Generative modeling with probabilistic constraints

Anonymous Authors[1]

Abstract

Generative models such as diffusion models and transformers are powerful tools for learning complex data distributions and generating new samples. However, their black-box nature limits interpretability, and the learned distributions may violate side knowledge arising from domain expertise. We represent such side knowledge as probability distributions over noisy functions of the modeled objects and seek to minimally adjust the generative model to satisfy such constraints. Our approach is to optimize the dual of the corresponding constrained optimization problem, encoding the infinite-dimensional dual variable using a neural network. We introduce a simple and efficient score-based method for fitting the parameters of this neural network, and for simulating from the resulting adjusted distribution. We evaluate our approach on a number of synthetic tasks, as well on two real-world problems: a regularized nonparametric maximum likelihood estimation problem, and the incorporation of class-level fairness constraints into image diffusion models.

1. Introduction

Generative models form powerful tools both for learning complex and high-dimensional data distributions, and then simulating new samples. Recent years have seen significant progress in the flexibility and applicability of such models, driven by deep neural network models such as Variational Autoencoders (VAEs) (Kingma & Welling, 2013), Generative Adversarial Networks (GANs) (Goodfellow et al., 2020), diffusion models (Song et al., 2021; Ho et al., 2020) and transformers (Vaswani et al., 2017), as well as probabilistic and stochastic process models like deep hierarchical Bayesian (Gelman et al., 2013) and nonparametric Bayesian models (Ghosal & Van der Vaart, 2017). However, these complex black-box models are usually hard to interpret, and they often have implications that are inconsistent with prior knowledge and domain requirements. We formalize 'model implications' as the *probability* distributions induced by known *probabilistic transformations* of the objects being modeled. The outputs of such transformations are typically much lower dimensional than the object being modeled, and we refer to their distribution a *marginal distribution*. In this work, we consider situations where the modeler seeks to constrain black-box generative models so that these marginal distributions match some other known distribution derived from domain knowledge. We impose three desiderata:

D1. They must satisfy the probabilistic marginal constraints
D2. They must stay as faithful to the original model as possible
D3. They must not require retraining the original model

Our problem is an important component of a variety of modern and classical problems in statistics and machine learning. We consider two motivating examples in this paper:

P1: Class-level fairness in generative models: Samples from generative models reflect the data used to train them. An active area of machine learning research focuses on steering generative models away from biases present in underlying datasets, so that the generated samples better reflect real-world or domain-specific distributions. Work here focuses on topics like fairness, bias, trustworthiness and responsibility in AI/ML, domain shift and privacy (among many others) (Barocas et al., 2023; Li et al., 2023; Choi et al., 2020; Kim et al., 2024). E.g., consider the distribution of image classes (whether semantic categories (e.g., "cat," "dog," "other") or attribute labels (e.g., "male," "female," "other")) from a pretrained generative model: one might seek to modify this to better reflect an underlying population. Often, image classes are latent and ambigious, and identifying them requires relying on imperfect classifiers.

P2: Nonparametric regularized maximum likelihood estimate: NPMLE is a classical approach (Laird, 1978; Lindsay, 1995) that has seen a resurgence of interest. This seeks to estimate the distribution of a latent variable X given noisy observations Y (often assumed to be additive noise on some transformation $f(X)$ of X). While it affords flexibility by avoiding parametric assumptions about the distribution of X, it does not incorporate prior knowledge

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

about this (especially important in ill-posed deconvolution problems). Further, the resulting estimate is typically a discrete distribution. Wilkins-Reeves et al. (2021) requires the nonparametric estimate to be close to a reference distribution, with the associated marginal over Y matching the suitably smoothed empirical distribution of Y.

Related challenges also arise in other applications. In domains such as healthcare and mobility analysis, generative models are increasingly used to synthesize realistic data for downstream analysis or data sharing. Models trained solely for maximal fidelity may inadvertently reproduce individual-level patterns, raising concerns about privacy leakage. By incorporating constraints on the distribution of aggregate statistics or risk measures related to individual identifiability, one can enforce privacy-preserving behavior while still generating samples that remain probabilistically close to the observed data (Schifeling & Reiter, 2016).

2. Problem Setup

Let $p_X^0(x)$ be the probability density of a random variable $X \in \mathcal{X}$ implied by some black-box generative model such as a diffusion model or a transformer, or by some complex (e.g. nonparametric) prior distribution. We will refer to p_X^0 as the *reference density*[1].Let $Y \in \mathcal{Y}$ be a random variable whose conditional distribution given X, $p_{Y|X}(y|x)$, is known. We refer to the latter as the *transformation density*. We can view this as resulting from a known, lossy and potentially noisy transformation $Y = f(X, \xi)$ of X, where ξ is some auxiliary random variable. Typically Y is lower dimensional than X, and can even be categorical (in this case, we continue to use densities, now with respect to the uniform discrete measure).

Together, the reference density $p_X^0(x)$ and the transformation density $p_{Y|X}(y|x)$ induce a marginal probability $p_Y^0(y) = \int_{\mathcal{X}} p_{Y|X}(y|x)p_X^0(x)dx := p_{Y|X} \circ p_X^0$ on Y. Often, the modeler has knowledge or has requirements about how the random variable Y is distributed. We write this as a known density $p_Y^1(Y)$, which we call the *constraint density*. In general, we cannot expect the p_Y^0 implied by p_X^0 to agree with p_Y^1. Examples of such side knowledge are:
a) Y corresponds to a smaller component of some larger stochastic system X whose behavior is well understood,
b) Y is constrained by factors like fairness or privacy to follow some prescribed distribution (e.g. the distribution of classes from an image generator must be balanced), or
c) Y corresponds to noisy measurements of X, p_Y^1 is the (smoothed) empirical distribution of a finite dataset of Y's.

In the simplest version of our task, we seek to find a distribution $p_X^*(x)$ that is as close as possible to the reference density p_X^0 (in terms of KL divergence), while ensuring that the induced marginal $p_Y^*(y) := p_{Y|X} \circ p_X^*$ equals the constraint density p_Y^1. We note that matching the constraint density $p_Y^1(y)$ need not be feasible for any transition density $p_{Y|X}(y|x)$. Two counterexamples are: 1) if the union of the support of $p_{Y|X}(y|x)$ across all $x \in \mathcal{X}$ is strictly contained within $p_Y^1(y)$, and 2) if the variance introduced by $p_{Y|X}(y|x)$ exceeds the variance of $p_Y^1(y)$. More generally, for this problem to be feasible, p_Y^1 must lie in the convex hull of $p_{Y|X}(y|x), x \in \mathcal{X}$, and with few exceptions, this is not easy to check. One exception is when Y is a deterministic function of X, now, a sufficient condition for feasibility is simply that the range of the function f equals $\mathcal{Y}$.

With this in mind, we relax our problem formulation as below; this is the formal optimization problem we consider:

$$\begin{aligned} p_X^* = \underset{p_X \in \mathcal{P}_{\mathcal{X}}}{\arg\min} \quad & D_{\mathrm{KL}}(p_X \| p_X^0) \\ \text{subject to} \quad & \|p_Y^1 - p_{Y|X} \circ p_X\|_2 \leq \varepsilon. \end{aligned} \tag{1}$$

Above $\mathcal{P}_{\mathcal{X}} = \left\{p_X \geq 0 : \int_{\mathcal{X}} p_X(x)\,dx = 1\right\}$ is the space of all probability densities on $\mathcal{X}$. Thus we seek to find the density on $\mathcal{X}$ closest to p_X^0 in the Kullback-Leibler sense, with the associated marginal over $\mathcal{Y}$ remaining within a ε L^2 ball of p_Y^1. More generally, one may replace the L^2-constraint by an L^r-constraint $\|p_Y^1 - p_Y^*\|_r$ for any $r \in [1, \infty]$. When $\varepsilon = 0$, the precise choice of norm is less critical, as all such norms induce equality of distributions.

Even relaxing the marginal constraint this way does not ensure feasibility. For any pair $(p_{Y|X}, p_Y^1)$, we can define an associated *constraint gap* as $\varepsilon_* := \inf_{p_X \in \mathcal{P}_{\mathcal{X}}(p_X^0)} \|p_Y^1 - p_{Y|X} \circ p_X\|_2$, where $\mathcal{P}_{\mathcal{X}}(p_X^0) = \{p_X \in \mathcal{P}_X \text{ s.t. } D_{\mathrm{KL}}(p_X \| p_X^0) < \infty\}$. We say Problem (1) is *feasible* if $\varepsilon \geq \varepsilon_*$, and *infeasible* otherwise.

In real applications, such as those in Section 4, ε_* is unknown, making it hard to set up a problem that is feasible. Now, constraints are typically understood in a "the tighter, the better" manner, and one can try to find ε_* via grid or annealing search. While this paper primarily focuses on the first regime, our proposed dual optimization shows promising empirical performance even in the second regime.

3. Methodology

As stated in Equation (1), our problem presents significant challenges: it is an infinite-dimensional optimization problem subject to two sets of constraints: the marginal constraint on p_Y as well as the fact that the optimization variable is constrained to the (infinite-dimensional) probability simplex. Rather than directly attempting to solve it, we apply Lemma 11 of Altun & Smola (2006), adapted to our notation and setting below, allowing us to write down and optimize a corresponding dual problem:

[1]We will work with densities and treat $\mathcal{X} \in \mathbb{R}^d$, though our ideas are more generally applicable

Lemma 3.1. *Let p_X^0 and p_Y^1 be two probability densities on domain $\mathcal{X}$ and $\mathcal{Y}$, respectively, and $p_{Y|X} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ be a conditional probability density. For any $\varepsilon \geq 0$, we have*

$$\min_{p_X \in \mathcal{P}_\mathcal{X}} \left\{ D_{\mathrm{KL}}(p_X \| p_X^0) \text{ s.t. } \left\| p_Y^1 - p_{Y|X} \circ p_X \right\|_2 \leq \varepsilon \right\} \quad (2)$$

$$= \max_{\phi \in L^2(\mathcal{Y})} \left\{ \left\langle \phi(y), p_Y^1(y) \right\rangle - \varepsilon \|\phi\|_2 + e^{-1} \right.$$

$$\left. - \log \int_\mathcal{X} p_X^0(x) \exp\left(\left\langle \phi(y), p_{Y|X}(y|x) \right\rangle \right) dx \right\} \quad (3)$$

When (2) is feasible (i.e., $\varepsilon \geq \varepsilon_*$), the unique solution is

$$p_X^*(x) = p_X^0(x) \exp\left(\int_\mathcal{Y} \phi^*(y) p_{Y|X}(y|x)\, dy \right) \Big/ C_{\phi^*} \quad (4)$$

where ϕ^* solves (3) and C_{ϕ^*} is the normalizing constant.

Note that when (2) is not feasible, the dual objective is unbounded, and a dual optimization that diverges helps identify feasibility. Before we discuss how we optimize the dual, we mention some implications of the result above. For the special case of $\varepsilon = 0$, and a deterministic transformation $Y = f(X)$ (so that $p_{Y|X} = \delta_{y=f(x)}$), the above reduces to the following:

Corollary 3.2. *Define the ε p_Y^1-approximation to p_Y^0 as $\widetilde{p}_Y^\varepsilon = \arg\min_{\|p_Y - p_Y^1\|_2 \leq \varepsilon} D_{\mathrm{KL}}(p_Y || p_Y^0)$. When $\varepsilon = 0$, $\widetilde{p}_Y^\varepsilon = p_Y^1$. Then, in the case of a deterministic transformation, where $p_{Y|X} = \delta_{y=f(x)}$ for some function $f : \mathcal{X} \to \mathcal{Y}$, the solution to Problem (1), is given by $p_X^*(x) = p_X^0(x) \frac{\widetilde{p}_Y^\varepsilon(f(x))}{p_Y^0(f(x))} = \widetilde{p}_Y^\varepsilon(f(x)) p_{X|Y}^0(x|f(x))$.*

For $\varepsilon = 0$, there are a few instances of this result being used in the literature (either explicitly or implicitly) (Kessler et al., 2015; Choi et al., 2020; Tang & Rao, 2025). This is exploited to produce two simple approaches to solve (1):

Conditional sampling: first simulate Y from p_Y^1 (ensuring the constraint is automatically satisfied), and given this, conditionally simulate X as specified by the reference density p_X^0. For instance, when Y is a class label and X is an image, one might draw a class-conditioned image.

Importance reweighting: estimate the marginal distributions over Y, and use these as importance weights to 'correct' samples from the reference distribution p^0 as above. Since p_Y^1 is part of the problem specification, we only need to estimate that marginal density p_Y^0 of the reference p_X^0.

While it is tempting to use one of the above schemes for the general setting with stochastic transformations, this can be suboptimal as the results below formalize:

Corollary 3.3. *Consider the general setting where $p_{Y|X}(y|x)$ is non-degenerate, and $\varepsilon \geq 0$. Then, if the Problem (1) is feasible, its solution can be written as the marginal of $p_Y^1(y) p_{X|Y}^0(x|y) w(x,y)$, where $w(x,y) = \frac{p_Y^0(y)}{p_Y^1(y)} \frac{p_X^*(x)}{p_X^0(x)}$.*

We can easily see that the solution for deterministic transformations in Corollary 3.2 arises as a special case of this result. For general $p_{Y|X}(y|x)$, $w(x,y)$ is not a constant, and ignoring this in the two schemes above will not only result in suboptimal solutions to Problem (1) but also solutions that no longer satisfy the marginal constraint p_Y^1. For example, let $p_X^0 = \mathcal{N}(0, 1^2)$, $p_{Y|X=x} = \mathcal{N}(2x+1, 1^2)$ and $p_Y^1 = \mathcal{N}(-1, 2^2)$. The optimal solution for $\varepsilon = 0$ can be shown to be $p_X^* = \mathcal{N}(-1, \frac{3}{4})$ (see Section E.1). However, the conditional simulation approach yields $p_X^{\text{det}}(x) := \int_\mathcal{Y} p_Y^1(y)\, p_{X|Y}^0(x|y) dy = \mathcal{N}(-\frac{4}{5}, \frac{21}{25})$, resulting in $p_Y^{\text{det}}(y) := \int_\mathcal{X} p_{Y|X}(y|x)\ p_X^{\text{det}}(x) dx = \mathcal{N}(-\frac{3}{5}, \frac{109}{25}) \neq p_Y^1$. In general, we have the following result:

Proposition 3.4. *Write $p_X^{det}(x) = p_X^0(x) \frac{p_Y^1(f(x))}{p_Y^0(f(x))} = p_Y^1(f(x)) p_{X|Y}^0(x|f(x))$ for the conditional sampling solution, corresponding to Problem (1) when Y is a deterministic function of X. For the general problem, we can always find a p_Y^1 such that the Y-marginals of this solution have L_2 distance from p_Y^1 that is greater than zero.*

The takeaway is that practitioners must by careful about extending schemes from the setting of deterministic transformations to probabilistic ones. Below, we describe a solution to the general problem.

3.1. Proposed approach

Our approach is to solve the general problem is to work with the unconstrained dual problem. We represent the infinite-dimensional dual variable $\phi \in L^2(\mathcal{Y})$ with a neural network $\phi_\theta(y) : \Theta \times \mathcal{Y} \to \mathbb{R}$ with sufficient expressive power. Here, θ are the neural network parameters, and the dual objective (3), negated from here onwards, becomes:

$$L(\theta) := \left\{ \varepsilon \left(\int_\mathcal{Y} \phi_\theta^2(y)\, dy \right)^{\frac{1}{2}} - \int_\mathcal{Y} \phi_\theta(y) p_Y^1(y)\, dy \right.$$

$$\left. + \log \int_\mathcal{X} p_X^0(x) \cdot \exp\left(\int_\mathcal{Y} \phi_\theta(y) p_{Y|X}(y|x)\, dy \right) dx \right\}$$

$$:= L_{\text{reg}}(\theta) + L_1(\theta) + L_2(\theta). \quad (5)$$

For any set of parameters θ, define the associated density

$$p_X(x;\theta) = p_X^0(x) \exp\left(\int_\mathcal{Y} \phi_\theta(y) p_{Y|X}(y|x)\, dy \right) \Big/ C_\theta$$

$$=: p_X^0(x) w_\theta(x) / C_\theta. \quad (6)$$

Now, for any $\theta^* \in \{\theta : \arg\min_{\theta \in \Theta} L(\theta)\}$, our approximation to the solution p_X^* of (2) is given by $\widehat{p_X^*}(x) \approx p_X(x;\theta^*)$. The approximation error of this approach reduces with the expressiveness of the neural network $\phi_\theta(y)$. A natural strategy to optimize the dual is by gradient descent, requiring us to calculate the gradient $\nabla_\theta L(\theta)$. While evaluating this is intractable, we show that it is fairly straightforward to obtain Monte Carlo estimates of this quantity,

allowing a fairly simple and efficient stochastic gradient descent (SGD) outlined in Algorithm 1. The ability to estimate the gradient will be important after fitting too, allowing us to generate samples from the density $\widehat{p_X^*}$.

Estimating $\nabla_\theta L(\theta) = \nabla_\theta L_{\text{reg}}(\theta) + \nabla_\theta L_1(\theta) + \nabla_\theta L_2(\theta)$: We start with the term $L_1(\theta) = -\mathbb{E}_{y\sim p_Y^1}[\phi_\theta(y)]$: we can directly differentiate under the integral and estimate the gradient with samples $y_1, \dots, y_N \overset{i.i.d.}{\sim} p_Y^1$:

$$\widehat{\nabla_\theta L_1}(\theta) = -\tfrac{1}{N}\textstyle\sum_{i=1}^N \nabla_\theta \phi_\theta(y_i) \approx \nabla_\theta L_1(\theta).$$

For $L_2(\theta) = \log \mathbb{E}_{x\sim p_X^0}\left[\exp\left(\mathbb{E}_{y\sim p_{Y|X}(\cdot|x)}[\phi_\theta(y)]\right)\right]$, we have the following easy identity (see Section B.5):

$$\nabla_\theta L_2(\theta) = \int_{\mathcal{X}}\int_{\mathcal{Y}} p_X(x;\theta) p_{Y|X}(y|x) \nabla_\theta \phi_\theta(y)\, dy\, dx. \quad (7)$$

Thus we can estimate $\nabla_\theta L_2(\theta)$ by first drawing $x_1, \dots, x_N \overset{\text{i.i.d.}}{\sim} p_X(\cdot;\theta)$, for each x_i, drawing $y_i' \sim p_{Y|X=x_i}$ and averaging: $\widehat{\nabla_\theta L_2}(\theta) = \frac{1}{N}\sum_{i=1}^N \nabla_\theta \phi_\theta(y_i')$.

The final term $L_{\text{reg}}(\theta)$ corresponds to the ε-relaxation of the marginal constraint; we note this disappears when $\varepsilon = 0$. For $\varepsilon > 0$, the following simple importance sampling approach, with p_Y^1 as proposal distribution works well:

$$L_{\text{reg}}(\theta) = \left(\int_{\mathcal{Y}} \phi_\theta^2(y)\, dy\right)^{\frac{1}{2}} = \left(\mathbb{E}_{y\sim p_Y^1(y)}\left[\frac{\phi_\theta^2(y)}{p_Y^1(y)}\right]\right)^{1/2}.$$

We choose $p_Y^1(y)$ as the proposal distribution since term $L_1(\theta)$ in Equation (5) suggests that $p_Y^1(y)$ will be close to $\phi_\theta(y)$ for θ in the neighborhood of θ^*. Then, we obtain a Monte Carlo estimate of the expectation using samples $y_1, \dots, y_N \overset{i.i.d.}{\sim} p_Y^1$, and take its square root to obtain $\widehat{L}_{\text{reg}}(\theta) := \left(\frac{1}{N}\sum_{i=1}^N \frac{\phi_\theta^2(y_i)}{p_Y^1(y)}\right)^{\frac{1}{2}}$. Then, from the chain rule:

$$\widehat{\nabla_\theta L_{\text{reg}}}(\theta) \approx \nabla_\theta \widehat{L}_{\text{reg}}(\theta) = \nabla_\theta \left(\tfrac{1}{N}\textstyle\sum_{i=1}^N \frac{\phi_\theta^2(y_i)}{p_Y^1(y)}\right)^{\frac{1}{2}}.$$

Note that while this is consistent, the square-root term makes this biased. However, since $\theta^{(t)}$ evolves gradually as we approach θ^*, we can control this using a weighted average across SGD iterations, thus avoiding the need for large Monte Carlo sample size N.

The only remaining question is how to sample from the intractable $p_X(\cdot;\theta)$ for any θ. This is needed to estimate $\nabla_\theta L_2(\theta)$, and having found the dual maximizer θ^*, is needed to simulate from the modified density $p_X(x;\theta^*)$.

Sampling $p_X(x;\theta)$ via SGLD. To sample from $p_X(x;\theta)$, a natural approach extends importance weighting/resampling from the deterministic constraint setting, and assigns weights $w_\theta(x)$ to samples from p_X^0 according to Equation (6). However, such a procedure requires estimating the weights (which is challenging), does not exploit gradient information and other information we have already calculated while optimizing the dual. More importantly, when p_X^0 and p_X^* have little overlap (a *density chasm* (Rhodes et al., 2020b)), this importance sampling scheme becomes extremely inefficient. Broadly, we cannot expect to explore p_X^* by reweighting samples from p_X^0, when the two do not overlap in some areas.

Instead, we take a Stochastic Gradient Langevin Dynamics (SGLD) approach (Welling & Teh, 2011). From (6) we have

$$\nabla_x \log p_X(x;\theta) = \nabla_x \log p_X^0(x) + \nabla_x \log w_\theta(x) - \cancel{\nabla_x \log C_\theta}.$$

In many situations, the score of the reference density $\nabla_x \log p_X^0(x)$ is available. This is the case in the examples we consider, where p_X^0 corresponds to a learned score-based model, is an analytically tractable nonparametric estimate of a data distribution, or can be efficiently obtained by automatic differentiation. For the second term above,

$$\begin{aligned}\nabla_x \log w_\theta(x) &= \nabla_x \int_{\mathcal{Y}} \phi_\theta(y) p_{Y|X}(y|x)\, dy \\ &= \mathbb{E}_{y\sim p_{Y|X=x}}\left[\phi_\theta(y)\cdot \nabla_x \log p_{Y|X}(y|x)\right],\end{aligned} \quad (8)$$

which admits an simple Monte Carlo estimator for any fixed $x \in \mathcal{X}$, provided (once again) $\nabla_x \log p_{Y|X}(y|x)$ is available either in closed-form or via automatic differentiation. Now we can sample from $p_X^1(x;\theta)$ via SGLD updates (Welling & Teh, 2011):

$$\begin{aligned}x_i \leftarrow x_{i-1} + \alpha\big(&\log p_X^0(x_{i-1}) \\ &+ \frac{1}{M}\sum_{j=1}^M \phi_\theta(y_j^i)\cdot \nabla_x \log p_{Y|X}(y_j^i|x_{i-1})\big) + \sqrt{2\alpha} z_i,\end{aligned} \quad (9)$$

for $i = 1, \dots, N$, $z_i \sim N(0, I)$, $y_j^i \overset{\text{i.i.d.}}{\sim} p_{Y|X}(\cdot|x_{i-1})$, M is the Monte Carlo batch size, and α is the step size.

Putting these three parts together gives an estimate of the gradient $\nabla_\theta L(\theta)$. Now, we can take a gradient step to update the neural network parameters θ. Specifically, we employ stochastic gradient descent (SGD) with a prescribed stepsize schedule $\{\eta_t\}_{t\geq 0}$, and update the parameters according to $\theta^{(t+1)} = \theta^{(t)} - \eta_t \nabla_\theta L(\theta^{(t)})$. We use Adam (Kingma, 2014) in our implementation.

In practice, we initialize $x_0 \sim p_X^0$, and maintain a persistent Markov chain for X during training. Recognizing that the distributions $p_X(x;\theta^{(t)})$ (before an SGD step) and $p_X(x;\theta^{(t+1)})$ (after the SGD step) are quite similar, the last state of each chain at iteration t (with parameter $\theta^{(t)}$) is used to initialize the chain at iteration $t+1$. At test time, the final training states are again used as initialization. Pseudocode is given in Algorithm 1, with more details in the Appendix.

Comments on our proposed algorithm: As stated earlier, the key advantage of our SGLD approach over naive importance resampling is its ability to handle situations where there is a *density chasm* (Rhodes et al., 2020b) between p_Y^1 and p_Y^0. This issue, corresponding to low overlap between

Algorithm 1 SGLD+SGD approach to optimize the dual

Training – Learn $\widehat{\theta}^*$ via SGD + SGLD
Initialize: $\theta^{(0)}$ and $x^{(0)} \sim p_X^0$
for $t = 1, \ldots, T$ **do**
  $\{x_i\}_{i=1}^N \leftarrow$ **SGLD**$(\theta^{(t-1)}, x^{(t-1)}, N)$ {equation 9}
  Calculate the stochastic gradient:
  $\{y_{1,i}\}_{i=1}^N, \{y_{\text{reg},i}\}_{i=1}^N \sim p_Y^1; \{y_{2,i}\}_{i=1}^N \sim p_{Y|X}(\cdot|x_i)$
  $\widehat{\nabla_\theta L_1^{(t)}} \leftarrow -\frac{1}{N}\sum_{i=1}^N \nabla_\theta \phi_{\theta^{(t-1)}}(y_{1,i})$
  $\widehat{\nabla_\theta L_2^{(t)}} \leftarrow \frac{1}{N}\sum_{i=1}^N \nabla_\theta \phi_{\theta^{(t-1)}}(y_{2,i})$
  $\widehat{\nabla_\theta L_{\text{reg}}^{(t)}} \leftarrow \varepsilon \nabla_\theta \left(\frac{1}{N}\sum_{i=1}^N \frac{\phi^2_{\theta^{(t)}}(y_{\text{reg},i})}{p_Y^1(y_{\text{reg},i})} \right)^{\frac{1}{2}}$
  $\widehat{\nabla_\theta L^{(t)}} \leftarrow \widehat{\nabla_\theta L_1^{(t)}} + \widehat{\nabla_\theta L_2^{(t)}} + \widehat{\nabla_\theta L_{\text{reg}}^{(t)}}$
  $\theta^{(t)} \leftarrow \text{SGD\_step}\left(\theta^{(t-1)}, \widehat{\nabla_\theta L^{(t)}}\right)$ {Update $\theta^{(t)}$}
  $x^{(t)} \leftarrow x_N$ {Update $x^{(t)}$}
end for

Simulation – Sample from $\widehat{p}_X^* \equiv p_X(x; \widehat{\theta}^*)$ via SGLD
$\{x_j\}_{j=1}^n \leftarrow$ **SGLD**$(\widehat{\theta}^*, x^{(T)}, n)$
Output: $\{x_j\}_{j=1}^n$

two distributions is a well-known problem with density ratio estimation, where most proposals from p_X^0 will have low weight; introducing low effective sample sizes and large variance. Our SGLD approach can instead explore the support of p_X^1 without repeatedly returning to p_X^0 as a proposal distribution, and instead requires the score function to be accurately estimated over high-probability areas of p_Y^1. While this seems challenging, recall that we learn this score over the dual optimization procedure. Here, we usually initialize $\nabla_x \log p_X^1(x; \theta^{(0)}) \approx \nabla_x \log p_X^0(x)$, and this is gradually 'annealed' to p_X^1 over training. The result is that, for θ near θ^*, combining unbiased gradient estimation in the dual with SGLD-based sampling in the primal is significantly more robust to chasms than purely importance-based schemes. We empirically validate this behavior in Section 4.

3.2. Related work

Existing work imposing marginal constraints has mostly focused on deterministic mappings from X to Y, and with $\varepsilon = 0$. An early work, focusing on Dirichlet process mixture models is that of Kessler et al. (2015). Their solution involved a density ratio estimate (to correct a Metropolis-Hastings MCMC acceptance probability), and thus is prone to the density-chasm problem. Both Schifeling & Reiter (2016) and Tang & Rao (2025) studied a special instance of the above problem, where the function f projects X onto a subset of its components (so that p_Y^0 is literally a marginal distribution of p_X^0). Schifeling & Reiter (2016) approximately enforce such marginal constraints by incorporating a synthetic dataset from the constraint distribution (while also working with restricted reference models where marginal estimation is easy). Tang & Rao (2025) enforced the marginal constraints through a conditional sampling approach (involving nonparametric conditional density modeling). Dai et al. (2022), worked with a similar setup as ours, even citing Lemma 11 of (Altun & Smola, 2006). However, they too worked with deterministic transformations, and additionally restricted themselves to settings with discrete set-data X, and binary attributes Y, allowing much simpler (but restricted) approach. A related paper is Kim et al. (2024), who sought to address dataset bias in diffusion models using a smaller unbiased reference dataset (rather than a target marginal). There are settings when this is useful, though our approach of controlling marginal distributions is more flexible, applicable beyond diffusion models, and allows modelers to guide models as they desire without an auxiliary dataset. Kim et al. (2024) tried to mitigate the density-chasm issue for diffusion-models by doing this along the diffusion models 'time'-axis. In the same spirit, we extend our constrained problem over time when specializing to diffusion models. There have been other attempts to incorporate domain knowledge into deep neural models through the design of specialized generative processes or model architectures (Andreas et al., 2016; Hu et al., 2017; Chen et al., 2016), such approaches are typically limited in both model expressiveness and the forms of knowledge they can accommodate.

More abstractly, our work is closely related to problems from optimal transport (specifically, the entropy-regularized optimal transport problem (Cuturi, 2013; Benamou et al., 2015)), as well the Schrödinger bridge problem (Schrödinger, 1932; Marino & Gerolin, 2020), both of which are instances of marginally constrained KL divergence minimization. Both of these problems involve *two* marginal constraints, and seek to find (respectively) a joint distribution that minimizes an expected cost (this cost can be viewed as a reference distribution) or a diffusion model that minimizes distance from a reference diffusion. Our problem can be viewed as a one-marginal version of these problems, but with added complexity introduced through the flexibility of $p(Y|X)$. One can try to adapt an IPF or Sinkhorn style algorithm from this work to our problem. This will typically involve discretization/gridding of $\mathcal{X}$, with the convergence of the resulting algorithm hard to manage (Wilkins-Reeves et al., 2021).

4. Experiments

For synthetic and NPMLE tasks below, we parameterize $\phi : \mathcal{Y} \to \mathbb{R}$ using a two-hidden-layer MLP with sizes (32, 16) and ReLU activations. Monte Carlo batch

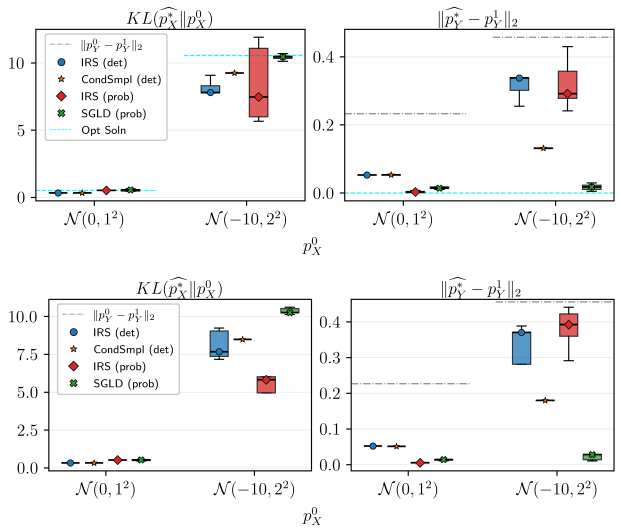


Figure 1. Synthetic results: (top) Gaussian (bottom) Laplace noise

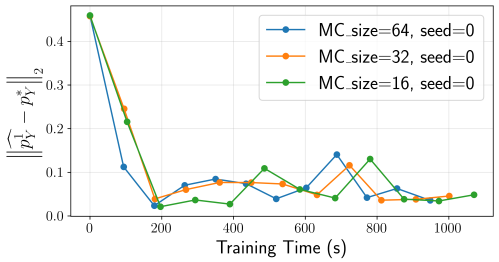


Figure 2. Constraint gap vs epoch for different SGLD batch sizes.

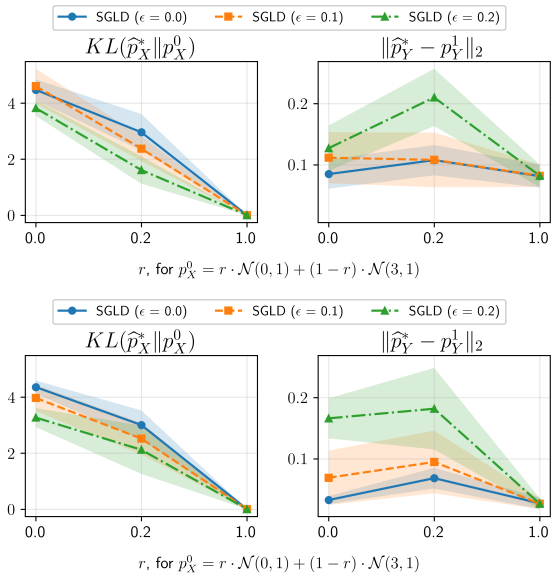


Figure 3. NPMLE results for dataset sizes 100 (top) and 100,000 (bottom). Increasing r along the x-axis **decreases** the density chasm between p_X^0 and p_Y^1.

sizes are chosen from $\{16, 32, 64\}$ and learning rates from $\{10^{-5}, 10^{-4}, 10^{-3}\}$. For image tasks, we use a time-dependent MLP $\phi : [0, T] \to \mathbb{R}^{|\mathcal{Y}|}$, with three hidden layers of size (32, 32, 32) and the same hyperparameter ranges. For more details on training and hyperparameter tuning, please refer to Section C.

4.1. Synthetic Tasks

We start with a simple synthetic setup: the reference p_X^0 is Gaussian, and Y is a linear transformation of X plus additive noise (we consider both Gaussian and Laplace noise here). For clarity, we present results for the 1-d setting with $f(x) = 2x + 1$, $\sigma_{Y|X} = 1$ and $p_Y^1(y) = \mathcal{N}(-1, 2^2)$ (with more results in the Appendix). We chose this setup so that the problem is feasible, with solution $p_X^* = \mathcal{N}\left(-1, \left(\frac{3}{4}\right)^2\right)$ when p_Y^1, p_X^0 and $p_{Y|X}$ are Gaussian (the solution does not depend on p_X^0, see Section E.1). We vary $p_X^0 \in \{\mathcal{N}(0, 1^2), \mathcal{N}(-10, 2^2)\}$, the latter inducing a severe density chasm between p_X^0 and p_X^*.

We evaluated any candidate solution $\widehat{p_X^*}$ along the following metrics: (i) *constraint gap*, $\|\widehat{p_Y^*} - p_Y^1\|_2$ (for $\varepsilon = 0$, smaller is better; for $\varepsilon > 0$, we check how closely $\|\widehat{p_Y^*} - p_Y^1\|_2 \leq \varepsilon$ is satisfied); (ii) *distributional shift*, $\mathrm{KL}(\widehat{p_X^*}\|p_X^0)$ (for the same constraint gap, smaller is better).

Figure 1 summarizes the results. Across all settings, all methods reduce the constraint gap $\|p_Y^0 - p_Y^1\|_2$ associated with the reference model p_X^0. When no density chasm is present, methods designed for probabilistic constraints consistently achieve smaller constraint gaps than those for deterministic transformations. Additionally, with a density chasm, importance-resampling-based approaches become ineffective due to the fixed misspecified proposal distribution. Our proposed SGLD approach remains effective in this regime, successfully reducing the constraint gap across all scenarios. This story is repeated in synthetic experiments with more complex densities and higher dimension, we include these in the appendix.

In Figure 2, we plot the evolution of the constraint gap over SGD training for different sizes of SGLD batch size. Our results are fairly stable, showing estimating gradients with batch sizes as small as 16 is effective.

4.2. Regularized NPMLE

Next, we consider the nonparametric MLE task (Wilkins-Reeves et al., 2021) outlined in Section 1. Recall we are given a dataset of i.i.d. observations $\{y_i\}_{i=1}^N$, obtained by perturbing another i.i.d. dataset $\{x_i\}_{i=1}^N$ through a known conditional distribution $p_{Y|X}$. Write p_Y^1 for the smoothed empirical density of the observations (we use a kernel density estimate), p_X^T for the true (unknown) data generation

density, and p_X^0 for the 'direction' towards which we wish to regularize our estimate p_X^* of p_X^T. p_X^* will then be obtained as the solution of our constrained optimization problem.

Our simulations use the following specifications: $p_X^0 = \mathcal{N}(0,1)$, $p_{Y|X}(\cdot|x) = \mathcal{N}(x, 0.2^2)$, $p_Y^1 = \text{KDE}\left(\{y_i\}_{i=1}^N\right)$, $x_i \overset{i.i.d.}{\sim} r\mathcal{N}(0,1) + (1-r)\mathcal{N}(3,1), y_i \sim p_{Y|X}(\cdot|x_i)$ for $r \in \{0, 0.2, 1\}$. Thus, p_X^T is a two-component Gaussian mixture model, with one component equal to p_X^0. The mixture weight controls the deviation from the prior, or equivalently, the density chasm.

We consider our problem with the relaxed constraints, setting $\varepsilon \in \{0, 0.1, 0.2\}$. In practice, the choice of this will be determined by the sample size N, with larger N requiring stronger constraints (as we seek to penalize deviations from the empirical estimate more heavily). Figure 3 shows results when the sample size $N = 100$ and $100,000$. We see that in most cases, the constraint is satisfied, with the returned solution close to the constraint boundary. In other words, as we would hope, our approach deviated from the target density to the maximum extent permitted, in order to minimize the KL divergence from the reference. Additionally, enforcing the constraint more rigidly results in solutions further away from the reference density.

There are two exceptions to this. The first is for small r and $\varepsilon = 0$ (the hardest, setting). Now, the constraint is violated, a consequence of the problem being infeasible. However, the solution obtained by terminating our dual optimization scheme is still a reasonable one, having L_2 distance of about 0.1 from the target marginal (instead of the impossible 0). An implication of this is that the feasibility threshold ε_* is approximately 0.1.

The second exception occurs when $r = 1$: now the constraint is inactive, with even settings with larger ε returning solutions closer to the target density. This however is a consequence of the fact that the reference distribution p_X^0 is already very close to the ground-truth solution p_X^*, so that the constraint is already satisfied. We see this effect even more strongly with the larger sample size, when the KDE estimate is closer to the true marginal. Since $\|p_Y^0 - p_Y^1\|_2$ is already within the tolerance ε before any score correction, all methods are driven primarily to minimize the KL divergence, converging to nearly identical results.

4.3. Image Generation with Fairness constraints

For this task, p_X^0 corresponds to a pre-trained score-based diffusion model associated (Song et al., 2021; Karras et al., 2022) with 'time'-dependent parameterized score function $s_\eta(\boldsymbol{x}_t, t)$ (with $\nabla_{\boldsymbol{x}_0} \log p_X^0(\boldsymbol{x}_0) = s_\eta(\boldsymbol{x}_0, 0)$). This, together with an attribute classifier $p_{Y|X}(y|x)$ implies a distribution over classes p_Y^0. We consider two target marginal distributions p_Y^1: (i) **Balanced distribution**: the standard

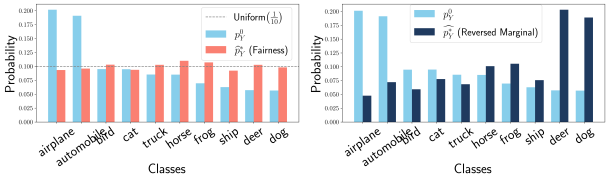


Figure 4. Class Distribution Before and After Training.

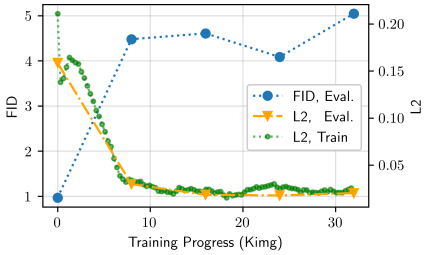


Figure 5. Training dynamics on CIFAR-10 (LT/5%). The training L_2 decreases smoothly and exhibits stable convergence. We track a moving average of L_2 (window size: 5K images). The training L_2 distance reaches its minimum at approximately 18K images.

setting in the literature (Choi et al., 2020; Kim et al., 2024), where the target marginal p_Y^1 is uniform over classes, and (ii) **Reversed distribution**: where p_Y^1 is constructed by reversing the rank order (by frequency) of classes in p_Y^0. We emphasize that our framework treats p_X^0 and $p_{Y|X}$ as given objects, as specified in Section 2. Their training in our experiments serves only to instantiate a concrete setting and is not part of the proposed methodology.

We extend our methodology to exploit the additional gradient structure provided by diffusion models. Now, we apply our marginal constraint to the diffusion model densities at all times (recall $t = 1$ corresponds to noise that the diffusion model converts to images at $t = 0$). We allow the constraint $\varepsilon(t)$ to vary with diffusion time (so that it is applied more strongly from $t = 1$ to $t = 0$ as the diffusion model converts noise to signal), with $\varepsilon(0)$ equal to the desired constraint parameter ε. This gives the time-weighted dual $L(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,T]}[\lambda(t)L(\theta,t)]$ where

$$L(\theta,t) := \Big\{\varepsilon(t)\left(\int_{\mathcal{Y}} \phi_\theta^2(y,t)\,dy\right)^{\frac{1}{2}} - \int_{\mathcal{Y}} \phi_\theta(y,t)p_Y^1(y)\,dy$$
$$+ \log \int_{\mathcal{X}} p_{X(t)}^0(\boldsymbol{x}_t,t)\cdot \exp\left(\int_{\mathcal{Y}} \phi_\theta(y)p_{Y|X(t)}(y|\boldsymbol{x}_t)\,dy\right) d\boldsymbol{x}_t\Big\}$$
$$:= L_{\text{reg}}(\theta,t) + L_1(\theta,t) + L_2(\theta,t), \qquad (10)$$

and $\lambda(t)$ is the temporal weighting function of the diffusion model. The optimization scheme described in Section 3.1 remains mostly unchanged, with the diffusion sampler having the adapted score function $s_\eta(\boldsymbol{x}_t,t) + w_\theta(\boldsymbol{x}_t,t)$. For more details, please refer to Section D.1.

	Task	$\mathrm{FID}(p_{\bar{\mathcal{D}}_{\mathrm{ref}}}, p^0_X)$	$\mathrm{FID}(p^0_X, p^0_X)$	$\mathrm{FID}(\widehat{p^*_X}, p^0_X)$	$\lVert p_{\bar{\mathcal{D}}_{\mathrm{ref}}} - p^1_Y \rVert$	$\lVert p^0_Y - p^1_Y \rVert$	$\lVert \widehat{p^*_Y} - p^1_Y \rVert$
CelebA	Balanced	2.799[a]	0.460	0.961	0	0.219	0.028
CIFAR10 (LT, 5%)	Balanced	13.791[b]	0.967	4.609	0	0.158	0.018
CIFAR10 (LT, 5%)	Reversed	13.791	0.967	13.174	0.158	0.279	0.043

[a] $\bar{\mathcal{D}}_{\mathrm{ref}}$ is constructed via class-wise random sampling following (Kim et al., 2024). [b] $\bar{\mathcal{D}}_{\mathrm{ref}}$ is obtained from GitHub; value 12.99 is reported in (Kim et al., 2024).

Table 1. Results on constrained image generation. Score adaptation substantially reduces the marginal discrepancy $\|\widehat{p^*_Y} - p^1_Y\|_2$ while maintaining proximity to the pretrained generator p^0_X, as reflected by $\mathrm{FID}(\widehat{p^*_X}, p^0_X)$. In contrast, the unbiased reference distribution $\bar{\mathcal{D}}_{\mathrm{ref}}$ (Kim et al., 2024) lies far from p^0_X. $\mathrm{FID}(p^0_X, p^0_X)$ is estimated via resampling, reflecting the intrinsic stochasticity (expected ≈ 0).

Datasets We use two standard benchmark datasets: CIFAR-10 (Krizhevsky et al., 2009) and CelebA (Liu et al., 2015). For CIFAR-10, we adopt the long-tailed (LT) biased dataset constructed in (Kim et al., 2024), using the most challenging LT/5% setting with maximal bias contamination (Cao et al., 2019). For CelebA, we use the original dataset and consider bias induced by combination attributes of hair color and gender, following common practice in fair image generation (Choi et al., 2020; Kim et al., 2024).

Setup For each dataset, we pre-train unconditional diffusion models from scratch using the EDM framework (Karras et al., 2022), yielding a time-dependent score function $s_\eta(\boldsymbol{x}_t, t) = \nabla_{\boldsymbol{x}_t} \log p^0_{X(t)}(\boldsymbol{x}_t)$. As expected, the bias in the training data is inherited by the generated samples (see Figure 4). To obtain the time-dependent conditional distribution $p_{Y|X(t)}$, we train classifier networks on the full datasets following the time-dependent discriminator training scheme in Kim et al. (2024; 2022). We then optimize our proposed time-dependent dual objective to learn ϕ_θ. In our case Y is discrete and so $\phi_\theta(\cdot, t) : \mathcal{T} \to \mathbb{R}^{|\mathcal{Y}|}$ reduces to 10 lightweight multi-layer perceptron with time as the only input. More implementation details are in Section C.

KL divergence is hard to estimate in this setting, and instead, as is standard (Heusel et al., 2017), we report Fréchet Inception Distance (FID) between generated samples and those from the original biased model p^0_X. Constraint satisfaction is measured by $\|\widehat{p^*_Y} - p^1_Y\|_2$, the L_2 distance between the empirical marginal of generated labels and the target p^1_Y.

Results Table 1 reports quantitative results. Across both datasets, our method substantially reduces the marginal discrepancy $\|\widehat{p^*_Y} - p^1_Y\|_2$ while maintaining image quality, as measured by FID relative to the original biased model. For CIFAR-10 (LT/5%), we achieve near-uniform class distributions in the balanced task and accurately match the reversed target distribution, confirming the flexibility of our framework beyond fairness. Figure 4 further illustrates that the adapted samples closely follow the desired class marginals.

As a baseline, we report the performance of the *entire unbiased dataset* $\bar{\mathcal{D}}_{\mathrm{ref}}$ introduced in (Kim et al., 2024, Table 5), an oracle target in importance-reweighting-based adaptation methods (Choi et al., 2020; Kim et al., 2024). The large values of $\mathrm{FID}(\bar{\mathcal{D}}_{\mathrm{ref}}, p^0_X)$ illustrate that satisfying the marginal constraint is relatively easy though simultaneously keep low

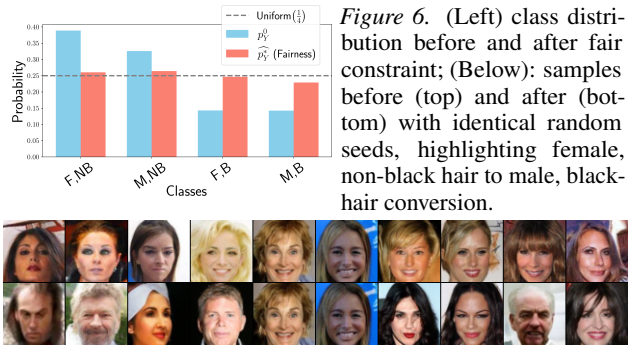


Figure 6. (Left) class distribution before and after fair constraint; (Below): samples before (top) and after (bottom) with identical random seeds, highlighting female, non-black hair to male, black-hair conversion.

deviation from a pre-trained generative model is not trivial. For more experimental details, see Section E.

5. Conclusion

In this work, we study a broad problem of incorporating side knowledge into generative models. Our side knowledge takes the form of probability distributions over stochastic transformations of the objects being modeled. We show that, while appropriate for deterministic transformations, naive importance reweighting or conditional sampling is not appropriate for our general setting. Instead, we propose to optimize a dual objective, and derive an SGD-SGLD scheme to learn a score-correcting neural network.

Several open issues remain. One involves extending our SGLD framework to more complex sample spaces, whether infinite-dimensional spaces corresponding to stochastic process models, combinatorial spaces (e.g., trees and random sets) and structured manifolds (e.g., protein structures and trajectories). In this work, we consider only a single marginal constraint. In practical settings, users may possess prior information about multiple aspects of the modeled objects, themselves not fully consistent. Our current analysis focuses on KL distance from the reference density, and the L_2 norm for relaxed marginal constraints. Exploring alternative divergence measures and norms may lead to dual loss gradients that are easier to estimate, or better suited to specific applications.

Impact Statement

This work aims to address some ethical concerns aris-

ing from complex black-box generative models. By enabling calibration of generative models with respect to user-specified probabilistic constraints, the proposed approach can help mitigate unintended or undesirable biases arising from data or model misspecification in real-world applications. Admittedly, such tools can also be misused by malicious agents, to explicitly steer generative models towards undesirable outcomes.

References

Altun, Y. and Smola, A. Unifying divergence minimization and statistical inference via convex duality. In *International Conference on Computational Learning Theory*, pp. 139–153. Springer, 2006.

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

Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*, 2016.

Barocas, S., Hardt, M., and Narayanan, A. *Fairness and machine learning: Limitations and opportunities*. MIT press, 2023.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

Choi, K., Grover, A., Singh, T., Shu, R., and Ermon, S. Fair generative modeling via weak supervision. In *International Conference on Machine Learning*, pp. 1887–1898. PMLR, 2020.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Dai, H., Yang, M., Xue, Y., Schuurmans, D., and Dai, B. Marginal distribution adaptation for discrete sets via module-oriented divergence minimization. In *International Conference on Machine Learning*, pp. 4605–4617. PMLR, 2022.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. *Bayesian Data Analysis*. Chapman & Hall/CRC Texts in Statistical Science Series. CRC, third edition, 2013.

Ghosal, S. and Van der Vaart, A. W. *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. Toward controlled generation of text. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2017.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577, 2022.

Kessler, D. C., Hoff, P. D., and Dunson, D. B. Marginally specified priors for non-parametric bayesian estimation. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 77(1):35–58, 2015.

Kim, D., Kim, Y., Kang, W., and Moon, I.-C. Refining generative process with discriminator guidance in score-based diffusion models. *arXiv preprint arXiv:2211.17091*, 2022.

Kim, Y., Na, B., Park, M., Jang, J., Kim, D., Kang, W., and Moon, I.-C. Training unbiased diffusion models from biased dataset. In *The Twelfth International Conference on Learning Representations*, 2024.

Kingma, D. P. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Laird, N. Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association*, 73(364):805–811, 1978.

Li, B., Qi, P., Liu, B., Di, S., Liu, J., Pei, J., Yi, J., and Zhou, B. Trustworthy ai: From principles to practices. *ACM Computing Surveys*, 55(9):1–46, 2023.

Lindsay, B. G. Nonparametric maximum likelihood. In *Mixture Models*, volume 5, pp. 108–127. Institute of Mathematical Statistics, 1995.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Marino, S. D. and Gerolin, A. An optimal transport approach for the schrödinger bridge problem and convergence of sinkhorn algorithm. *Journal of Scientific Computing*, 85(2):27, 2020.

Rhodes, B., Xu, K., and Gutmann, M. U. Telescoping density-ratio estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020a. Curran Associates Inc. ISBN 9781713829546.

Rhodes, B., Xu, K., and Gutmann, M. U. Telescoping density-ratio estimation. *Advances in neural information processing systems*, 33:4905–4916, 2020b.

Schifeling, T. A. and Reiter, J. P. Incorporating marginal prior information in latent class models. *Bayesian Analysis*, 11(2):499–518, 2016.

Schrödinger, E. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pp. 269–310, 1932.

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

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

Tang, B. and Rao, V. Marginally constrained nonparametric bayesian inference through gaussian processes. *Journal of Statistical Planning and Inference*, 237:106261, 2025.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Wilkins-Reeves, S., Chen, Y.-C., and Chan, K. C. G. Data harmonization via regularized nonparametric mixing distribution estimation. *Annals of Applied Statistics*, 2021.

Supplementary Material for
Generative modeling with probabilistic constraints

A. Infeasibility under stochastic transformations

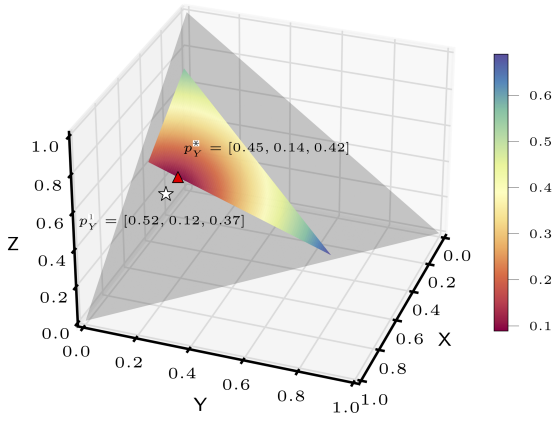


Figure 7. Solutions may fail to exist in the stochastic transformation setting when ε is small. For a given target marginal $p_Y^1 \in \mathcal{P}_3$ (shown as a star), there exists a stochastic transition matrix $P_{Y|X} \in \mathbb{R}^{3\times 3}$ such that $\inf_{p_X \in \mathcal{P}_3} \left\| p_Y^1 - P_{Y|X} \circ p_X \right\|_2 > 0$, where $\mathcal{P}_3$ denotes the three-dimensional probability simplex. The grey region represents $\mathcal{P}_3$, while the colored surface corresponds to $\{P_{Y|X} \circ p_X : p_X \in \mathcal{P}_3\}$, with the colormap indicating the associated L_2 distance to p_Y^1. In particular, the red triangle marks the point p_Y^* that empirically minimizes the L_2 distance via grid search.

B. Proofs

B.1. Derivation of Corollary 3.2 via Jensen's Inequality

Let $(\mathcal{X}, \mathcal{A})$ and $(\mathcal{Y}, \mathcal{B})$ be measurable spaces. Let p_X^0 be a probability measure on $(\mathcal{X}, \mathcal{A})$. It induces a probability measure p_Y^0 on $(\mathcal{Y}, \mathcal{B})$ via

$$p_Y^0(B) = p_X^0\big(\{x : f(x) \in B\}\big), \qquad B \in \mathcal{B}.$$

Assume $p_Y^1 \ll p_Y^0$. We consider the optimization problem

$$\begin{aligned} p_X^* = \underset{p_X \in \mathcal{P}_\mathcal{X}}{\arg\min} \quad & \mathrm{D}_{\mathrm{KL}}(p_X \,||\, p_X^0) \\ \text{subject to} \quad & \|p_Y - p_Y^1\|_2 \le \varepsilon. \end{aligned}$$

where $\mathcal{P}_\mathcal{X}$ denotes the set of probability measures on $(\mathcal{X}, \mathcal{A})$, and p_Y is the pushforward of p_X under f, i.e.,

$$p_Y(B) = p_X\big(\{x : f(x) \in B\}\big), \qquad B \in \mathcal{B}.$$

The solution is derived as below.

$$\begin{aligned} & \min_{p_X \in \mathcal{P}_\mathcal{X}} \int_\mathcal{X} \log\left(\frac{dp_X}{dp_X^0}(x)\right) p_X(dx) \\ & \text{subject to} \ \ \|p_Y - p_Y^1\|_2 \le \varepsilon \\ & = \min_{p_X \in \mathcal{P}_\mathcal{X}} \int_\mathcal{X} \log\left(\frac{dp_{XY}}{dp_{XY}^0}(x, f(x))\right) p_X(dx) \\ & \text{subject to} \ \ \|p_Y - p_Y^1\|_2 \le \varepsilon \end{aligned}$$

$$
\begin{aligned}
&= \min_{p_X \in \mathcal{P}_\mathcal{X}} \int_\mathcal{X} \log\left(\frac{dp_{X|Y}(\cdot|f(x))}{dp^0_{X|Y}(\cdot|f(x))}(x)\frac{dp_Y}{dp^0_Y}(f(x))\right) p_X(dx) \\
&\text{subject to } \|p_Y - p^1_Y\|_2 \le \varepsilon \\
&= \min_{p_X \in \mathcal{P}_\mathcal{X}} \left[\int_\mathcal{X} \log\left(\frac{dp_{X|Y}(\cdot|f(x))}{dp^0_{X|Y}(\cdot|f(x))}(x)\right) p_X(dx) + \int_\mathcal{X} \log\left(\frac{dp_Y}{dp^0_Y}(f(x))\right) p_X(dx)\right] \\
&\text{subject to } \|p_Y - p^1_Y\|_2 \le \varepsilon \\
&= \min_{p_X \in \mathcal{P}_\mathcal{X}} \left[\int_\mathcal{Y}\int_{f^{-1}(y)} \log\left(\frac{dp_{X|Y}(\cdot|y)}{dp^0_{X|Y}(\cdot|y)}(x)\right) p_{X|Y}(dx|y)\, p_Y(dy) + \int_\mathcal{Y} \log\left(\frac{dp_Y}{dp^0_Y}(y)\right) p_Y(dy)\right] \\
&\text{subject to } \|p_Y - p^1_Y\|_2 \le \varepsilon \\
&= \min_{p_X \in \mathcal{P}_\mathcal{X}} \left[\int_\mathcal{Y}\int_{f^{-1}(y)} \log\left(\frac{dp_{X|Y}(\cdot|y)}{dp^0_{X|Y}(\cdot|y)}(x)\right) p_{X|Y}(dx|y)\, p_Y(dy) + \int_\mathcal{Y} \log\left(\frac{dp_Y}{dp^0_Y}(y)\right) p_Y(dy)\right] \\
&\text{subject to } \|p_Y - p^1_Y\|_2 \le \varepsilon \\
&= \min_{p_X \in \mathcal{P}_\mathcal{X}} \left[\int_\mathcal{Y} p_Y(dy)\left(-\int_{f^{-1}(y)} \log\left(\frac{dp^0_{X|Y}(\cdot|y)}{dp_{X|Y}(\cdot|y)}(x)\right) p_{X|Y}(dx|y)\right) + \int_\mathcal{Y} \log\left(\frac{dp_Y}{dp^0_Y}(y)\right) p_Y(dy)\right] \\
&\text{subject to } \|p_Y - p^1_Y\|_2 \le \varepsilon \\
&\ge \min_{p_X \in \mathcal{P}_\mathcal{X}} \left[\int_\mathcal{Y} p_Y(dy)\left(-\log\int_{f^{-1}(y)} \frac{dp^0_{X|Y}(\cdot|y)}{dp_{X|Y}(\cdot|y)}(x)\, p_{X|Y}(dx|y)\right) + \int_\mathcal{Y} \log\left(\frac{dp_Y}{dp^0_Y}(y)\right) p_Y(dy)\right] \\
&\text{subject to } \|p_Y - p^1_Y\|_2 \le \varepsilon \\
&= \min_{\|p_Y - p^1_Y\|_2 \le \varepsilon} \int_\mathcal{Y} \log\left(\frac{dp_Y}{dp^0_Y}(y)\right) p_Y(dy).
\end{aligned}
$$

The second-to-last line is obtained by applying Jensen's inequality to the conditional measures $p_{X|Y}(\cdot \mid y)$. Equality holds if and only if

$$p_{X|Y}(\cdot \mid y) = p^0_{X|Y}(\cdot \mid y) \quad \text{for } p_Y\text{-almost every } y.$$

Therefore, the minimizer p^*_X admits the closed-form expression

$$
\begin{aligned}
p^*_X(A) &= \int_\mathcal{Y} p^0_{X|Y}(A \mid y)\, \widetilde{p}^\varepsilon_Y(dy), \qquad A \in \mathcal{A}, \\
&= \int_A \frac{d\widetilde{p}^\varepsilon_Y}{dp^0_Y}\big(f(x)\big)\, p^0_X(dx).
\end{aligned}
$$

where

$$\widetilde{p}^\varepsilon_Y = \operatorname*{arg\,min}_{\|p_Y - p^1_Y\|_2 \le \varepsilon} D_{\mathrm{KL}}(p_Y \| p^0_Y). \tag{11}$$

when $\varepsilon = 0$, we obtain $\widetilde{p}^\varepsilon_Y = p^1_Y$.

Remark. During the above derivation for the deterministic transformation case, we effectively optimize separately over the conditional distribution $p_{X|Y}$ and the marginal distribution p_Y. This separation is valid because the deterministic constraint $Y = f(X)$ is implicitly enforced when optimizing over $p_{X|Y}$; once $p_{X|Y}$ is supported on the manifold $f(x) = y$, the transformation structure is automatically satisfied. However, this argument does not extend to the stochastic transformation case. When the transformation is specified by a fixed conditional distribution $p_{Y|X}$, optimizing $p_{X|Y}$ and p_Y independently does not, in general, guarantee consistency with the prescribed stochastic mapping. In particular, the constraint $p_{Y|X}$ cannot be enforced by separate optimization over $p_{X|Y}$ and p_Y.

B.2. Derivation of Corollary 3.2 from Lemma 3.1.

For both deterministic transformation cases $Y = f(X)$ (i.e., $p_{Y|X} = \delta_{y=f(x)}$ and $\varepsilon \geq 0$), Equation (4) reduces to

$$p_X^*(x) = \frac{p_X^0(x) \exp\left(\phi^*(f(x))\right)}{\int_{\mathcal{X}} p_X^0(x) \exp\left(\phi^*(f(x))\right) dx},$$

we obtain that its induced distribution on $\mathcal{Y}$ is

$$\tilde{p}_Y^{\epsilon}(y) = \frac{p_Y^0(y) \exp\left(\phi^*(y)\right)}{\int_{\mathcal{X}} p_X^0(x) \exp\left(\phi^*(f(x))\right) dx}, \tag{12}$$

$$\begin{aligned} p_{X|Y}^*(x|y) &\coloneqq \frac{p_X^*(x) p_{Y|X}(y|x)}{\tilde{p}_Y^{\epsilon}(y)} \\ &= \frac{p_{XY}^0(x,y) \exp\left(\phi^*(f(x))\right)}{p_Y^0(y) \exp\left(\phi^*(y)\right)} \\ &= p_{X|Y}^0(x|y). \end{aligned} \tag{13}$$

To clarify, Equation (11) and Equation (12) are equivalent. This equivalence can be established either via the dual formulation of Equation (11) or by using Equation (13).

B.3. Proof of Corollary 3.3

As p_X^* is the marginal of the joint $p_X^* \; p_{Y|X}$, we define $w(x,y)$ as below.

$$\begin{aligned} w(x,y) &\coloneqq \frac{p_X^*(x)\, p_{Y|X}(y \mid x)}{p_Y^1(y)\, p_{X|Y}^0(x \mid y)} \\ &= \frac{p_X^*(x)\, p_{Y|X}(y \mid x)\, p_X^0(x)}{p_Y^1(y)\, p_{X|Y}^0(x \mid y)\, p_Y^0(y)} \cdot \frac{p_Y^0(y)}{p_X^0(x)} \\ &= \frac{p_X^*(x)\, p_{XY}^0(x,y)}{p_Y^1(y)\, p_{XY}^0(x,y)} \cdot \frac{p_Y^0(y)}{p_X^0(x)} \\ &= \frac{p_X^*(x)\, p_Y^0(y)}{p_X^0(x)\, p_Y^*(y)}. \end{aligned}$$

B.4. Proof of Proposition 3.4

Proposition 3.4 follows immediately from the following Lemma B.1.

Lemma B.1. *For the general problem, based on the definition of p_X^{det}, we write it as*

$$p_X^{det} = \int_{\mathcal{Y}} p_{X|Y}^0(x|y') p_Y^1(y') dy'.$$

The Y-marginal correponding to p_X^{det} could be written as:

$$p_Y^{det} = \int_{\mathcal{X}} p_{Y|X}(y|x) p_X^{det}(x).$$

We defined the operator T on a distribution on $\mathcal{Y}$ as

$$(Tq)(y) \coloneqq \int q^+(y) \coloneqq \int q(y) K(y|y') dy', \quad K(y|y') = \int p_{Y|X}(y|x) p_{X|Y}^0(x|y') dx,$$

and a corresponding operator as

$$D \coloneqq T - I.$$

It is easy to verify that $p_Y^{det} = T p_Y^1$.

Let

$$\mathcal{R} := \Big\{ q : q(y) = \int r(x) p_{Y|X}(y|x)\, dx \text{ for some density } r \text{ defined on } \mathcal{X} \Big\}.$$

Notice that this construction ensures that the feasibility of our optimization problem $\forall q \in \mathcal{R}$.

We make a non-degeneracy assumption that,

$$\exists\, q_0 \in \mathcal{R} \quad \text{such that} \quad Dq_0 \neq 0.$$

Under this assumption, we conclude that there exists a density $p_Y^1 \in \mathcal{R}$ *and a constant* $\alpha > 0$ *such that*

$$\|T p_Y^1 - p_Y^1\|_2 \;\geq\; \alpha \, \|p_Y^1 - p_Y^0\|_2.$$

Proof. By assumption, there exists $q_0 \in \mathcal{R}$ such that $Dq_0 \neq 0$. For $t \in [0,1]$, define $q_t := (1-t)p_Y^0 + tq_0$. Since $\mathcal{R}$ is convex and contains both p_Y^0 and q_0, we have $q_t \in \mathcal{R}$ for all t.

We compute the l_2 norm distance between q_t and p_y^0 as below.

$$\|q_t - p_Y^0\|_2 = \|(1-t)p_Y^0 + tq_0 - p_Y^0\|_2 = \|t(q_0 - p_Y^0)\|_2 = t\|q_0 - p_Y^0\|_2. \tag{14}$$

Using linearity of T, we obtain

$$\begin{aligned} Tq_t - q_t &= T\big((1-t)p_Y^0 + tq_0\big) - \big((1-t)p_Y^0 + tq_0\big) \\ &= (1-t)Tp_Y^0 + tTq_0 - (1-t)p_Y^0 - tq_0 \\ &= (1-t)\big(Tp_Y^0 - p_Y^0\big) + t\big(Tq_0 - q_0\big) \\ &= (1-t)Dp_Y^0 + tDq_0. \end{aligned}$$

Let

$$a := Dp_Y^0, \qquad b := Dq_0.$$

Then

$$Tq_t - q_t = (1-t)a + tb.$$

Using triangular inequality bound, we have $\|u+v\| \geq |\|u\| - \|v\||$ with $u = (1-t)a$ and $v = tb$,

$$\|Tq_t - q_t\|_2 \geq \big|\, (1-t)\|a\|_2 - t\|b\|_2 \big|.$$

Choose a $t \in (0,1]$ such that

$$t\|b\|_2 \geq 2(1-t)\|a\|_2.$$

This is possible since $b \neq 0$. Solving the above inequality gives

$$t \geq \frac{2\|a\|_2}{\|b\|_2 + 2\|a\|_2}.$$

From the chosen condition,

$$(1-t)\|a\|_2 \leq \frac{1}{2} t \|b\|_2.$$

Hence,

$$\|Tq_t - q_t\|_2 \geq t\|b\|_2 - (1-t)\|a\|_2$$

$$\geq t\|b\|_2 - \frac{1}{2}t\|b\|_2$$
$$= \frac{1}{2}t\|b\|_2. \tag{15}$$

According to Equation (14), we have

$$\|q_t - p_Y^0\|_2 = t\|q_0 - p_Y^0\|_2.$$

Thus,

$$t = \frac{\|q_t - p_Y^0\|_2}{\|q_0 - p_Y^0\|_2}. \tag{16}$$

Plug Equation (16) into Equation (15), we obtain

$$\|Tq_t - q_t\|_2 \geq \frac{1}{2}\left(\frac{\|q_t - p_Y^0\|_2}{\|q_0 - p_Y^0\|_2}\right)\|b\|_2 = \left(\frac{\|b\|_2}{2\|q_0 - p_Y^0\|_2}\right)\|q_t - p_Y^0\|_2.$$

Let

$$\alpha := \frac{\|b\|_2}{2\|q_0 - p_Y^0\|_2} > 0.$$

Then

$$\|Tq_t - q_t\|_2 \geq \alpha\,\|q_t - p_Y^0\|_2.$$

Thus there exists $p_Y^1 \in \mathcal{R}$ and $\alpha > 0$ such that

$$\|Tp_Y^1 - p_Y^1\|_2 \geq \alpha\,\|p_Y^1 - p_Y^0\|_2.$$

□

B.5. Unbiased Gradient Estimator for the Second Term of the Dual

$$\begin{aligned}
&\nabla_\theta \log \int_{\mathcal{X}} p_X^0(x)\cdot\exp\left(\int_{\mathcal{Y}}\phi_\theta(y)p_{Y|X}(y|x)\,dy\right)dx\Big|_{\theta=\theta^{(t)}}\\
=&\frac{\nabla_\theta \int_{\mathcal{X}} p_X^0(x)\cdot\exp\left(\int_{\mathcal{Y}}\phi_\theta(y)p_{Y|X}(y|x)\,dy\right)dx\Big|_{\theta=\theta^{(t)}}}{\int_{\mathcal{X}} p_X^0(u)\cdot\exp\left(\int_{\mathcal{Y}}\phi_{\theta^{(t)}}(v)p_{Y|X}(v|u)\,dv\right)du}\\
=&\int_{\mathcal{X}}\frac{p_X^0(x)\cdot\exp\left(\int_{\mathcal{Y}}\phi_{\theta^{(t)}}(y)p_{Y|X}(y|x)\,dy\right)}{\int_{\mathcal{X}} p_X^0(u)\cdot\exp\left(\int_{\mathcal{Y}}\phi_{\theta^{(t)}}(v)p_{Y|X}(v|u)\,dv\right)du}\nabla_{\theta^{(t)}}\left(\int_{\mathcal{Y}}\phi_{\theta^{(t)}}(y)p_{Y|X}(y|x)\,dy\right)dx\\
=&\int_{\mathcal{X}} p_X(x;\theta^{(t)})\left(\int_{\mathcal{Y}}\nabla_{\theta^{(t)}}\phi_{\theta^{(t)}}(y)p_{Y|X}(y|x)\,dy\right)dx\\
=&\int_{\mathcal{X}}\int_{\mathcal{Y}} p_X(x;\theta^{(t)})p_{Y|X}(y|x)\nabla_{\theta^{(t)}}\phi_{\theta^{(t)}}(y)\,dy\,dx
\end{aligned} \tag{17}$$

C. Experimental Details

C.1. Sample Availability

C.1.1. Synthetic and NPMLE Tasks

For all synthetic tasks, we assume that $p^0_X(x)$, $\nabla_x \log p^0_X(x)$, $p_{Y|X}(y|x)$, and $p^1_Y(y)$, are available for both exact sampling and density evaluation when required.

Conditional Sampling. We describe how $p^0_{X|Y}$ is sampled in our experiments for the synthetic tasks.

When both p^1_Y and $p_{Y|X}$ are Gaussian, $p^0_{X|Y}$ can be calculated in closed-form for exact sampling. Specifically, let $p^0_X = \mathcal{N}(\mu_0, \Sigma_0)$ and $p_{Y|X} = \mathcal{N}(\mathbf{a}^\top X + b, \sigma^2_{Y|X})$, where $\mu_0, \mathbf{a} \in \mathbb{R}^d$, $\Sigma_0 \in \mathbb{R}^{d\times d}$, and $b, \sigma_{Y|X} \in \mathbb{R}$. Note that Y can be equivalently represented as a linear transformation

$$Y = \tilde{\mathbf{a}}^\top \widetilde{X} + b,$$

where

$$\tilde{\mathbf{a}} \coloneqq \begin{pmatrix}\mathbf{a}\\ 1\end{pmatrix}, \qquad \widetilde{X} \coloneqq \begin{pmatrix}X\\ \epsilon\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mu_0\\ 0\end{pmatrix}, \begin{pmatrix}\Sigma_0 & \mathbf{0}\\ \mathbf{0}^\top & \sigma^2_{Y|X}\end{pmatrix}\right) \equiv \mathcal{N}(\tilde{\mu}, \tilde{\Sigma}).$$

Since this transformation is linear, $(Y, \widetilde{X})$ is jointly Gaussian. It follows that the conditional distribution of $\widetilde{X}$ given Y is

$$p^0_{\widetilde{X}|Y} \sim \mathcal{N}(\mu_{\widetilde{X}|Y}, \Sigma_{\widetilde{X}|Y}),$$

with

$$\mu_{\widetilde{X}|Y} = \tilde{\mu} + \tilde{\Sigma}\tilde{\mathbf{a}}(\tilde{\mathbf{a}}^\top\tilde{\Sigma}\tilde{\mathbf{a}})^{-1}\big(Y - (\tilde{\mathbf{a}}^\top\tilde{\mu} + b)\big),$$
$$\Sigma_{\widetilde{X}|Y} = \tilde{\Sigma} - \tilde{\Sigma}\tilde{\mathbf{a}}(\tilde{\mathbf{a}}^\top\tilde{\Sigma}\tilde{\mathbf{a}})^{-1}\tilde{\mathbf{a}}^\top\tilde{\Sigma}.$$

Finally, we sample $x \sim X|Y \equiv \mathcal{N}(\mu_{\widetilde{X}|Y,1:d}, \Sigma_{\widetilde{X}|Y,1:d})$ – the posterior distribution of $\widetilde{X}|Y$ excluding the dimension of noise ϵ.

When the Gaussian conditions are violated—for example, when $p_{Y|X}$ follows a Laplace distribution or when p^1_Y is estimated by KDE—we sample from the posterior $p^0_{X|Y}$ using a Metropolis–Hastings algorithm with a simple symmetric proposal, such as $\mathcal{N}(x, 1)$. For high-dimensional or more complex settings, more expressive parameterized proposal models are likely required for efficient sampling.

C.1.2. Images Tasks

For image-based tasks, we pre-train diffusion models on biased data to represent $p^0_{X(t)}$, together with temporal classifiers $p_{Y|X(t)}$.

We follow the training scheme and hyperparameter configurations in (Kim et al., 2024, Table 6). Specifically, $p^0_{X(t)}$ is implemented using EDM (Karras et al., 2022) with its second-order ODE sampler. The classifier $p_{Y|X(t)}$ is implemented as a two-layer U-Net encoder: the first layer serves as a fixed feature extractor with weights imported from ADM `https://github.com/openai/guided-diffusion`, while the second layer is a shallow, trainable U-Net encoder with a softmax output whose dimension equals the number of target classes. The classifier is trained on the full training sets of CIFAR-10 and CelebA (with targets defined by gender and hair color), achieving training accuracies of 99.9% and 99.7% on intact the images, respectively.

There are two minor deviations from the original configuration in (Kim et al., 2024). First, for CelebA we use a batch size of 256 instead of 128, following the recommendation in the authors' GitHub repository: `https://github.com/alsdudrla10/TIW-DSM`. Second, we reduce the number of diffusion training samples on CelebA from 200 Mimg to 87 Mimg due to computational resource constraints; empirically, this budget is sufficient to reach stable training and high-quality generations for our downstream adaptation tasks.

After pre-training, both the diffusion model $p^0_{X(t)}$ and the classifiers are frozen for all subsequent experiments. In addition, we utilize the score correction framework from DG (Kim et al., 2022): `mhttps://github.com/alsdudrla10/DG`.

C.2. Models

For synthetic tasks, we parameterize $\phi_\theta(\cdot) : \mathcal{Y} \to \mathbb{R}$ using simple neural networks. Specifically, we employ shallow multi-layer perceptrons (MLPs) with two fully connected hidden layers of widths (32, 16), each followed by ReLU activations.

For image tasks with diffusion models, we parameterize $\phi_\theta(\cdot, t) : [0, T] \to \mathbb{R}^{|\mathcal{Y}|}$ where $|\mathcal{Y}|$ denotes the number of target classes (e.g., 4 for CelebA and 10 for CIFAR-10). In this case, we again adopt shallow MLPs, consisting of three fully connected hidden layers of width (32, 32, 32) with ReLU activations.

C.3. Training

In synthetic and NPMLE tasks, we conduct five independent runs with different random seeds and choose the optimal hyperparameters (such as learning rates) for each single experimental setting determined by $(p_X^0, p_{Y|X}, p_Y^1)$, with the following heuristics:

1. For each set of hyperparameters, optimize the model with different random seeds.
2. Count the number of $\|\widehat{p_Y^*} - p_Y^1\|_2 \le \varepsilon$ in all runs and choose the set of hyperparameters that results in the **highest count of constraint satisfactions**.
   - When the constraint is satisfied in all runs, choose hyperparameters that result in the lowest average $KL(\widehat{p_X^*} \| p_X^0)$.
   - When there is a tie in the counts of constraint satisfactions and the constraint is not satisfied in some runs, choose hyperparameters that result in the lowest average $\|\widehat{p_Y^*}(y) - p_Y^1(y)\|_2$ over unsatisfied runs.

We search over the following grid of key hyperparameters:

- Learning rate: [1e-5, 1e-4, 1e-3]
- Batch size: [16, 32, 64]

Once the hyperparameters are fixed, we select checkpoints corresponding to the minimum training L_2 value and re-evaluate the model in terms of both KL divergence and L_2 distance to report the final results.

For image tasks, we do grid search over hyperparameters (mainly for learning rates) by tracking a moving average moving average of L_2 (window size: 5K images) computed from the generated samples used to estimate $\nabla_\theta L_2(\theta, t)$ **??**. We select the models with checkpoints that is closet to its minimum training L_2 dynamics, and report the results with the best model.

C.3.1. Evaluation

KL Divergence. For synthetic tasks, we need to estimate $KL\left(\widehat{p_X^*} \| p_X^0(x)\right)$, where p_X^0 admits a known density function. This estimation is challenging as it requires evaluating the density ratio $\frac{\widehat{p_X^*}(x)}{p_X(x)}$. Instead, we assume $\widehat{p_X^*}$ is approximately Gaussian, compute its empirical mean and covariance, and then use the closed-form formula for the KL divergence between Gaussians: for $p = \mathcal{N}(\mu_1, \Sigma_1)$ and $q = \mathcal{N}(\mu_2, \Sigma_2)$,

$$\mathrm{KL}(p\|q) = \tfrac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} + \mathrm{Tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_1 - \mu_2)^\top \Sigma_2^{-1}(\mu_1 - \mu_2) - d\right].$$

We also consider kernel density estimation (KDE) where we first approximate p_X^1 by $\hat{p}_X^1$ via KDE, and then compute

$$\mathrm{KL}(p_X^1, p_X) \approx \mathbb{E}_{x\sim p_X^1}\left[\log\frac{\hat{p}_X^1(x)}{p_X(x)}\right].$$

In most settings, these two estimators agree closely. However, for certain tasks such as NPMLE with $r = 0.2$ Section 4.2, the optimal solution is multi-modal, in which case KDE-based estimates are more appropriate. For high-dimensonal case, Gaussian approximation usually results in more robust estimates than KDE.

For very high-dimensional data, such as images, we instead report the Fréchet Inception Distance (FID) (Heusel et al., 2017), which compares distributions after projecting samples into a pre-trained feature space.

L_2 Distance. To estimate $\left\|\widehat{p_Y^*}(y) - p_Y^1(y)\right\|_2$, we first approximate

$$\widehat{p_Y^*}(y) = \mathbb{E}_{x \sim \widehat{p_X^*}(x)}[p_{Y|X}(y|x)].$$

When Y is discrete, $\left\|\widehat{p_Y^*}(y) - p_Y^1(y)\right\|_2$ reduces to the Euclidean distance between two finite-dimensional vectors. When Y is continuous, however, there are multiple ways to estimate $\left\|\hat{p}_Y^1(y;\theta) - p_Y^1(y)\right\|_2$. The simplest, which we adopt, is numerical integration, feasible since Y is one-dimensional.

D. Diffusion Models and Time-dependent Constrained Problems

D.1. Background

Score-Based Diffusion Models. Let p_X^0 denote the data distribution we aim to model. Score-based diffusion models (Song et al., 2021) consider diffusion processes that perturb $\boldsymbol{x}_0 \sim p_X^0$ into a noise variable $\boldsymbol{x}_T$, enabling data generation through the corresponding backward process (Anderson, 1982). Specifically, the forward process is modeled as the solution to a stochastic differential equation (SDE):

$$d\boldsymbol{x}_t = f(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t \tag{18}$$

where $\boldsymbol{x}_0 \sim p_X^0$; $\boldsymbol{w}_t$ is the standard Wiener process; the drift coefficient f and diffusion coefficient g are typically chosen such that $p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$ is Gaussian with known mean $\mu_t(f, g, \boldsymbol{x}_0)$ and covariance $\Sigma_t(f, g, \boldsymbol{x}_0)$. The corresponding reversed-time SDE, i.e., the data generation process, is:

$$d\boldsymbol{x}_t = [f(\boldsymbol{x}_t, t)dt - g^2(t)\nabla \log p_t(\boldsymbol{x}_t)]d\bar{t} + g(t)d\bar{\boldsymbol{w}}_t \tag{19}$$

where $\bar{\boldsymbol{w}}$ is the standard Wiener process with reversed time $\bar{t} \in [T, 0]$. Once $\nabla \log p_t(\boldsymbol{x}_t)$ is known for any t, we can sample $p_X^0(\boldsymbol{x})$ from the reversed-time SDE using numerical methods like the Euler-Maruyama method.

Denoising score matching (DSM) is a popular approach to learning $\nabla \log p_t(\boldsymbol{x}_t)$ with a parameterized model $s_\eta(\boldsymbol{x}_t, t)$:

$$\mathcal{L}_{\text{DSM}}(\eta; p_0) = \mathbb{E}_{t\sim\mathcal{U}[0,T]}\left[\mathbb{E}_{p_0(\boldsymbol{x}_0)}\left[\mathbb{E}_{p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\lambda(t)\left\|s_\eta(\boldsymbol{x}_t, t) - \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\right\|_2^2\right]\right]\right], \tag{20}$$

where $\lambda(t)$ are weights over time.

Time-independent Fair Generation Kim et al. (2024) study the same problem as (Choi et al., 2020, see **??**) [SW: Fix this] with a focus on diffusion models. Simply put, they want to utilize the abundant samples from a biased distribution p_X^0 when training diffusion models for the target distribution $p_X^1 \neq p_X^0$ whose samples are scarce. An importance reweighing scheme can be simply applied to the denoising score matching objective (20) in a time-independent Importance reWeighted (IW) manner:

$$\mathcal{L}_{\text{IW-DSM}}(\eta; p_X^0, w_\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,T]}\left[\mathbb{E}_{p_X^0(\boldsymbol{x}_0)}\left[w_\theta(\boldsymbol{x}_0)\mathbb{E}_{p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\lambda(t)\left\|s_\eta(\boldsymbol{x}_t, t) - \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\right\|_2^2\right]\right]\right], \tag{21}$$

where $w_\theta(\cdot)$ is a parameterized model trained in advance via binary classification (Sugiyama et al., 2012) to model the density ratio $\frac{p_X^1(\cdot)}{p_X^0(\cdot)}$.

Kim et al. (2024) observe that the error of w_θ is often caused by the *density chasm* (Rhodes et al., 2020a) between p_X^0 and p_X^1, which can be greatly alleviated when noise is added to data along the diffusion process. Formally, they expand $w_\theta(\boldsymbol{x}, t)$ by adding the time axis to model the time-dependent density ratio $\frac{p_t^1(\boldsymbol{x}_t)}{p_t^0(\boldsymbol{x}_t)} = \frac{\int p_X^1(\boldsymbol{x}_0)p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\,d\boldsymbol{x}_0}{\int p_X^0(\boldsymbol{x}_0)p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\,d\boldsymbol{x}_0}$, and train it to minimize the time-dependent Binary Cross Entropy (BCE) loss:

$$\mathcal{L}_{\text{BCE}}(\eta; p_X^0, w_\theta) = \int_0^T \lambda'(t)\left[\mathbb{E}_{p_t^0(\boldsymbol{x}_t)}[-\log d_\theta(\boldsymbol{x}_t, t)] + \mathbb{E}_{p_t^1(\boldsymbol{x}_t)}[-\log(1 - d_\theta(\boldsymbol{x}_t, t)]\right] dt, \tag{22}$$

where $\lambda'(t)$ is a weighting function. The time-dependent density ratio is then represented by

$$w_\theta(\boldsymbol{x}_t, t) = \frac{d_\theta(\boldsymbol{x}_t, t)}{1 - d_\theta(\boldsymbol{x}_t, t)}.$$

A central result of Kim et al. (2024, Theorem 1) is that $\mathcal{L}_{\text{IW-DSM}}$ can be rewritten (up to a normalizing constant) as the following Time-dependent Importance reWeighting (TIW) form such that $w_\theta(\boldsymbol{x}_t, t)$ is explicitly involved in the objetive and thus improves training:

$$\begin{aligned}&\mathcal{L}_{\text{TIW-DSM}}(\eta; p_X^0, w_\theta)\\ =&\mathbb{E}_{t\sim\mathcal{U}[0,T]}\left[\mathbb{E}_{p_X^0(\boldsymbol{x}_0)}\left[\mathbb{E}_{p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\lambda(t)w_\theta(\boldsymbol{x}_t, t)\left\|s_\eta(\boldsymbol{x}_t, t) - \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log w_\theta(\boldsymbol{x}_0, t)\right\|_2^2\right]\right]\right].\end{aligned} \tag{23}$$

In other words, $\mathcal{L}_{\text{TIW-DSM}}$ only changes the way of optimizing $\mathcal{L}_{\text{IW-DSM}}$ with the gain from more accurately estimated density ratios $w_\theta(\boldsymbol{x}_t, t)$.

They further show in the ablation study that replacing $w_\theta(\boldsymbol{x}_0, t)$ in the weights of (23) by equal weights (EW) of 1 leads to comparable results:

$$\begin{aligned}&\mathcal{L}_{\text{TIW-DSM-EW}}(\eta; p_0, w_\theta)\\ =&\mathbb{E}_{t\sim\mathcal{U}[0,T]}\left[\mathbb{E}_{p_X^0(\boldsymbol{x}_0)}\left[\mathbb{E}_{p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\lambda(t)\left\|s_\eta(\boldsymbol{x}_t,t)-\nabla\log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)-\nabla\log w_\theta(\boldsymbol{x}_t,t)\right\|_2^2\right]\right]\right].\end{aligned} \tag{24}$$

Alternative to optimizing $\mathcal{L}_{\text{TIW-DSM-EW}}(\eta; p_0, w_\theta)$ directly, this form suggests a correction-based sampling scheme – to see this, replace $s_\eta(\boldsymbol{x}_t, t) - \nabla \log w_\theta(\boldsymbol{x}_t, t)$ with $s'_{\eta,\theta}(\boldsymbol{x}_t, t)$ in $\mathcal{L}_{\text{TIW-DSM-EW}}(\eta; p_0, w_\theta)$. The objective is then identical to (21), and thus $s'_{\eta,\theta}(\boldsymbol{x}_t, t)$ converges to $\nabla \log p_t^0(\boldsymbol{x}_t, t)$. It follows that $s'_{\eta,\theta}(\boldsymbol{x}_t, t) + \nabla \log w_\theta(\boldsymbol{x}_t, t)$ converges to $\nabla \log p_t^0(\boldsymbol{x}_t) + \nabla \log \frac{p_t^1(\boldsymbol{x}_t)}{p_t^0(\boldsymbol{x}_t)} = \nabla \log p_t^1(\boldsymbol{x}_t)$. In other words, one can train $s_\eta(\boldsymbol{x}_t, t)$ with (20) and $w_\theta(\boldsymbol{x}_t, t)$ with (22) separately; at inference time, use the corrected score $s_\eta(\boldsymbol{x}_t, t) + \nabla \log w_\theta(\boldsymbol{x}_t, t)$ for sampling. Nonetheless, such an approach trades off inference-time for flexibility and re-usability.

Time-dependent Discriminator Guidance. There is an closely related earlier work by Kim et al. (2022). In this study, the authors aim to improve the sample quality of a given diffusion model with score function $s_\eta(\boldsymbol{x}_t, t)$ by similar score adjustment. To do this, they first learn density ratio $w_\theta(\boldsymbol{x}_t, t) \approx \frac{p_t^1(\boldsymbol{x}_t)}{p_t^0(\boldsymbol{x}_t)} = \frac{\int p_X^1(\boldsymbol{x}_0) p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\, d\boldsymbol{x}_0}{\int p_X^0(\boldsymbol{x}_0) p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\, d\boldsymbol{x}_0}$, where, in this case, p_X^1 is the empirical distribution of real data and p_X^0 is the data distribution associated with $s_\eta(\boldsymbol{x}_t, t)$, by minimizing the BCE loss (22). At inference time, use the corrected score $s_\eta(\boldsymbol{x}_t, t) + \nabla \log w_\theta(\boldsymbol{x}_t, t)$ for sampling.

D.2. Proposal: Time-dependent Constrained Problem for Diffusion Models

In this section, we study the constrained problem (1) when the reference distribution p_X^0 is given by a pre-trained diffusion model with parameterized score function $s_\eta(\boldsymbol{x}_t, t)$. A naïve approach, analogous to (21), is to adapt for $t = 0$ only. However, it means a sudden huge jump will appear in the adapted scores at $t = 0$: $\nabla_{\boldsymbol{x}_0} \log p_{\boldsymbol{x}_0}(\boldsymbol{x}_0; \theta) = s_\eta(\boldsymbol{x}_0, 0) + w_\theta(\boldsymbol{x}_0)$ is adapted while $\nabla_{\boldsymbol{x}_t} \log p_{X(t)}(\boldsymbol{x}_t) = s_\eta(\boldsymbol{x}_t, t)$ remains intact for $t > 0$. Inspired by (Kim et al., 2024; 2022), we consider an extended time-dependent constrained problem:

$$\begin{aligned} p^*_{X(t)} = \underset{p_{X(t)}\in\mathcal{P}_\mathcal{X}}{\arg\min} \quad & KL(p_{X(t)} \| p^0_{X(t)}) \\ \text{subject to} \quad & \|p^1_{Y(t)} - p_{Y|X(t)} \circ p_{X(t)}\|_2 \le \varepsilon \end{aligned} \tag{25}$$

for $t \in [0, T]$, where $p^0_{X(t)}$ is characterized by the diffusion process with $s_\eta(\boldsymbol{x}_t, t)$ and a known noise $\boldsymbol{x}_T$, and $p^1_{Y(t)} = \int_\mathcal{Y} p^0_{X(t)}(\boldsymbol{x}_t) p_{Y|X(t)}(y|\boldsymbol{x}_t)\, d\boldsymbol{x}_t$ is the time-dependent marginal distribution under noisy transformation $p_{Y|X(t)}$. Note that our ultimate goal (1) is recovered at $t = 0$; for $t > 0$, we want to relax our problem such that $w_\theta(\boldsymbol{x}_t, t) \approx \frac{p^*_{X(t)}(\boldsymbol{x}_t)}{p^0_{X(t)}(\boldsymbol{x}_t)} = \frac{\int p^*_{X(0)}(\boldsymbol{x}_0) p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\, d\boldsymbol{x}_0}{\int p^0_{X(0)}(\boldsymbol{x}'_0) p_t(\boldsymbol{x}_t|\boldsymbol{x}'_0)\, d\boldsymbol{x}'_0}$ is smooth in time t and easy to estimate. In addition, one can further relax the problem by time-dependent tolerance $\varepsilon(t) \ge \varepsilon$ provided that $\varepsilon(0) = \varepsilon$.

Unlike (Kim et al., 2024) where a target distribution p_X^1 is available in scarce samples, however, our target solution $p^*_{X(t)}$ is unknown in distribution nor samples. That is, we cannot learn $w_\theta(\boldsymbol{x}_t, t)$ by optimizing $\mathcal{L}_{BCE}$. Unlike Kim et al. (2024), in our setting the target distribution $p^*_{X(t)}$ is not given through samples (or even an explicit density). Consequently, we cannot learn $w_\theta(\boldsymbol{x}_t, t)$ using a BCE objective such as equation 22. Instead, we train the time-dependent dual potential $w_\theta(\boldsymbol{x}_t, t)$ by maximizing the time-averaged dual objective $L(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,T]}[\lambda(t) L(\theta, t)]$ where

$$\begin{aligned}&\max_{\theta\in\Theta} L(\theta, t)\\ =&\max_{\theta\in\Theta}\left\{\int_\mathcal{Y}\phi_\theta(y,t)p_Y^1(y)\,dy - \log\int_\mathcal{X} p^0_{X(t)}(\boldsymbol{x}_t)\cdot\exp\left(\int_\mathcal{Y}\phi_\theta(y,t)p_{Y|X(t)}(y|\boldsymbol{x}_t)\,dy\right)d\boldsymbol{x} - \varepsilon\left(\int_\mathcal{Y}\phi_\theta^2(y,t)\,dy\right)^{\frac{1}{2}}\right\},\end{aligned} \tag{26}$$

and $\lambda(t)$ is a temporal weighting function.

The training scheme remains mostly identical with additional time-axis extension. For instance, the unbiased gradient estimator of $L_2(\theta, t)$ is given by:

$$\nabla_\theta L_2(\theta, t) = \int_{\mathcal{X}} \int_{\mathcal{Y}} p_{X(t)}(\boldsymbol{x}_t; \theta) p_{Y|X(t)}(y|\boldsymbol{x}_t) \nabla_\theta \phi_\theta(y, t)\, dy\, d\boldsymbol{x}_t. \tag{27}$$

It is noteworthy that $p_{X(t)}(\boldsymbol{x}_t; \theta) \propto p_{X(0)}(\boldsymbol{x}_0; \theta) p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$ implies a two-stage sampling – we first sample $\boldsymbol{x}_0 \sim p_{X(0)}(\boldsymbol{x}_0; \theta)$ by invoking the diffusion sampler with the adapted score function:

$$\begin{aligned}
\nabla_{\boldsymbol{x}_t} \log p_{X(t)}(\boldsymbol{x}_t; \theta) &= \nabla_{\boldsymbol{x}_t} \log p^0_{X(t)}(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log w_\theta(\boldsymbol{x}_t, t) \\
&= s_\eta(\boldsymbol{x}_t, t) + \nabla_{\boldsymbol{x}_t} \int_{\mathcal{Y}} \phi_\theta(y, t) p_{Y|X(t)}(y|\boldsymbol{x}_t)\, dy \qquad (28) \\
&= s_\eta(\boldsymbol{x}_t, t) + \int_{\mathcal{Y}} \phi_\theta(y, t) \nabla_{\boldsymbol{x}_t} p_{Y|X(t)}(y|\boldsymbol{x}_t)\, dy \\
&= s_\eta(\boldsymbol{x}_t, t) + \int_{\mathcal{Y}} \phi_\theta(y, t) \cdot \nabla_{\boldsymbol{x}_t} \log p_{Y|X(t)}(y|\boldsymbol{x}_t) \cdot p_{Y|X(t)}(y|\boldsymbol{x}_t)\, dy, \qquad (29)
\end{aligned}$$

followed by adding time-dependent noise per $p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$ defined by the forward diffusion process.

At inference-time, we sample data from $\boldsymbol{x}_0 \sim p_{X(0)}(\boldsymbol{x}_0; \widehat{\theta}^*)$ with the adapted score defined above without adding noise, where $\widehat{\theta}^* = \text{argmax}_\theta L(\theta, t)$.

For continuous Y, (29) is preferred as it admits an efficient Monte Carlo approximation by sampling $y \sim p_{Y|X(t)}(\cdot \mid \boldsymbol{x}_t)$. For discrete Y, direct computation (28) is more appealing as the integration reduces to a dot product of two finite-dimensional real vectors which is exact and efficient. Particularly, for image tasks with discrete class labels, (28) is adopted and the diffusion sampler remains requiring $\mathcal{O}(T)$ score evaluations. However, we do observe roughly a $2\times$ slowdown in wall-clock sampling time. Our inspection reveals that the extra time is dominated by evaluating the classifier $p_{Y|X(t)}(y|\boldsymbol{x}_t)$ (due to its architecture and model size; see Section C.1.2).

E. Additional Experimental Results

E.1. Optimal Solutions

Specifically, we consider the problem (1) with $p_X^0(x) \coloneqq \mathcal{N}(\mu^0, (\sigma_X^0)^2)$, $p_{Y|X}(y|x) \coloneqq \mathcal{N}(2x+1, 1)$ and $p_Y^1(y) = \mathcal{N}(-1, {\sigma_Y^1}^2)$. In particular, we consider the strict constraint $\varepsilon = 0$.

If the problem is feasible (i.e., $\sigma^1 > 1$ in this case), then the solution is given by

$$p_X^1(x) = \mathcal{N}\left(-1, \left(\frac{\sqrt{{\sigma_Y^1}^2 - 1}}{2}\right)^2\right) \coloneqq \mathcal{N}(\mu_X^1, (\sigma_X^1)^2) \tag{30}$$

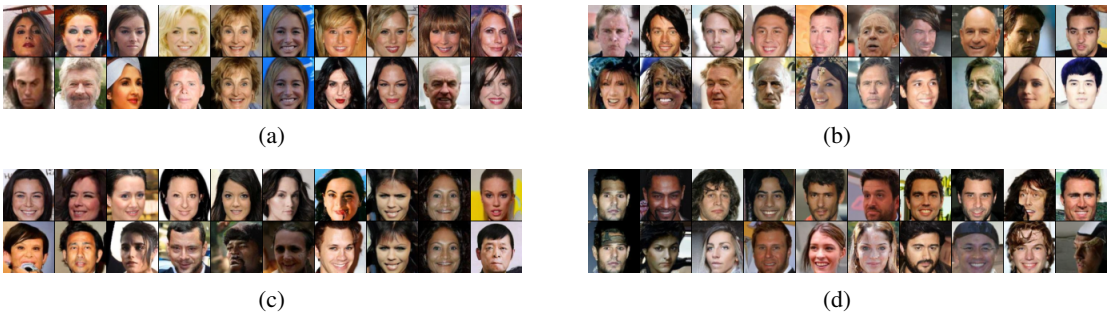
To see this, note that mean μ_X^1 and variance $(\sigma_X^1)^2$ of p_Y^1 are uniquely determined by the $2 \cdot \mu_X^1 + 1 = -1$ and $2^2 \cdot (\sigma_X^1)^2 + 1 = {\sigma_Y^1}^2$. It follows from simple calculations that KL divergence is minimized when $p_X^1(x)$ is a normal distribution (see: https://math.stackexchange.com/a/2354469/719714).

E.2. Image Tasks

	Task	p_Y^0	$\widehat{p_Y^*}$
CelebA	Balanced	[0.389, 0.326, 0.143, 0.142]	[0.260, 0.264, 0.246, 0.229]
CIFAR10 (LT, 5%)	Balanced	[0.202, 0.191, 0.095, 0.095, 0.085, 0.085, 0.07, 0.063, 0.057, 0.057]	[0.093, 0.096, 0.103, 0.094, 0.103, 0.110, 0.107, 0.092, 0.103, 0.098]
	Reversed		[0.048, 0.072, 0.059, 0.078, 0.068, 0.101, 0.105, 0.076, 0.203, 0.189]

Table 2. **Class distributions of generated images before and after adaptation.** Class labels are listed in the same order as the distributions reported in the table. For CelebA, the order is: female without black hair (F,NB), male without black hair (M,NB), female with black hair (F,B), and male with black hair (M,B). For CIFAR-10, the order is: airplane, automobile, bird, cat, truck, horse, frog, ship, deer, and dog. We report predicted class labels obtained from the trained classifier $p_{Y|X(0)}$ at $t = 0$, rather than ground-truth annotations, to align with our problem setup.

E.2.1. CelebA, Fair Generation

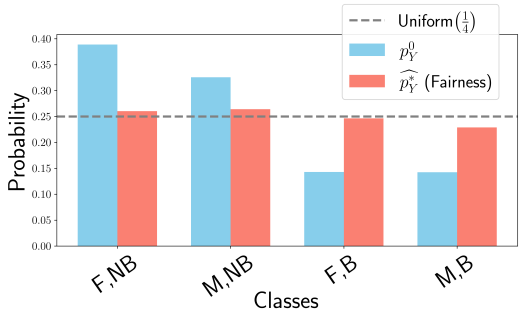


(a)

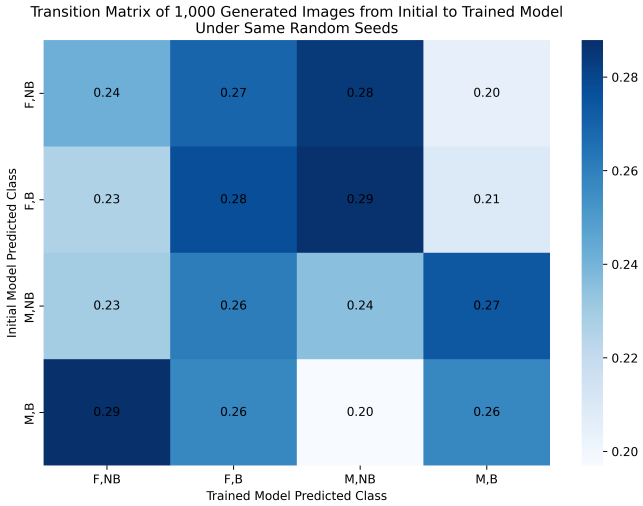
(b)

(c)

(d)



(e)



(f)

Figure 8. **(CelebA)** Our proposal effectively balances the class distribution of CelebA with respect to the target attributes (gender and hair color). Panels (a–d) show samples generated with identical random seeds before and after adaptation, illustrating distribution-level attribute conversion across gender and hair color. Panels (e) and (f) report the induced class distribution shift and transition dynamics, respectively. Overall, while individual image changes may appear subtle, the aggregate class distribution moves significantly toward the uniform target.

E.2.2. CIFAR-10, FAIR GENERATION

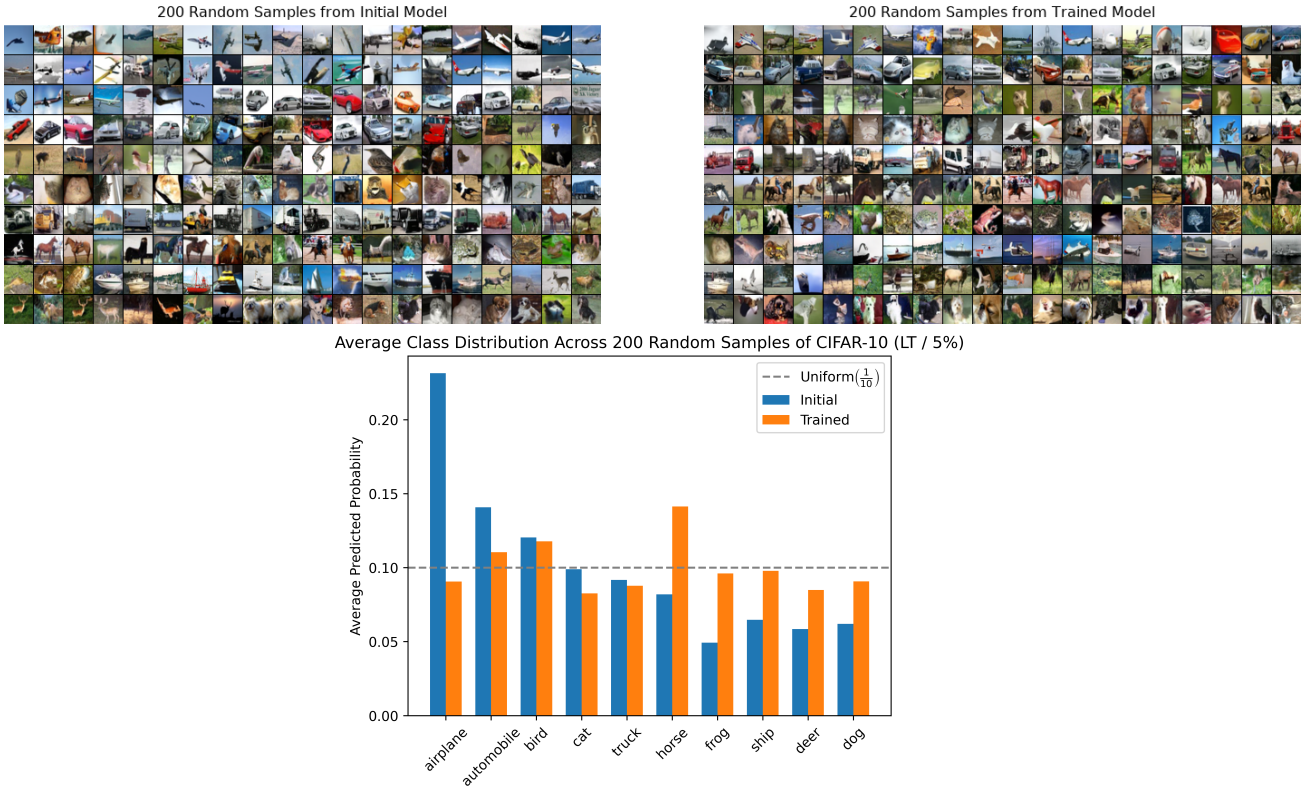


Figure 9. **(CIFAR-10 (LT/5%), Fair Generation)** Our proposal effectively corrects the biased class distribution of CIFAR-10 (LT/5%). From 200 randomly sampled images, we observe that image quality is largely preserved while the class distribution is significantly more balanced.

E.2.3. CIFAR-10, Reversed Class Distribution

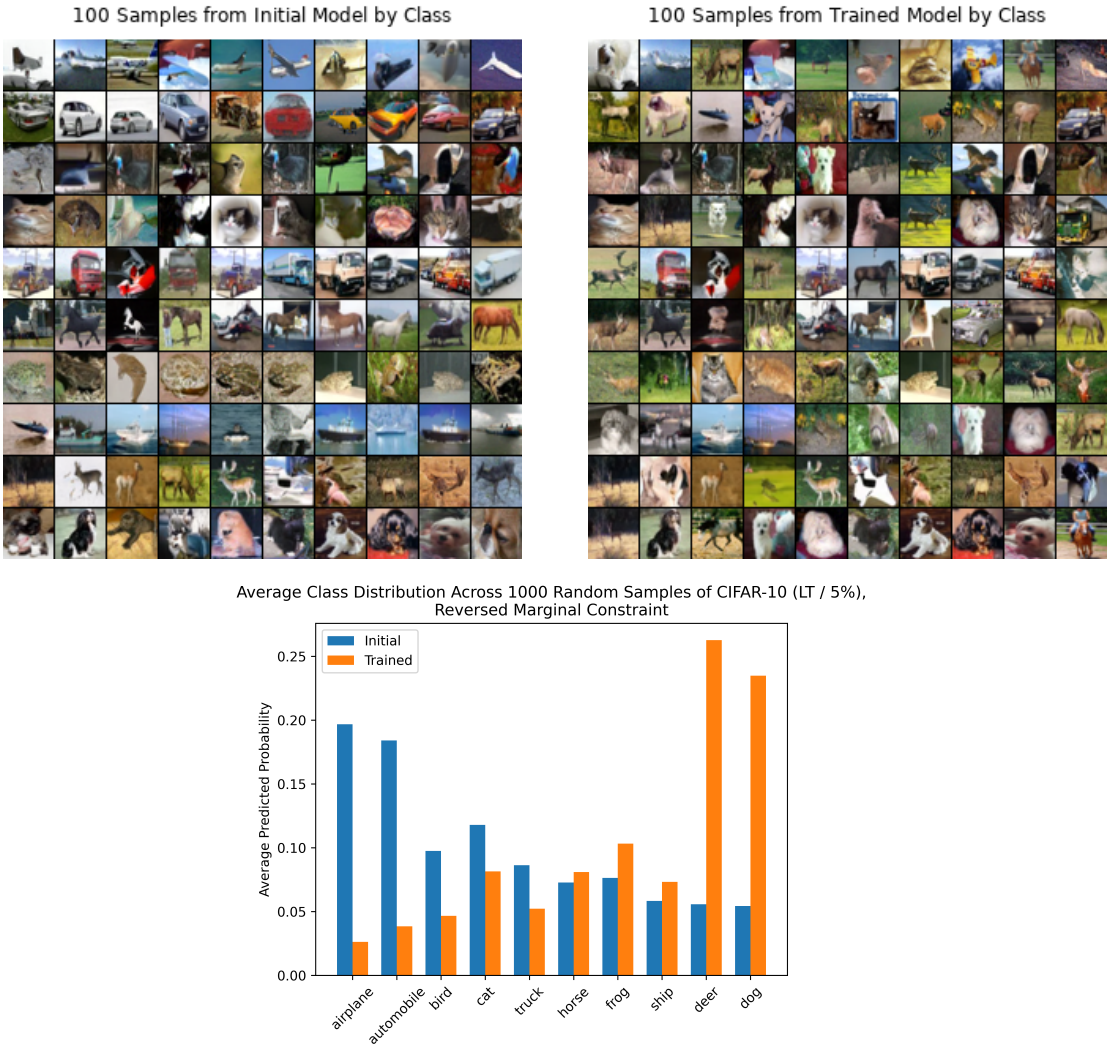


Figure 10. **(CIFAR-10 (LT/5%), Reversed Distribution)** Our proposal effectively flips the class distribution of CIFAR-10 (LT/5%). Using samples generated under the same random seeds, we observe that image quality is largely preserved while the class distribution is reversed in accordance with the marginal constraint. In particular, most airplanes and automobiles are transformed into underrepresented classes, whereas rare classes such as deer and dogs tend to remain unchanged.

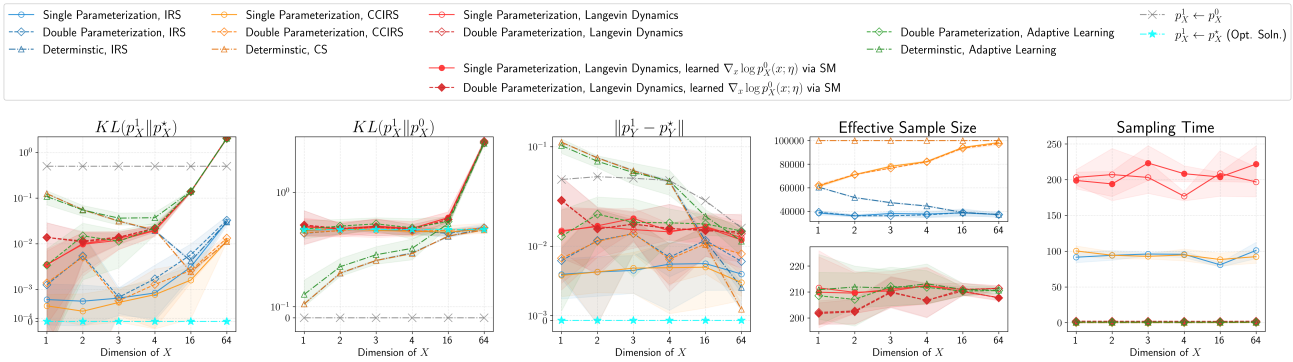
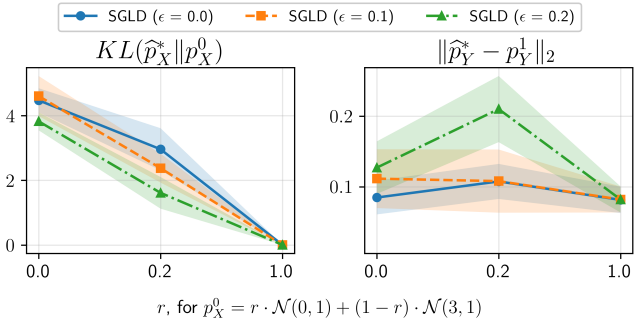
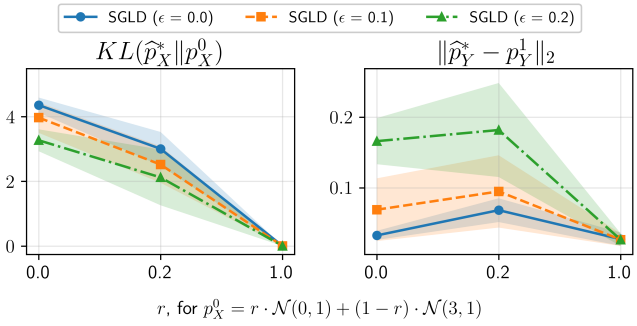


Figure 11. **Results on high-dimensional synthetic tasks.** This experiment extends the synthetic setup in Section 4 to higher dimensions. We take $p_X^0 = \mathcal{N}(0, \mathbf{I}_d)$, $f(\mathbf{x}) = \mathbf{1}^\top \mathbf{x}$, $p_{Y|X} = \mathcal{N}(\mathbf{x}, \mathbf{I}_d)$ and $p_Y^1 = \mathcal{N}(\sqrt{d}, d+1)$, where d is the dimension of X. The problem remain feasible with the ground-truth solution $p_X^* = \mathcal{N}\Big((\frac{1}{\sqrt{d}}, \ldots, \frac{1}{\sqrt{d}}), \mathbf{I}_d\Big)$. Methods designed for probabilistic constraints perform well in low dimensions ($d \leq 32$), but are outperformed by deterministic schemes (denoted by blue and orange triangles) at $d = 64$ in terms of L_2 distance. In contrast, deterministic approaches—both importance resampling and conditional sampling—perform poorly in low dimensions, yet improve as d increases, since the effective noise in the constraint diminishes with dimensionality. The "double parameterization" variants in the legend approximate $w_\theta(x) = \int \mathcal{Y} \phi_\theta(y) p_{Y|X}(y|x), dy$ using an auxiliary neural network, improving sampling efficiency at inference time at the cost of additional training computation.

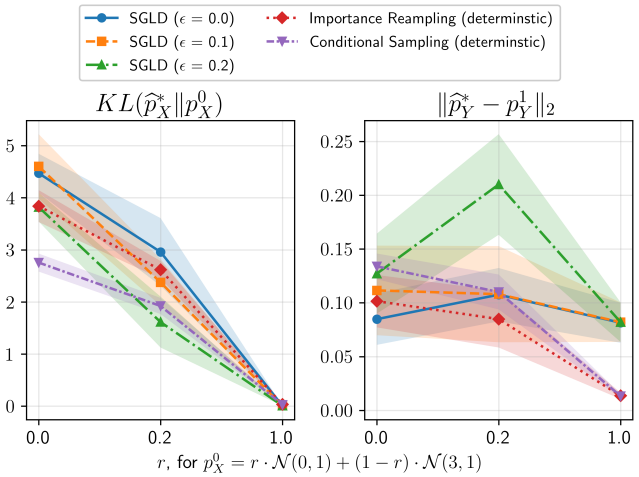


(a) Number of samples for p_Y^1 KDE = 100.

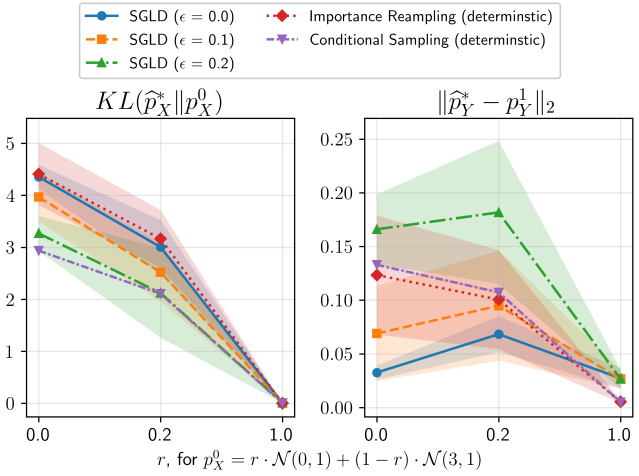


(b) Number of samples for p_Y^1 KDE = 100,000.

Figure 12. Results on empirical Bayes tasks.



(a) Number of samples for p_Y^1 KDE = 100.



(b) Number of samples for p_Y^1 KDE = 100,000.

Figure 13. Results on empirical Bayes tasks (with solutions for deterministic transformation).

Algorithm 2 SGLD+SGD approach to optimize the dual objective and simulate from the primal solution

Input: Monte Carlo batch sizes N, M, K, training iterations T, SGLD step size α, new sample size n.

function `SGLD` (parameter θ; state x^0; number of steps K)
  for $k = 1, \ldots, K$ **do**
    Sample $\{y_j\}_{j=1}^M \sim p_{Y|X}(\cdot|x^{k-1})$, $z_k \sim \mathcal{N}(0,1)$
    $\widehat{\nabla_x \log p_X^1}(x^{k-1};\theta) \leftarrow \nabla_x \log p_X^0(x^{k-1}) \; + \frac{1}{M}\sum_{j=1}^M \phi_\theta(y_j) \cdot \nabla_x \log p_{Y|X}(y_j|x^{k-1})$
    $x^k \leftarrow x^{k-1} + \alpha \widehat{\nabla_x \log p_X^1}(x^{k-1};\theta) + \sqrt{2\alpha} z_k$
  end for
  return $\{x^k\}_{k=1}^K$
end function

Training – Learn $\widehat{\theta}^*$ via SGD + SGLD
Initialize: $\theta^{(0)}$, $x^{(0)} \sim p_X^0$
for $t = 1, \ldots, T$ **do**
  Compute empirical estimate of the 1^{st} *term:*
    Sample $\{y_{1,i}\}_{i=1}^N \overset{i.i.d.}{\sim} p_Y^1$
    $l_1^{(t)} \leftarrow -\frac{1}{N}\sum_{i=1}^N \phi_{\theta^{(t-1)}}(y_{1,i})$
    $\{x_i\}_{i=1}^N \leftarrow$ `SGLD`$(\theta^{(t-1)}, x^{(t-1)}, N)$
  Compute proxy of the 2^{nd} *term for unbiased gradient estimate:*
    Sample $\{y_{2,i}\}_{i=1}^N \sim p_{Y|X}(\cdot|x_i)$
    $l_2^{(t)} \leftarrow \frac{1}{N}\sum_{i=1}^N \phi_{\theta^{(t-1)}}(y_{2,i})$
  Compute empirical estimate of the 3^{rd} *term:*
    Sample $\{y_{\text{reg},i}\}_{i=1}^N \overset{i.i.d.}{\sim} p_Y^1$
    $l_{\text{reg}}^{(t)} \leftarrow \varepsilon \left(\frac{1}{N}\sum_{i=1}^N \frac{\phi^2_{\theta^{(t)}}(y_{\text{reg},i})}{p_Y^1(y_{\text{reg},i})} \right)^{1/2}$
    $l^{(t)} \leftarrow l_1^{(t)} + l_2^{(t)} + l_{\text{reg}}^{(t)}$
    $\theta^{(t)} \leftarrow \text{SGD\_step}\big(\theta^{(t-1)}, \nabla_\theta l^{(t)}\big)$ {Update $\theta^{(t)}$}
    $x^{(t)} \leftarrow x_N$ {Update $x^{(t)}$}
end for

Simulation – Sample from $\widehat{p}_X^* \equiv p_X(x;\widehat{\theta}^*)$ via SGLD
$\{x_j\}_{j=1}^n \leftarrow$ `SGLD`$(\widehat{\theta}^*, x^{(T)}, n)$
Output: $\{x_j\}_{j=1}^n$