IMPERIAL

Imperial College London
Department of Mathematics

Sequential Monte Carlo Guided Diffusion Sampling for General Optimization

Brendan Dowling

CID: 02458886

Supervised by Dr Deniz O. Akyildiz

August 17, 2024

Submitted in partial fulfilment of the requirements for the MSc in Statistics at Imperial College London

The work contained in this thesis is my own work unless otherwise stated.

Signed: Brendan Dowling Date: August 17, 2024

Acknowledgements

I want to give special thanks to my supervisor, Dr Deniz Akyildiz, who has provided excellent guidance over the course of this work. The dedication of his time towards answering my questions and suggesting avenues to explore went beyond expectations and was greatly appreciated. I also want to thank Benjamin Boys for meeting with me several times and providing insightful and guiding discussions related to this research. I also want to give special thanks to my family for their continued support of my studies and for providing me the resources needed to excel. Lastly, I want to thank my fiancée for her support over this challenging past year; without her this thesis could not have happened.

Abstract

Diffusion models have have been shown to learn underlying data distributions, even in high-dimensional settings as demonstrated by their state-of-the-art performance on image and video synthesis tasks. From a Bayesian perspective, this makes them suitable as prior models. Recent research has explored how to accurately *guide* the diffusion process for posterior sampling. Such methods have primarily focused on solving ill-posed inverse problems by approximating the time conditional score function to use in the reverse sampling steps. In this paper, we present a novel and general framework, `SMCOpt`, for guiding diffusion models with sequential Monte Carlo methods. The framework considers inverse problems as a special case of optimization, where we've reframed the optimization task as a sampling problem from an annealed Boltzmann-Gibbs distribution defined by the objective and diffusion model. This framework can be applied even in settings where gradients are not available, making it both computationally efficient and applicable in black-box optimization settings. Relatedly, this framework does not necessitate conditional score approximations though can flexibly accomodate doing so through different *twistings* of the proposal distribution. We demonstrate the efficacy of the algorithm through experiments on synthetic and real-world inverse problems and optimization tasks.

Contents

1. Introduction

2. Background

2.1. Notation

- $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ is the *state space*.
- $\mathcal{Y} \subseteq \mathbb{R}^{d_y}$ is the *measurement space*.

2.2. Inverse Problems

Definition 2.2.1 (Inverse Problem with Gaussian Noise)**.** Given a datapoint $\mathbf{x} \in \mathcal{X}$, denote some lossy measurement by

$$\mathbf{y} = g(\mathbf{x}) + \sigma_y \epsilon \in \mathcal{Y}, \quad \epsilon \sim \mathcal{N}\left(0, \mathbf{I}_{d_y}\right) \tag{1}$$

where $g : \mathcal{X} \to \mathcal{Y}$ is some *measurement operator*, and $\sigma_y \in \mathbb{R}^+$ controls the measurement noise. The goal of an inverse problem is to recover $\mathbf{x}$ from $\mathbf{y}$.

Definition 2.2.2 (Linear Inverse Problem)**.** A linear inverse problem is an inverse problem where the measurement operator is such that $g(\mathbf{x}) = A\mathbf{x}$ for some matrix $A \in \mathbb{R}^{d_y \times d_x}$ called the *measurement matrix*.

Example 2.2.3 (Gaussian Deblurring)**.** Suppose we have some $\mathbf{x}$ representing a flattened $h \times w$ image (ignoring channels for simplicity). Let $g(\mathbf{x})$ represent the convolution of $\mathbf{x}$ with a Gaussian kernel. Given the discrete nature of images, this kernel can be represented as a $k \times k$ matrix. Given that the convolution operator is linear, we can form some matrix A from the kernel matrix (by using it to form a block Toeplitz matrix) so that $g(\mathbf{x}) = A\mathbf{x}$. It follows that if we have some blurry image, $\mathbf{y}$, generated according to 1, the goal of inferring the unblurred image, $\mathbf{x}$, represents a linear inverse problem.

Remark 2.2.4 (Bayesian Solution to Ill-posed Inverse Problems)**.** Typically $d_x > d_y$, leading to a many-to-one $\mathbf{x} \to \mathbf{y}$ mapping (Chung et al., 2022), and requiring some *prior* information about $\mathbf{x}$. We call such an inverse problem *ill-posed*. If we assume some prior distribution of the data, $p(\mathbf{x})$, "solving" the inverse problem amounts to sampling from some posterior:

$$p(\mathbf{x} \mid \mathbf{y}) \propto p(\mathbf{x})g(\mathbf{y} \mid \mathbf{x}) = p(\mathbf{x})\mathcal{N}\left(\mathbf{y} \mid g(\mathbf{x}), \sigma_y^2 \mathbf{I}_{d_y}\right)$$

For many problems, $p(\mathbf{x})$ is generally unknown or does not conjugate with $g(\mathbf{y} \mid \mathbf{x})$, necessitating numerical methods to sample from $p(\mathbf{x} \mid \mathbf{y})$.

2.3. General Optimisation through Sampling

Definition 2.3.1 (Gibbs Measure / Boltzmann Distribution)**.** Let $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ be a state space and let $h : \mathcal{X} \to \mathbb{R}$ be a measurable function, called the *energy function* or *Hamiltonian*. The Gibbs measure is the probability density function over $\mathcal{X}$ given by:

$$q(\mathbf{x}; \beta) = \frac{\exp\{-\beta h(\mathbf{x})\}}{Z(\beta)} \tag{2}$$

where $\beta > 0$ is the *inverse temperature* parameter and $Z(\beta)$ is the normalizing constant (also known as the partition function), defined by:

$$Z(\beta) = \int_{\mathcal{X}} \exp\{-\beta h(\mathbf{x})\}\, d\mathbf{x}$$

Proposition 2.3.2 (Optimization via Sampling (Hwang, 1980; Kong et al., 2024))**.** Assume $h \in \mathrm{C}^3(\mathbb{R}^{d_x}, \mathbb{R})$ with $\{\mathbf{x}_i^*\}_{i=1}^M$ the set of its minimizers. Let p be a density on $\mathbb{R}^d$ such that there exists $i_0 \in \{1, \ldots, M\}$ with $p(\mathbf{x}_{i_0}^*) > 0$. Then Q_β, the distribution with density w.r.t the Lebesgue measure $\propto q(\mathbf{x}; \beta)p(\mathbf{x})$, weakly converges to Q_∞ as $\beta \to \infty$ and we have that:

$$Q_\infty = \frac{\sum_{i=1}^M a_i \delta_{\mathbf{x}_i^*}}{\sum_{i=1}^M a_i}$$

with $a_i = p(\mathbf{x}_i^*) \det(\nabla^2 h(\mathbf{x}_i^*))^{-1/2}$

Remark 2.3.3 (Annealing)**.** Proposition 2.3.2 tells us that by properly *tempering* (or *annealing*) β (i.e. slowly increasing it / reducing the temperature), we can globally optimize (minimize or maximize via negation) the function h. The density function p can be interpreted as some *prior* distribution we use to sample the points $\mathbf{x}$ (ideally such that more 'mass' is placed near the optima, $\{\mathbf{x}_i^*\}_{i=1}^M$).

2.4. Diffusion Models

Definition 2.4.1 (Stein Score)**.** Given a probability density function $p(\mathbf{x})$ on $\mathbb{R}^{d_x}$, the Stein score function is given by $\nabla_{\mathbf{x}} \log p(\mathbf{x})$, the gradient of the log-density with respect to the data (*not* the distribution's parameters).

Going forwards, we refer to the Stein score simply as the *score*, in-line with convention in related literature.

Definition 2.4.2 (Forwards Noising Process (Dou & Song, 2023))**.** Let $\mathbf{x}_0$ be a sample from some p_{data} distribution. Define the *forward noising process* as a Markov chain:

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} \mathcal{N}\left(a_t \mathbf{x}_{t-1}, b_t^2 \mathrm{I}_{d_x}\right)$$

where $\{(a_t, b_t)\}_{t=1}^T$ differ depending on formulation of the problem, and T is typically quite large (≈ 1000). a_t and b_t can be interpreted as the *memory retention factor* and *noise magintude* of the forward process which typically decrease and increase, respectively, with time.

Definition 2.4.3 (Forward Marginal (Dou & Song, 2023))**.** Given some forward noising process defined by $\{(a_t, b_t)\}_{t=1}^T$, the *forward marginal* is given by:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(c_t \mathbf{x}_0, d_t^2 \mathrm{I}_{d_x})$$

where c_t, d_t are derived from a_t and b_t. c_t essentially represents the signal strength (from the original sample) while d_t represents the cumulative noise. It follows that we should have $c_T \approx 0, d_T \approx 1$.

Definition 2.4.4 (Backwards Denoising Process (Dou & Song, 2023))**.** Given some forward noising process, we assume[1] some backwards process ($t = T \to 0$) to be likewise a Markov chain with one-step backwards transition kernel given by:

$$p(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(u_t \cdot \frac{\mathbf{x}_t + d_t^2 \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)}{c_t} + v_t \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t), w_t^2 \mathrm{I}_{d_x}\right) \tag{3}$$

with u_t, v_t some functions of a_t, b_t, and the mean derived from Tweedie's formula.

Remark 2.4.5 (DDPM Representation)**.** Under the denoising diffusion probabilistic model (DDPM) of Ho et al. (2020), we take:

$$a_t = \sqrt{\alpha_t} \quad b_t = \sqrt{\beta_t} \quad c_t = \sqrt{\overline{\alpha}_t} \quad d_t = \sqrt{1 - \overline{\alpha}_t} \tag{4}$$

and

$$u_t = \sqrt{\alpha_{t-1}} \quad v_t = -\sqrt{\alpha_t}(1 - \overline{\alpha}_{t-1}) \quad w_t = \sqrt{\beta_t \cdot \frac{1 - \overline{\alpha}_{t-1}}{1 - \overline{\alpha}_t}} \tag{5}$$

[1] This assumption is well-founded for large T since the forward SDE can be analytically reversed using the score thanks to the result of Anderson (1982). This, along with reducing the discretisation error of sampling from the SDE, is why we take T as large as feasbile.

with $\alpha_t = 1 - \beta_t$, for some $\beta_t \in (0, 1)$ known as the *noise schedule*, and $\overline{\alpha}_t = \prod_{\tau=1}^{t} \alpha_\tau$. For this paper, we primarily consider the DDPM formulation for simplicity. However, other formulations, such as denoising diffusion implicit models (DDIM) of J. Song et al. (2020) and Predictor-Corrector (PC) Y. Song et al. (2021), are equally applicable to our proposed framework; the only requirement is that the formulations conform to the above representations of the forwards and backwards process, and enable sampling at discrete time points.

Remark 2.4.6 (Noise Schedule)**.** Under the DDPM framework, the core parameter to tune is the noising schedule, β_t. Typically, we take this as an increasing function from some very low $\beta_{\min}$ to some $\beta_{\max}$. A linear schedule is the obvious and common choice, but other schedules, such as the cosine schedule of Nichol and Dhariwal (2021), can lead to better sampling performance. For this paper, the exact details of the schedule are not of signifcant importance.

Remark 2.4.7 (SDE Representation)**.** The DDPM approach corresponds to a discretized time rescaled Ornstein-Uhlenbeck process (Boys et al., 2023; Y. Song et al., 2021):

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

and is often referred to as the variance-preserving SDE (Y. Song et al., 2021). Going forwards, we will stick to the Markov (discretised) representation, but ultimately they are equivalent when it comes to sampling. This point is worth noting though as the SDE representation is what analytically justifies the backwards process (see 1).

Proposition 2.4.8 (Score to Noise Conversion)**.** Let $\mathbf{x}_t$ be a forward noised observation of clean data $\mathbf{x}_0$. Then, the score function satisfies the relationship:

$$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t) = -\frac{\epsilon_t}{b_t^2}.$$

Given 2.4.4 (and 2.4.8), it follows that if we have access to the score (or noise), we can sample from p_{data} by sampling $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I}_{d_x})$ ($\approx q(\mathbf{x}_T \mid \mathbf{x}_0)$) and iterating backwards in time according to Equation 3 until $t = 0$.

Unfortunately, $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$ is intractable[2]. As such, we train a score-approximating (Y. Song et al., 2021) (or noise-predicting (Ho et al., 2020), and apply 2.4.8) neural network, denoted $s_\theta(\mathbf{x}_t, t)$ ($\epsilon_\theta(\mathbf{x}_t, t)$). We defer to the extensive literature (foundationally Ho et al. (2020), Y. Song and Ermon (2020), Y. Song et al. (2021) and Nichol and Dhariwal (2021)) on how precisely to train such a network, but generally this is achieved via minmization of a "re-weighted variant of the evidence lower bound" (Y. Song et al., 2021):

$$\theta^* = \arg\min_\theta \sum_{t=1}^{T} (1 - \alpha_t) \mathbb{E}_{\mathbf{x}_0 \sim \hat{p}_{\text{data}}} \mathbb{E}_{\mathbf{x}_t \mid \mathbf{x}_0 \sim q_{\alpha_t}} \left[\| s_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_{\alpha_t}(\mathbf{x}_t \mid \mathbf{x}_0) \|_2^2 \right]$$

where $\hat{p}_{\text{data}}$ is the empirical data distribution (i.e. based on our training samples), and q_{α_t} is the forward marginal at time t. Plugging in this score network to Equation 3, we yield an approximate *backwards transition kernel*, $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \approx q(\mathbf{x}_{t-1} \mid x_t)$, which we use to sample backwards in time. Formally, we sample according to:

$$p_\theta(\mathbf{x}_0) = \int p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \, d\mathbf{x}_{1:T}, \quad p(\mathbf{x}_T) = \mathcal{N}(0, \mathbf{I}_{d_x}) \tag{6}$$

with, given a well trained network, $p_\theta(\mathbf{x}_0) \approx p_{\text{data}}(\mathbf{x}_0)$.

2.5. Sequential Monte Carlo

Definition 2.5.1 (State Space Model)**.** Let $\mathbf{x}_t \in \mathcal{X}$ be a *state vector* at time t, which encapsulates all the information about some system necessary to predict future states

[2]Note this is not $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_0)$, which is tractable but "useless" for sampling since we don't know $\mathbf{x}_0$ a priori (it's what we're trying to sample); this quantity is useful, however, for training where we *do* know $\mathbf{x}_0$.

(perhaps stochastically). Let $\mathbf{y}_t \in \mathcal{Y}$ be some observation vector at time t, representing (potentially noisy) measurements of the system. Let $\phi : \mathcal{X} \to \mathcal{X}$ be some function we call the *transition function*. Let $g : \mathcal{X} \to \mathcal{Y}$ be a measurement operator. Then the following dynamical system represents a (Gaussian) *state space model* (SSM):

$$\mathbf{X}_t \mid \mathbf{X}_{t-1} = \mathbf{x}_{t-1} \sim \mathcal{N}(\phi(\mathbf{x}_{t-1}), \sigma_x^2 \mathbf{I}_{d_x}) \tag{7}$$

$$\mathbf{Y}_t \mid \mathbf{X_t} = \mathbf{x}_t \sim \mathcal{N}(g(\mathbf{x}_t), \sigma_y^2 \mathbf{I}_{d_y}) \tag{8}$$

We refer to 7 as the *transition kernel*, with density $f(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, and 8 as the *likelihood*, with density $g(\mathbf{y}_t \mid \mathbf{x}_t)$.

Definition 2.5.2 (Bayesian Filtering)**.** Supopse we have access to measurements $\mathbf{y}_{0:t}$ emitted from some SSM. The task of using such measurements to infer the hidden states, $\mathbf{x}_{0:t}$, is known as *Bayesian filtering*. We may be interested in the *joint filtering distribution*, with density $p(\mathbf{x}_{0:t} \mid \mathbf{y}_{0:t})$, or the *filtering distribution*, with density $p(\mathbf{x}_t \mid \mathbf{y}_{0:t})$. Focusing on the latter, given the setup of 2.5.1, we can derive the two-step recursive relationship:

$$p(\mathbf{x}_t \mid \mathbf{y}_{0:t-1}) = \int p(\mathbf{x}_t, \mathbf{x}_{t-1} \mid \mathbf{y}_{0:t-1})\, d\mathbf{x}_{t-1} \tag{9}$$

$$= \int f(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{y}_{0:t-1})\, d\mathbf{x}_{t-1} \tag{10}$$

and

$$p(\mathbf{x}_t \mid \mathbf{y}_{0:t}) \propto p(\mathbf{x}_t, \mathbf{y}_t \mid \mathbf{y}_{0:t-1}) \tag{11}$$

$$\propto p(\mathbf{x}_t \mid \mathbf{y}_{0:t-1}) g(\mathbf{y}_t \mid \mathbf{x}_t) \tag{12}$$

Generally, however, the integeral in 10 and normalising constant in 12 are intractable, necessitating approximate methods.

Definition 2.5.3 (Sequential Monte Carlo[3] for Filtering)**.** Sequential Monte Carlo

[3] Also known as Particle Filtering.

(SMC) provides a method for approximately sampling from the sequence of distributions as described in 2.5.2 (Chopin & Papaspiliopoulos, 2020). It works by generating N particles, $\{\mathbf{x}_t^{(i)}\}_{i=1}^N$, according to some initial distribution, $r_0(\cdot)$, moving them according to some *proposal distribution*, $r_t(\cdot \mid \mathbf{x}_{t-1}, \mathbf{y}_t)$, weighting the particles according to some weighting function, $w_t(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{y}_t)$, and resampling[4] the particles according to their (self-normalised) weights $(W_t^{(i)})$. That is, the algorithm takes the following steps from $t = 1, \ldots, T$:

- **Propose**: $\tilde{\mathbf{x}}_t^{(i)} \sim r_t(\cdot \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t), \quad i = 1, \ldots, N$
- **Weight**: $w_t^{(i)} \leftarrow w_t(\tilde{\mathbf{x}}_t^{(i)}, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t), \quad i = 1, \ldots, N$
- **Resample**: $\{\mathbf{x}_{0:t}^{(i)}\}_{i=1}^N \sim \text{Multinomial}\left(\{\tilde{\mathbf{x}}_{0:t}^{(i)}\}_{i=1}^N; \{w_t^{(i)}\}_{i=1}^N\right)$

At the last step, we can yield an empirical distribution over the particles:

$$\hat{p}(\mathbf{x}_T \mid \mathbf{y}_{0:T}) = \sum_{i=1}^{N} W_T^{(i)} \delta_{\mathbf{x}_T^{(i)}(\mathbf{x}_T)}$$

which asymptotically (in N) approximates the true posterior distribution, $p(\mathbf{x}_T \mid \mathbf{y}_{0:T})$ (Chopin & Papaspiliopoulos, 2020; Del Moral, 2011).

Remark 2.5.4 (Weight Function)**.** The weight function's purpose is to correct for discrepancies between the proposal and true posterior, $p(x_t \mid \mathbf{x}_{t-1}, \mathbf{y}_t)$, which it aims to approximate. In the filtering case for an SSM as in 2.5.1, the weight function is given by the ratio of the two (Chopin & Papaspiliopoulos, 2020):

$$\begin{aligned} w_t^{(i)} &= \frac{p(\tilde{\mathbf{x}}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}{r_t(\tilde{\mathbf{x}}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)} \\ &\propto \frac{f(\tilde{\mathbf{x}}_t \mid \mathbf{x}_{t-1}^{(i)}) g(\mathbf{y}_t \mid \tilde{\mathbf{x}}_t^{(i)})}{r_t(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_t)} \end{aligned}$$

[4] We present multinomial resampling for simplicity, but in practice alternative methods, such as systematic resampling, are used due to inefficiencies in the multinomial scheme (Chopin & Papaspiliopoulos, 2020)

Remark 2.5.5 (Resampling)**.** The purpose of resampling is to avoid the issue of weight-degeneracy. Without resampling, after several iterations (i.e. for large T), most particles will end up having negligible weight (since without resamping we need to multiply all previous weights for the particle to get its time τ weight) and eventually a single particle will dominate the approximation. Without resampling, we would not get the desired asymptotic results.

2.6. Related Work

The goal of conditionally sampling from p_{data} based on some $\mathbf{y}$ has been the center of significant recent research. The principle approach of most methods is to approximate $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{y})$, and plug this in to 3 in place of its unconditional counterpart. The obvious approach is to train a conditional model to learn some $s_\theta(\mathbf{x}_t, \mathbf{y}, t)$ and use this. Indeed, this is the approach that modern text-to-image, image-to-image, and image-to-video take (Li et al., 2023; Nichol et al., 2021; Rombach et al., 2021; Saharia et al., 2021). This approach generally achieves the best conditional sampling but is limited in application; it requires training a model specific to each particular inverse problem (of which the aforementioned are examples) which may be costly. Furthermore, such training requires having labelled data which may impose limitations in certain settings and for certain tasks.

A separate branch of research, and the area with which this work aligns, considers taking a pre-trained *unconditional* diffusion model (i.e. one where we just have a parameterised estimator of $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$) and *guide* the diffusion process to enable conditional samping. This is typically achieved by exploiting the following relationship:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t \mid \mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y} \mid \mathbf{x}_t) \tag{13}$$

Replacing the first term with our pre-trained score network, optimal guidance can be achieved then by well-approximating $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y} \mid \mathbf{x}_t)$ (which is intractable). Such ap-

proximation methods are generally non-specific to the inverse problem, enabling taking any unconditional diffusion model and using it to sample from the desired posterior distribution. The diffusion-posterior-sampling method of Chung et al. (2022) was foundational, and can be applied to solve general inverse problems. J. Song et al. (2023) and, recently, Boys et al. (2023) have iterated on this with more analytically refined[5] approaches (though these are only applicable to linear inverse problems).

Given the difficulty of such approximations, and the sequential nature of the sampling procedure from diffusion models, significant recent research (Cardoso et al., 2023; Dou & Song, 2023; Janati et al., 2024; Trippe et al., 2023; Wu et al., 2023) has considered using SMC methods to instead more directly target the posterior (or rather, the intermediate distributions $p(\mathbf{x}_t \mid \mathbf{y})$) and circumvent the need to compute this term. Our framework is most heavily inspired by the scheme of Cardoso et al. (2023); in 3.3 we demonstrate how their scheme is essentially a special case of our framework. In fact, each of these methods can be applied under our framework since their differences essentially lie primarily in their choice of proposal distribution.

However, our framework is generalised to solving any optimization task (not just inverse problems). Through this lens, our framework's objective and formulation aligns with that of Kong et al., 2024, where we consider using interacting particle systems (i.e. SMC) as a means to circumvent the need to compute gradients of the objective function. This makes our method even better suited to black-box optimization tasks since it removes the need to train a surrogate model.

[5] See Boys et al., 2023 Section 5 for an excellent summary of how the methods co-relate.

3. SMC-Guided Conditional Diffusion Sampling

3.1. `SMCOpt` for General Optimization Problems

Text

3.2. Inverse Problems as a Special Case

Text

3.3. Relation to Related Work

Text

4. Experiments and Results

4.1. Gaussian Mixture Model Inverse Problem Experiment

Text

4.2. Branin Function Optimization Experiment

Text

4.3. Black-Box Optimization Experiment

Text

5. Discussion

Tips for the Discussion section.

- Begin your Discussion with a summary and the main findings of your research in 2-5 sentences.
- Describe the implications of your main findings in the context of existing work concisely and precisely using scientific language.
- Describe the limitations in your statistical methods and your main findings. Be honest about the limitations in your approach, and substantiate what could have been done differently as needed. Explain if your main findings are robust or sensitive to these limitations.
- Avoid Subsections and Subsubsections in the Discussion.
- In the last paragraph, conclude your report with a pitch using plain language that summarises the key implications of your research in the context of previous work. Write this last paragraph for the Imperial Press Office or journalists as audience.
- Aim for approximately 1-3 pages, similar in style to a general science or statistics research paper.

5.1. Future Work

Text

5.2. Conclusion

Text

6. Endmatter

All code for the research is openly available on a GitHub repository. All results in this paper can be easily reproduced by following the instructions of said repository's `README`. Code was run on an NVIDIA RTX3080.

Black-box data and oracle models are available originally through the `design-bench` repository.

References

Anderson, B. D. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, *12*(3), 313–326. https://doi.org/10.1016/0304-4149(82)90051-5

Boys, B., Girolami, M., Pidstrigach, J., Reich, S., Mosca, A., & Akyildiz, O. D. (2023, November 22). *Tweedie Moment Projected Diffusions For Inverse Problems*. arXiv: 2310.06721 [stat]. Retrieved June 23, 2024, from http://arxiv.org/abs/2310.06721

Cardoso, G., Idrissi, Y. J. E., Corff, S. L., & Moulines, E. (2023, October 25). *Monte Carlo guided Diffusion for Bayesian linear inverse problems*. arXiv: 2308.07983 [cs, stat]. https://doi.org/10.48550/arXiv.2308.07983

Chopin, N., & Papaspiliopoulos, O. (2020). *An introduction to sequential Monte Carlo*. Springer.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2022). Diffusion Posterior Sampling for General Noisy Inverse Problems. https://doi.org/10.48550/ARXIV.2209.14687

Del Moral, P. (2011). Central Limit Theorems. In *Feynman-Kac formulae: Genealogical and interacting particle systems with applications* (pp. 291–330). Springer. OCLC: 1063493341.

Dou, Z., & Song, Y. (2023). Diffusion Posterior Sampling for Linear Inverse Problem Solving: A Filtering Perspective. Retrieved June 8, 2024, from https://openreview.net/forum?id=tplXNcHZs1&referrer=%5Bthe%20profile%20of%20Yang%20Song%5D(%2Fprofile%3Fid%3D~Yang_Song1)

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. https://doi.org/10.48550/ARXIV.2006.11239

Hwang, C.-R. (1980). Laplace's Method Revisited: Weak Convergence of Probability Measures. *The Annals of Probability*, *8*(6). https://doi.org/10.1214/aop/1176994579

Janati, Y., Durmus, A., Moulines, E., & Olsson, J. (2024, March 17). *Divide-and-Conquer Posterior Sampling for Denoising Diffusion Priors*. arXiv: 2403.11407 [cs, stat]. Retrieved May 24, 2024, from http://arxiv.org/abs/2403.11407

Kong, L., Du, Y., Mu, W., Neklyudov, K., De Bortoli, V., Wang, H., Wu, D., Ferber, A., Ma, Y.-A., Gomes, C. P., & Zhang, C. (2024, April 29). *Diffusion Models as Constrained Samplers for Optimization with Unknown Constraints*. arXiv: 2402.18012 [cs]. Retrieved July 26, 2024, from http://arxiv.org/abs/2402.18012

Li, X., Ren, Y., Jin, X., Lan, C., Wang, X., Zeng, W., Wang, X., & Chen, Z. (2023). Diffusion Models for Image Restoration and Enhancement – A Comprehensive Survey. https://doi.org/10.48550/ARXIV.2308.09388

Nichol, A., & Dhariwal, P. (2021, February 18). *Improved Denoising Diffusion Probabilistic Models*. arXiv: 2102.09672 [cs, stat]. Retrieved May 29, 2024, from http://arxiv.org/abs/2102.09672

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., & Chen, M. (2021). GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. https://doi.org/10.48550/ARXIV.2112.10741

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). High-Resolution Image Synthesis with Latent Diffusion Models. https://doi.org/10.48550/ARXIV.2112.10752

Saharia, C., Chan, W., Chang, H., Lee, C. A., Ho, J., Salimans, T., Fleet, D. J., & Norouzi, M. (2021). Palette: Image-to-Image Diffusion Models. https://doi.org/10.48550/ARXIV.2111.05826

Song, J., Meng, C., & Ermon, S. (2020). Denoising Diffusion Implicit Models. https://doi.org/10.48550/ARXIV.2010.02502

Song, J., Vahdat, A., Mardani, M., & Kautz, J. (2023). Pseudoinverse-guided diffusion models for inverse problems. *International Conference on Learning Representations*. https://openreview.net/forum?id=9_gsMA8MRKQ

Song, Y., & Ermon, S. (2020, October 10). *Generative Modeling by Estimating Gradients of the Data Distribution*. arXiv: 1907.05600 [cs, stat]. https://doi.org/10.48550/arXiv.1907.05600

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021, February 10). *Score-Based Generative Modeling through Stochastic Differential Equations*. arXiv: 2011.13456 [cs, stat]. Retrieved May 31, 2024, from http://arxiv.org/abs/2011.13456

Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., & Jaakkola, T. (2023, March 19). *Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem*. arXiv: 2206.04119 [cs, q-bio, stat]. Retrieved June 5, 2024, from http://arxiv.org/abs/2206.04119

Wu, L., Trippe, B. L., Naesseth, C. A., Blei, D. M., & Cunningham, J. P. (2023, June 30). *Practical and Asymptotically Exact Conditional Sampling in Diffusion Models*. arXiv: 2306.17775 [cs, q-bio, stat]. Retrieved June 5, 2024, from http://arxiv.org/abs/2306.17775

Appendices

A. Figures

Text

B. Tables

Text

C. Proofs

Text

D. Extra

Remark D.1 (DDIM Representation)**.** Under DDIM:

$$u_t = \sqrt{\alpha_{t-1}} \quad v_t = -\sqrt{1 - \overline{\alpha}_{t-1} - \sigma_t^2}\sqrt{1 - \overline{\alpha}_t} \quad w_t = \sigma_t$$

with $\{\sigma_t\}_{t=1}^T$ an arbitrary conditional variance sequence. It's common to consider:

$$\sigma_t(\eta) = \eta\sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}}\sqrt{1-\frac{\alpha_t}{\alpha_{t-1}}}, \quad \eta \in [0,1]$$

with $\eta = 1$ ultimately reducing u_t, v_t, w_t to the DDPM values, and $\eta = 0$ corresponding to deterministic generation (J. Song et al., 2020).

Remark D.2 (Time Respacing)**.** One of the remarkable features of the DDIM algorithm is it enabling *time respacing* whereby we can sample from the backwards process in fewer timesteps. In this paper, we don't consider such respacing though in principle our methodology does not prohibit it.