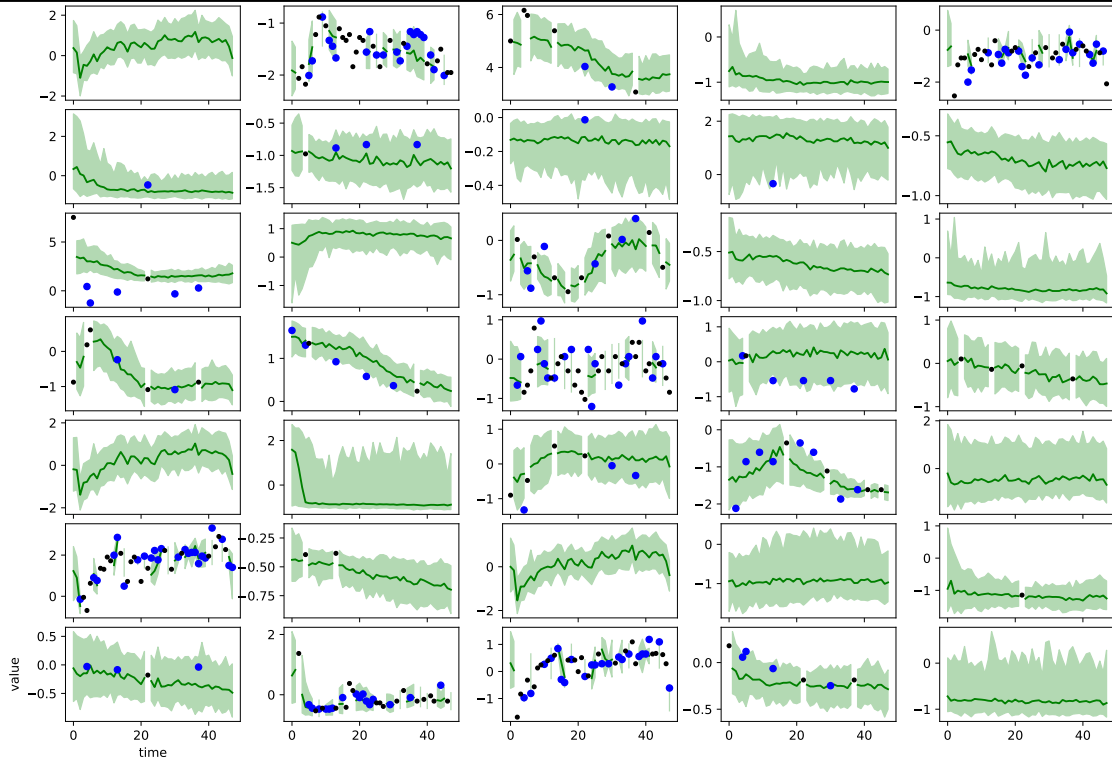
Figure 11. Example of imputation for PhysioNet dataset with artificial missing ratio 0.1. The missing ratio is the portion of selected artificial missing values among observations. The figure shows one sample with 35 features in each subplot and 48 time points. Smaller dark dots are conditions, larger blue dots are ground true values of the artificial missing values, the green belt shows the 80% confidence interval and the median curve of the imputed values. The time series data do not have observations at every time point, the missing ones are the unknowns. As the data is sparse with 80% of unknowns, the imputation has a very wide confidence band.
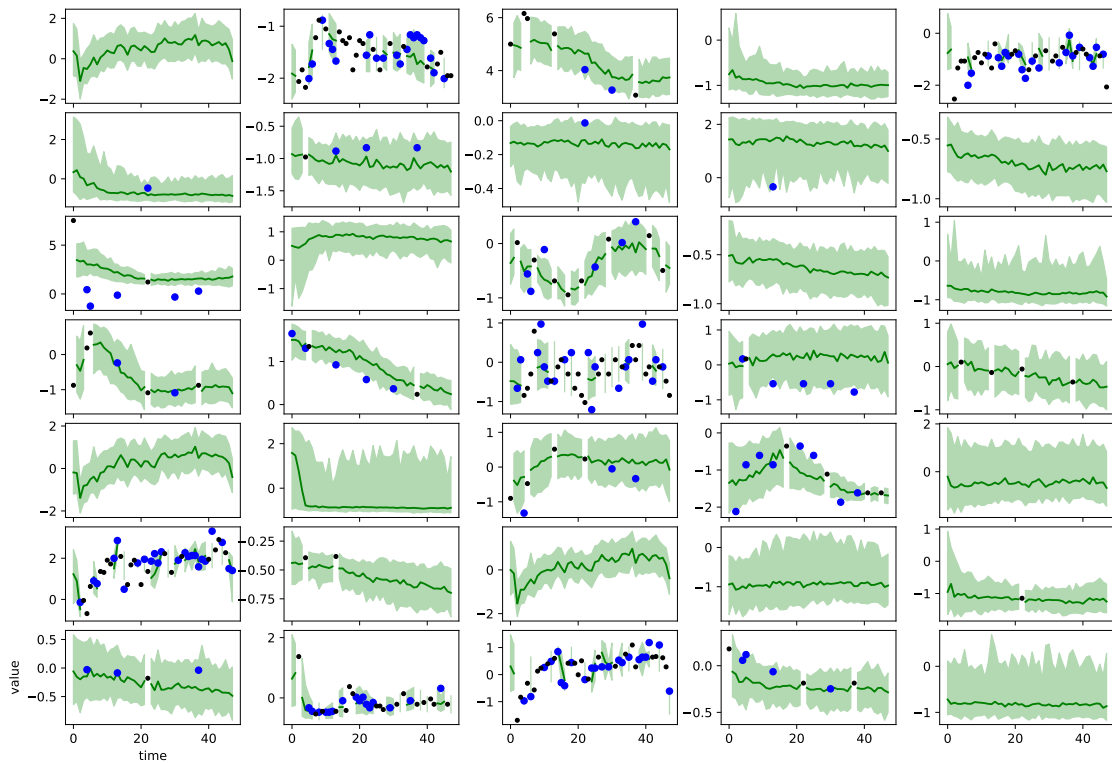

Figure 12. Example of imputation for PhysioNet dataset with artificial missing ratio 0.5.