Zero-Shot Conditioning of Score-Based Diffusion Models by Neuro-Symbolic Constraints

Davide Scassola[1,2], **Sebastiano Saccani**[2], **Ginevra Carbone**[2], **Luca Bortolussi**[1]

[1]AILAB, University of Trieste, Trieste, Italy
[2]Aindo, AREA Science Park, Trieste, Italy
davide.scassola@phd.units.it, sebastiano@aindo.com, ginevracoal@gmail.com, lbortolussi@units.it

Abstract

Score-based diffusion models have emerged as effective approaches for both conditional and unconditional generation. Still conditional generation is based on either a specific training of a conditional model or classifier guidance, which requires training a noise-dependent classifier, even when a classifier for uncorrupted data is given. We propose a method that, given a pre-trained unconditional score-based generative model, samples from the conditional distribution under arbitrary logical constraints, without requiring additional training. Differently from other zero-shot techniques, that rather aim at generating valid conditional samples, our method is designed for approximating the true conditional distribution. Firstly, we show how to manipulate the learned score in order to sample from an un-normalized distribution conditional on a user-defined constraint. Then, we define a flexible and numerically stable neuro-symbolic framework for encoding soft logical constraints. Combining these two ingredients we obtain a general, but approximate, conditional sampling algorithm. We further developed effective heuristics aimed at improving the approximation. Finally, we show the effectiveness of our approach in approximating conditional distributions for various types of constraints and data: tabular data, images and time series.

Code — https://github.com/DavideScassola/score-based-constrained-generation

1 Introduction

Score-based (Song and Ermon 2019) and diffusion (Ho, Jain, and Abbeel 2020; Sohl-Dickstein et al. 2015) generative models based on deep neural networks have proven effective in modelling complex high-dimensional distributions in various domains. Controlling these models in order to obtain desirable features in samples is often required, still most conditional models require additional constraint-specific training in order to perform conditional sampling.

This represents a limit since either one needs to train an extremely flexible conditional model (as those based on text prompts), or alternatively to train a conditional model for any specific constraint to be enforced. Moreover, these conditional models often lack robustness, since the constraint is only learned through labelled data, even when the constraint is a user-defined function. As a consequence, zero-shot conditional generation with arbitrary but formal logical constraints specified at inference time is currently hard. This would be useful in different contexts, for example:

- *Tabular data*: generation of entries that obey formal requirements described by logical formulas, without a specific training for every formula.
- *Surrogate models*: using unconditional surrogate models to efficiently sample imposing physical constraints, or exploring scenarios defined by additional constraints.

Text prompt conditioned image generation with diffusion models (Rombach et al. 2022) has proven extremely flexible and effective, still it lacks fine grained control and requires a massive amount of labelled samples.

Recent work focuses on methods to perform guided diffusion on images, without the need to retrain a noise-dependent classifier (Bansal et al. 2023; Graikos et al. 2022; Kadkhodaie and Simoncelli 2021; Nair et al. 2023), but the performance of the approximation of the conditional distribution is never evaluated.

In this article we develop a method for sampling from pre-trained unconditional score-based generative models, enforcing arbitrary user-defined logical constraints, that does not require additional training. Despite being originally designed for tabular data, we also show the application of our method to images and time series. In summary, we present the following key contributions:

- We develop a zero-shot method for applying constraints to pre-trained unconditional score-based generative models. The method enables sampling approximately from the conditional distribution given a soft constraint.
- We define a general neuro-symbolic language for building soft constraints that corresponds to logical formulas. These constraints are numerically stable, satisfy convenient logical properties, and can be relaxed/hardened arbitrarily through a control parameter.
- We test our method on different types of datasets and constraints, showing good performance on *approximating conditional distributions* on tabular data, while previous methods were rather meant to obtain high-quality samples that satisfy some given constraints.

- Comparing our method with a state-of-the-art method (Bansal et al. 2023), we gather evidence that plug-and-play conditioning techniques designed for images are not necessarily suited for modelling the true conditional distribution. Moreover, we show our neuro-symbolic language to be useful for defining constraints also within this other method.
- We show that our method allows to sample conditioning on constraints that involve multiple data instances.

2 Background

Score-Based Generative Models

Score-based generative models (Song and Ermon 2019; Song et al. 2021), are a class of models developed in recent years, closely related to diffusion probabilistic models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020). Given the objective to sample from the distribution $p(\mathbf{x})$ that generated the data, these models aim at estimating the (*Stein*) score of $p(\mathbf{x})$, defined as $\mathbf{s}(\mathbf{x}) := \nabla_{\mathbf{x}} \ln p(\mathbf{x})$ and then use sampling techniques that exploit the knowledge of the score of the distribution. There are several methods for estimating the score: score matching (Hyvärinen and Dayan 2005), sliced score matching (Song et al. 2020), and denoising score matching (Vincent 2011). Denoising score matching is probably the most popular one, it uses corrupted data samples $\tilde{\mathbf{x}}$ in order to estimate the score of the distribution for different levels of added noise, which is in practice necessary for sampling in high dimensional spaces (Song and Ermon 2019).

Given a neural network $\mathbf{s}_\theta(\mathbf{x}, t)$ and a diffusion process $q_t(\tilde{\mathbf{x}}|\mathbf{x})$ defined for $t \in [0, 1]$ such that $q_0(\tilde{\mathbf{x}}|\mathbf{x}) \approx \delta(\mathbf{x})$ (no corruption) and $q_1(\tilde{\mathbf{x}}|\mathbf{x})$ is a fixed prior distribution (e.g., a Gaussian), the denoising score matching loss is:

$$\mathbb{E}_{t\sim u(0,1),\mathbf{x}\sim p(\mathbf{x}),\tilde{\mathbf{x}}\sim q_t(\tilde{\mathbf{x}}|\mathbf{x})} \|\mathbf{s}_\theta(\tilde{\mathbf{x}}, t) - \nabla_{\tilde{\mathbf{x}}} \ln q_t(\tilde{\mathbf{x}}|\mathbf{x})\|_2^2$$

With sufficient data and model capacity, denoising score matching ensures $\mathbf{s}_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$ for almost all $\mathbf{x}$ and t, where $p_t(\mathbf{x}) := \int q_t(\mathbf{x}|\mathbf{x}_0)p_0(\mathbf{x}_0)d\mathbf{x}_0$ is the distribution of the data for different levels of added noise. Given the estimate of the time/noise dependent score $\mathbf{s}_\theta(\mathbf{x}, t)$, one can resort to different techniques for sampling from $p(\mathbf{x}) = p_0(\mathbf{x})$ as annealed Langevin dynamics (Song and Ermon 2019), denoising diffusion probabilistic models (Ho, Jain, and Abbeel 2020) or stochastic differential equations (Song et al. 2021), a generalization of the aforementioned approaches.

Conditional Sampling with Score-Based Models

Given a joint distribution $p(\mathbf{x}, \mathbf{y})$, one is often interested in sampling from the conditional distribution $p(\mathbf{x}|\mathbf{y})$, where $\mathbf{y}$ is for example a label. While it is possible to directly model the conditional distribution (as usually done in many generative models), in this case by estimating $\nabla_{\mathbf{x}} \ln p_t(\mathbf{x}|\mathbf{y})$, score-based generative models allow conditional sampling without explicit training of the conditional generative model. Applying the Bayesian rule $p_t(\mathbf{x}|\mathbf{y}) = \frac{p_t(\mathbf{y}|\mathbf{x})p_t(\mathbf{x})}{p(\mathbf{y})}$ one can observe that:

$$\nabla_{\mathbf{x}} \ln p_t(\mathbf{x}|\mathbf{y}) = \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \ln p_t(\mathbf{y}|\mathbf{x})$$

It follows that one can obtain the conditional score from the unconditional score by separately training a noise-dependent classifier $p_t(\mathbf{y}|\mathbf{x})$. This technique is known as "guidance", and it has been used for class conditional image generation (Dhariwal and Nichol 2021).

3 Related Work

A method for training-free controllable generation with score-based models is discussed in Song et al. (2021) and applied to conditional generation tasks such as inpainting and colorization. Still, the estimate is applicable only assuming the possibility to define $\mathbf{y}_t$ such that $p(\mathbf{y}_t|\mathbf{y})$ and $p(\mathbf{x}_t|\mathbf{y}_t)$ are tractable. Earlier works explored the combination of unconditional latent variables generative models such as Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) or Variational Autoencoders (VAEs) (Kingma and Welling 2013) with constraints to produce conditional samples (Engel, Hoffman, and Roberts 2017). As for diffusion models, recent research has focused on leveraging pre-trained unconditional models as priors for solving inverse problems as in Kadkhodaie and Simoncelli (2021), Graikos et al. (2022), Bansal et al. (2023) andNair et al. (2023) where they generalize to a generic guidance. Despite these works focused on sampling high-quality samples that satisfy some given properties, it was not verified if samples followed the correct conditional distributions, a more difficult task that is relevant for example when dealing with tabular data and time series. Given that these methods are often based on the introduction of an optimization phase in the original sampling process, it is not guaranteed that the true conditional distribution will be well approximated. Moreover, these methods are often fitted for imaging problems, and were not tested on different kinds of data and constraints.

Despite the existence of many related prior works with different focuses, our goal is different and more challenging: obtaining samples distributed according to the target conditional distribution, while previous methods rather aim at obtaining high quality samples. To our knowledge, there are no works focusing on the correct approximation of the conditional distribution for tabular data as we do. The best comparison we can do is with Bansal et al. (2023), since it is the state of the art for diffusion based zero-shot conditional generation and a synthesis of previous techniques. We show that our method is significantly better with tabular data when the objective is the approximation of the conditional distribution.

4 Method

Problem Formalization

Given a set of observed samples $\mathbf{x}_i \in \mathbb{R}^d$, the goal is to sample from the distribution $p(\mathbf{x})$ that generated $\mathbf{x}_i$ conditioning on a desired property. Let $\pi(\mathbf{x}) : \mathbb{R}^d \to \{0, 1\}$ be the function that encodes this property, such that $\pi(\mathbf{x}) = 1$ when the property is satisfied and $\pi(\mathbf{x}) = 0$ otherwise. Then the target conditional distribution can be defined as:

$$p(\mathbf{x}|\pi) \propto p(\mathbf{x})\pi(\mathbf{x})$$

Alternatively, one can also define soft constraints, expressing the degree of satisfaction as a real number. Let $c(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function expressing this soft constraint. In this case we define the target distribution as:

$$p^c(\mathbf{x}) \propto p(\mathbf{x})e^{c(\mathbf{x})}$$

Moreover, since the form is analogous to the previous formulation, given a hard constraint $\pi(\mathbf{x})$ one can build a soft constraint $c(\mathbf{x})$ such that $p^c(\mathbf{x}) \approx p(\mathbf{x}|\pi)$. We then consider $p^c(\mathbf{x})$ as the target distribution we want to sample from.

Constraint-Based Guidance

Our method exploits score-based generative models as the base generative model. As previously introduced, a stochastic process that gradually adds noise to original data $q(\tilde{\mathbf{x}}|\mathbf{x})$ is defined such that at $t = 0$ no noise is added so $X_0 \sim p(\mathbf{x})$ and at $t = 1$ the maximum amount of noise is added such that $X_1 \sim q_1(\tilde{\mathbf{x}}|\mathbf{x})$ is a known prior distribution (for example a Gaussian). Given the possibility to efficiently sample from $p_t(\mathbf{x})$, the time-dependent (*Stein*) score of $p_t(\mathbf{x})$ is estimated by score matching using a neural network, let it be $\mathbf{s}(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$. As discussed in the previous section, there are different possible sampling schemes once the score is available. Given the target distribution:

$$p^c(\mathbf{x}) := \frac{p(\mathbf{x})e^{c(\mathbf{x})}}{Z}$$

where Z is the unknown normalization constant, and the distribution of samples from $p^c(\mathbf{x})$ that are successively corrupted by $q(\tilde{\mathbf{x}}|\mathbf{x})$:

$$p_t^c(\mathbf{x}) := \int q_t(\mathbf{x}|\mathbf{x}_0)p^c(\mathbf{x}_0)d\mathbf{x}_0$$

we observe the following relationship:

$$\nabla_{\mathbf{x}} \ln p_0^c(\mathbf{x}) = \nabla_{\mathbf{x}} \ln p^c(\mathbf{x}) = \nabla_{\mathbf{x}} \ln \frac{p(\mathbf{x})e^{c(\mathbf{x})}}{Z}$$

$$= \nabla_{\mathbf{x}}[\ln p(\mathbf{x}) + c(\mathbf{x}) - \ln Z] = \nabla_{\mathbf{x}} \ln p(\mathbf{x}) + \nabla_{\mathbf{x}} c(\mathbf{x})$$

It follows that at $t = 0$ one can easily obtain an estimate of the score by summing the gradient of the constraint to the estimate of the score of the unconstrained distribution. Notice that this is possible since taking the gradient of the logarithm eliminates the intractable integration constant Z. At $t = 1$ instead one can assume $\nabla_{\mathbf{x}} \ln p_1^c(\mathbf{x}) = \nabla_{\mathbf{x}} \ln p_1(\mathbf{x})$, since it is reasonable to assume enough noise is added to make samples from $p_1^c(\mathbf{x})$ distributed as the prior. In general there is no analytical form for $\nabla_{\mathbf{x}} \ln p_t^c(\mathbf{x})$, also, it cannot be estimated by score matching since we are not assuming samples from $p_0^c(\mathbf{x})$ are available in the first place.

Conditional Score Approximation

Given this limit, we resort to approximations $\tilde{\mathbf{s}}_c(\mathbf{x}, t)$ for $\mathbf{s}_c(\mathbf{x}, t) = \nabla_{\mathbf{x}} \ln p_t^c(\mathbf{x})$. The approximations we use are constructed knowing the true value of the score for $t = 0$ and $t = 1$:

$$\tilde{\mathbf{s}}_c(\mathbf{x}, 0) = \mathbf{s}(\mathbf{x}, 0) + \nabla_{\mathbf{x}} c(\mathbf{x}) \quad (1)$$

$$\tilde{\mathbf{s}}_c(\mathbf{x}, 1) = \mathbf{s}(\mathbf{x}, 1) \quad (2)$$

A simple way to obtain this is by weighting the contribution of the gradient of the constraint depending on time:

$$\tilde{\mathbf{s}}_c(\mathbf{x}, t) = \mathbf{s}(\mathbf{x}, t) + g(t)\nabla_{\mathbf{x}} c(\mathbf{x})$$

where $g(t) : [0, 1] \to [0, 1]$ satisfies $g(0) = 1$ and $g(1) = 0$. This is equivalent to extending the domain of the constraint to noisy data points $c(\mathbf{x}, t)$ and then approximating it with $c(\mathbf{x}, t) = g(t)c(\mathbf{x})$. Sampling from the target distribution then reduces to substituting the score of the base model with the modified score $\tilde{\mathbf{s}}_c(\mathbf{x}, t)$. Notice that this approach does not require any re-training of the model. The only necessary ingredients are the unconditional score model $\mathbf{s}(\mathbf{x}, t)$ and the differentiable constraint $c(\mathbf{x})$ encoding the degree of satisfaction of the desired property.

Multiple Instances Constraints We may also want to sample multiple instances $\mathbf{v} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ that are tied together by a single multivariate constraint $c(\mathbf{v})$, i.e., sampling from: $p^c(\mathbf{v}) \propto p(\mathbf{v})e^{c(\mathbf{v})} = e^{c(\mathbf{v})} \prod_{i=1}^n p(\mathbf{x}_i)$. In this case, it is easy to show that $\nabla_{\mathbf{x}_i} \ln p_0^c(\mathbf{x}_i) = \nabla_{\mathbf{x}_i} \ln p(\mathbf{x}_i) + \nabla_{\mathbf{x}_i} c(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, then the approximated score for each instance $\mathbf{x}_i$, $i \in \{1, \ldots, n\}$ is:

$$\tilde{\mathbf{s}}_c(\mathbf{x}_i, t) = \mathbf{s}(\mathbf{x}_i, t) + g(t)\nabla_{\mathbf{x}_i} c(\mathbf{x}_1, \ldots, \mathbf{x}_n)$$

This can be computed in parallel for each instance $\mathbf{x}_i$, as the computation of $\mathbf{s}(\mathbf{x}_i, t)$ can be parallelized by batching the score network and $\nabla_{\mathbf{x}_i} c(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is just a component of $\nabla_{\mathbf{v}} c(\mathbf{v})$. When sampling, the instances will also be generated in parallel, as they were a single instance.

Langevin MCMC Correction. Depending on the type of data, we found it useful to perform additional Langevin dynamics steps (these are referred to as "corrector steps" in Song et al. (2021)) at time $t = 0$ when the score of the constrained target distribution is known without approximation. Langevin dynamics can be used as a Monte Carlo method for sampling from a distribution when only the score is known (Welling and Teh 2011; Parisi 1981), performing the following update, where ϵ is the step size and $\mathbf{z}^i$ is sampled from a standard normal with the dimensionality of $\mathbf{x}$:

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \epsilon\, \tilde{\mathbf{s}}_c(\mathbf{x}, 0) + \mathbf{z}^i\sqrt{2\epsilon}$$

In the limit $i \to \infty$ and $\epsilon \to 0$ Langevin dynamics samples from $p^c(\mathbf{x})$. Nevertheless, as most Monte Carlo techniques, Langevin dynamics struggles in exploring all the modes when the target distribution is highly multimodal. Algorithm 1 summarizes the modified sampling algorithm.

Choice of Score Approximation Scheme. One should choose a $g(t)$ that is strong enough to guide samples towards the modes of $p_0^c(\mathbf{x})$ but at the same time that does not disrupt the reverse diffusion process in the early steps. We experimented with various forms of $g(t)$, mostly with the following two functions:

- **Linear**: $g(t) = 1 - t$
- **SNR**: $g(t)$ is equal to the signal-to-noise ratio of the diffusion kernel. For example, if the diffusion kernel $q_t(\tilde{\mathbf{x}}|\mathbf{x})$ is $\mathcal{N}(\mathbf{x}, \sigma_t)$, then $g(t) = (1 + \sigma_t^2)^{-\frac{1}{2}}$, assuming normalized data.

In many of our experiments we found **SNR** to be the most effective, so we suggest using it as the first choice.

Algorithm 1: Constraint guidance sampling

Input: constraint $c(\mathbf{x})$, score $\mathbf{s}(\mathbf{x}, t)$, score-based sampling algorithm $A(\mathbf{s})$
Parameters: $g(t)$, ϵ, n

$\tilde{\mathbf{s}}_c(\mathbf{x}, t) \leftarrow \mathbf{s}(\mathbf{x}, t) + g(t)\nabla_{\mathbf{x}} c(\mathbf{x})$
$\mathbf{x} \leftarrow A(\tilde{\mathbf{s}}_c)$
for $i = 1$ **to** n **do**
  $\mathbf{z} \leftarrow \mathcal{N}(0, 1)$ (with the dimensionality of $\mathbf{x}$)
  $\mathbf{x} \leftarrow \mathbf{x} + \epsilon\tilde{\mathbf{s}}_c(\mathbf{x}, 0) + \mathbf{z}\sqrt{2\epsilon}$
end for
return $\mathbf{x}$

Neuro-Symbolic Logical Constraints

We will consider a general class of constraints expressed in a logical form. Hard logical constraints cannot be directly used in the approach presented above, hence we turn them into differentiable soft constraints leveraging neuro-symbolic ideas (Badreddine et al. 2020). More specifically, we consider predicates and formulae defined on the individual features $\mathbf{x} = (x_1, \ldots, x_d)$ of the data points we ought to generate. Given a Boolean property $P(\mathbf{x})$ (i.e. a predicate or a formula), with features $\mathbf{x}$ as free variables, we associate with it a constraint function $c(\mathbf{x})$ such that $e^{c(\mathbf{x})}$ approximates the corresponding non-differentiable hard constraint $\mathbf{1}_{P(\mathbf{x})}$.[1] In this paper, we consider constraints that can be evaluated on a single or on a few data points, hence we can restrict ourselves to the quantifier-free fragment of first-order logic. Therefore, we can define by structural recursion the constraint $c(\mathbf{x})$ for atomic propositions and Boolean connectives. As atomic propositions, we consider here simple equalities and inequalities of the form $a(\mathbf{x}) \geq b(\mathbf{x})$, $a(\mathbf{x}) \leq b(\mathbf{x})$, $a(\mathbf{x}) = b(\mathbf{x})$, where a and b can be arbitrary differentiable functions of feature variables $\mathbf{x}$. Following Badreddine et al. (2020), we refer to such sets of functions as *real logic*.

In particular, we define the semantics directly in log-probability space, obtaining the **Log-probabilistic logic**. The definitions we adopt are partially in line with those of the newly defined *LogLTN*, see Badreddine, Serafini, and Spranger (2023), and are reported in Table 1.

Formula	Differentiable function
$c[a(\mathbf{x}) \geq b(\mathbf{x})]$	$-\ln(1 + e^{-k(a(\mathbf{x})-b(\mathbf{x}))})$
$c[a(\mathbf{x}) \leq b(\mathbf{x})]$	$-\ln(1 + e^{-k(b(\mathbf{x})-a(\mathbf{x}))})$
$c[a(\mathbf{x}) = b(\mathbf{x})]$	$a(\mathbf{x}) \geq b(\mathbf{x}) \wedge a(\mathbf{x}) \leq b(\mathbf{x})$
$c[\varphi_1 \wedge \varphi_2]$	$c[\varphi_1] + c[\varphi_2]$
$c[\varphi_1 \vee \varphi_2]$	$\ln(e^{c[\varphi_1]} + e^{c[\varphi_2]} - e^{c[\varphi_1]+c[\varphi_2]})$
$c[\neg\varphi]$	$\ln(1 - e^{c[\varphi]})$

Table 1: Semantic rules of log-probabilistic logic. In the table, $c[\varphi](\mathbf{x})$ is the soft constraint associated with the formula φ.

[1] $\mathbf{1}_{P(\mathbf{x})}$ is indicator function equal to 1 for each $\mathbf{x}$ such that $P(\mathbf{x})$ is true.

Atomic Predicates. We choose to define the inequality $a(\mathbf{x}) \geq b(\mathbf{x})$ as $c(\mathbf{x}) = -\ln(1 + e^{-k(a(\mathbf{x})-b(\mathbf{x}))})$, introducing an extra parameter k that regulates the "hardness" of the constraint. Indeed, in the limit $k \to \infty$ one have $\lim_{k\to\infty} e^{c(\mathbf{x})} = \mathbf{1}_{a(\mathbf{x})\geq b(\mathbf{x})}$. This definition is consistent with the negation but its gradient is nonzero when the condition is satisfied,[2] though this was not creating issues in the experiments for sufficiently large values of k. For the equality we use the standard definition based on inequalities. We also experimented with a definition based on the l2 distance, corresponding to a Gaussian kernel, even if this form does not benefit from a limited gradient.

Boolean Connectives. The conjunction and the disjunction correspond to the product t-norm and its dual t-conorm (probabilistic sum) (van Krieken, Acar, and van Harmelen 2022) but in logarithmic space. We use the material implication rule to reduce the logical implication to a disjunction: $a \to b \equiv \neg a \vee b$. The negation, instead, is consistent with the semantic definition of inequalities: negating one inequality, one obtains its flipped version. For numerical stability reasons, however, we choose to avoid using the soft negation function in any case. Instead, we reduce logical formulas to the negation normal form (NNF) as in Badreddine, Serafini, and Spranger (2023), where negation is only applied to atoms, for which the negation can be computed analytically or imposed by definition. In order to simplify the notation, in the following we will also use quantifiers ($\forall$ and $\exists$) as syntactic sugar (in place of *finite* conjunctions or disjunctions) only when the quantified variable takes values in a finite and known domain (e.g. time instants in a time series or pixels in an image). So $\forall i \in \{1, \ldots, n\}\ p_i$ is used as a shorthand for $p_1 \wedge p_2 \wedge \ldots \wedge p_n$ and $\exists i \in \{1, \ldots, n\} : p_i$ for $p_1 \vee p_2 \vee \ldots \vee p_n$.

The difference in our definition with respect to *LogLTN* is in the logical disjunction ($\vee$): they define it using the Log-MeanExp (LME) operator, an approximation of the maximum that is numerically stable and suitable for derivation. They do it at the price of losing the possibility to reduce formulas to the NNF exactly (using De Morgan's laws) that follows from having as disjunction the dual t-conorm of the conjunction. We choose instead to use the log-probabilistic sum as the disjunction, since in the domain of our experiments it proved numerically stable and effective.

When sampling, we can regulate the tradeoff between similarity with the original distribution and strength of the constraint by tuning the parameter k of inequalities in log-probabilistic logic. Alternatively, we can multiply by a constant λ the value of the constraint in order to scale its gradient.

Training a Score Model for Tabular Data

Score-based models have been mainly used for image generation, adapting them to tabular data and time series requires special care. In particular, the challenge is to correctly model the noise of the target distribution, implying the tricky task

[2] This can be addressed by defining a simplified version: $a(\mathbf{x}) \geq b(\mathbf{x}) \equiv k(a(\mathbf{x}) - b(\mathbf{x}))\mathbf{1}_{a(\mathbf{x})<b(\mathbf{x})}$, however such a definition will no more be consistent with negation.

of estimating the score at $t \approx 0$ (no noise). We improved the score estimate mainly by parametrizing carefully the score network and using large batches.

Correctly estimating the score at $t \approx 0$ is fundamental to make our method work in practice, since it allows us to perform Langevin MCMC at $t \approx 0$, where the conditional score is known without approximation. Combining a correct score estimation at $t \approx 0$ with many steps of Langevin MCMC allows us to (asymptotically) sample from the exact conditional distribution, in particular when the data distribution is not particularly multimodal.

5 Experiments

We tested our method on several datasets, still, evaluating the quality of conditionally generated samples is challenging. First of all, one should compare conditionally generated samples with another method to generate conditionally in an exact way. We chose then to compare our approach with rejection sampling (RS), that can be used to sample exactly from the product of two distributions $p(\mathbf{x})$ and $q(\mathbf{x})$, where sampling from $p(\mathbf{x})$ is tractable and $q(\mathbf{x})$ density is known up to a normalization constant. In our case $p(\mathbf{x})$ is the unconditional generative model and $q(\mathbf{x}) = e^{c(\mathbf{x})}$. Assuming constraints are such that $\forall \mathbf{x}\, c(\mathbf{x}) \leq 0$, then $q(\mathbf{x})$ is upper-bounded by 1. This upper bound is guaranteed by the real logic we defined previously. RS then reduces to sampling from $p(\mathbf{x})$ and accepting each sample with probability $q(\mathbf{x})$. This can be problematic when the probability of a random sample from $p(\mathbf{x})$ having a non-null value of $q(\mathbf{x})$ is low. In the second place, comparing the similarity of two samples is a notoriously difficult problem. For relatively low dimensional samples, we will compare the marginal distributions and the correlation matrix. For comparing one-dimensional distributions among two samples X and Y we use the l1 histogram distance $D(X, Y) := \frac{1}{2}\sum_i |x_i - y_i|$ where x_i and y_i are the empirical probabilities for a given common binning. This distance is upper bounded by 1. When computationally feasible, we consider RS as the baseline method. By reporting the acceptance rate of RS, we show the satisfaction rate of the constraint on data generated by the original model. Moreover, we discuss in Section 5 a comparison with Bansal et al. (2023) that is arguably the state of the art method for zero-shot conditional generation of images.

We mostly used unconditional models based on denoising score matching and SDEs, following closely Song et al. (2021).

Tabular Data

We made experiments with the white wine table of the popular UCI Wine Quality dataset (Paulo et al. 2009), consisting of 11 real-valued dimensions ($\mathbb{R}^{11}$) and one discrete dimension, the quality, that we discarded. In order to evaluate the effectiveness with categorical variables, we also made experiments with the Adult dataset (Becker and Kohavi 1996), consisting of 5 numerical dimensions and 10 categorical dimensions, which we embedded in a continuous space using one-hot encodings. We fitted unconditional score-based diffusion models based on SDEs, then we generated samples under illustrative logical constraints.

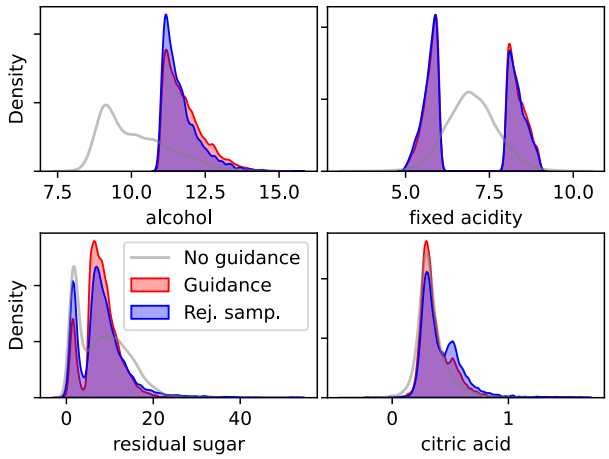

Figure 1: Marginals of white wine data experiment. We generated 5000 samples using our constrained sampling algorithm and as many by RS. The plot compares the marginals of the dimensions directly involved in the constraint. The last two dimensions are the ones with the largest l1 histogram distance with respect to RS marginals: 0.15 and 0.13. While the median distance across all dimensions is ≈ 0.1.

First, we generated samples from the white wine model under the following complex logical constraint: (fixed acidity $\in [5.0, 6.0] \vee$ fixed acidity $\in [8.0, 9.0]) \wedge$ alcohol $\geq 11.0 \wedge$ (residual sugar $\leq 5.0 \rightarrow$ citric acid ≥ 0.5). We show in Figure 1 the marginals of the generated samples, compared with samples generated by RS. There is a high overlap between marginals for most dimensions, and we measured an average l1 distance between correlation coefficients of ≈ 0.07. The largest error, that is associated with one of the dimensions heavily affected by the constraint, is still relatively small. For that constraint, the acceptance rate of RS was only $\approx 1.67\%$, meaning that our method samples efficiently in low-probability regions. The satisfaction rates of the relative hard constraint were similar: $\approx 92\%$ for RS and $\approx 86\%$ for our method (these can be increased by increasing the parameter k). Additionally, we tested the application of a simple multi-instance constraint acting on pair of data points $\mathbf{x}_1$ and $\mathbf{x}_2$: alcohol$_1$ > alcohol$_2$ + 1. Comparing again with RS, we achieved a negligible error and the relative hard constraint was met in 99% of the generated samples, compared to the 100% of RS, with an acceptance rate of 27%. We further confirmed the effectiveness of our method by generating samples from the Adult model under the following logical constraint: age $\geq 40 \wedge$ (race $\neq$ "White" $\vee$ education $=$ "Masters"). In this case equalities and inequalities involving discrete components are obtained by imposing the desired component of the corresponding one-hot encoding equal to 1 for equality, or to 0 for inequality. The median l1 histogram distance was ≈ 0.05, the maximum was ≈ 0.13 and the error in correlations was negligible. The relative hard constraint was met in all samples while the acceptance rate of RS was 1.73%.

Time Series Surrogate Models

A surrogate model is a simplified and efficient representation of a complex, eventually computationally expensive model. It is possible to learn a surrogate model of a complex stochastic dynamical system by fitting a statistical model to a dataset of trajectories observed from it. Following our approach, one can use a score-based generative model to learn an unconditional surrogate model, and then apply constraints to enforce desirable properties. These can be physical constraints the system is known to respect, or features that are rare in unconditioned samples. So we can exploit this method to both assure consistency of trajectories and explore rare (but not necessarily with low density) scenarios. As a case study, we apply our proposed method for the conditional generation of ergodic SIRS (eSIRS) trajectories. The eSIRS model (1927) is widely used to model the spreading of a disease in an open population[3]. The model assumes a fixed population of size N composed of Susceptible (S), Infected (I), and Recovered (R) individuals. We consider trajectories with H discretized time steps, thus we have that the sample space is $\mathcal{X}_{\textbf{eSIRS}} := \left(\mathbb{N}_0{}^2\right)^H$, where the two dimensions are S and I (R is implicit since $R = N - S - I$).

First we train a score-based generative model to fit trajectories that were generated by a simulator, with $H = 30$ and $N = 100$. Then we experimented with the application of different constraints, including the following consistency constraints:

- *Non-negative populations*: $\forall t\ S(t) \geq 0 \wedge I(t) \geq 0$
- *Constant population*: $\forall t\ S(t) + I(t) \leq N$

We show in Figure 2 and Figure 3 two experiments with two different constraints. In both experiments the consistency constraints (positive and constant population) were always met, with a small improvement over the unconditional model. In the two experiments we additionally imposed a bridging constraint and an inequality, that were also met with minimal error.

Images

We test our method on image datasets in order to investigate the potential in high-dimensional data. We consider satisfactory validating the results by visual inspection of the generated images, since in this case the quality of individual samples is often considered more important than matching the true underlying distribution. We do not use classifier-based metrics such as FID or Inception score since data samples from the original conditioned distribution are not available, hence a comparison is not possible.

We use as pre-trained unconditional models a model based on a U-net that we trained on the MNIST dataset, and a pre-trained model for CelebA (Liu et al. 2015) 64x64 images made available in Song and Ermon (2020).

Digits sum. Using a pre-trained MNIST classifier, we define a multi-instance constraint that forces pairs of mnist digits to sum up to ten. Given pairs of images $(\mathbf{x}, \mathbf{y})$, we define the constraint in the following way:

$$\bigvee_{i=1}^{9} \text{class}(\mathbf{x}, i) \wedge \text{class}(\mathbf{y}, 10 - i)$$

where $\text{class}(\mathbf{x}, i) := P\{\mathbf{x} \text{ is classified as } i\} = 1$, and P is obtained from a pre-trained classifier.

The generated pairs of digits add up to ten in $\approx 96\%$ of cases[4], however, only 2-8 and 4-6 pairs were generated.

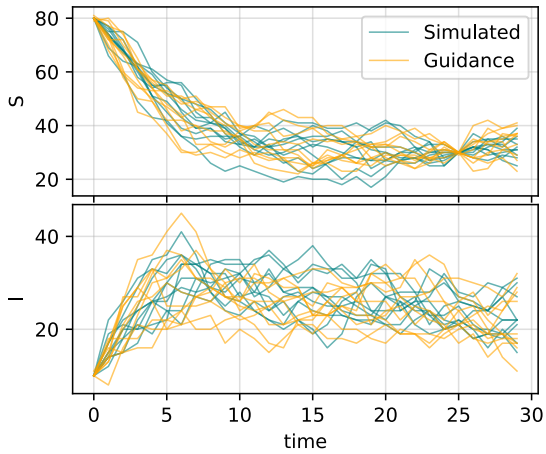

Figure 2: Bridging with eSIRS time series. We show here a subsample of the 5000 time series generated with constraint guidance (orange) with a subsample of the > 50000 time series generated by RS from the simulator (green). Additionally to the consistency constraints, that are always met, we imposed the following equalities: $S(0) = 95$, $I(0) = 5$, $S(25) = 30$. Constraints are generally met: the average l1 absolute difference with all three target values is below 0.19. The l1 histogram distance for each time step marginal is relatively small, considering that it accounts also for the error of the unconditional model: for S and I the median l1 histogram distance across time are ≈ 0.11 and ≈ 0.13.

Restoration. Given a differentiable function $f(\cdot)$, that represents a corruption process in which information is lost, we define the following constraint:

$$\forall i\ f(\mathbf{x})_i = \tilde{\mathbf{y}}_i$$

where i is the pixel index and $\tilde{\mathbf{y}}$ is a corrupted sample, possibly such that there is a $\mathbf{y}$ that satisfies $\forall i\ \tilde{\mathbf{y}}_i \approx f(\mathbf{y})_i$. Such constraint has the effect of sampling possible $\mathbf{x}$ such that $\forall i\ f(\mathbf{x})_i = \tilde{\mathbf{y}}_i$, i.e., "inverting" f or reconstructing the original $\mathbf{y}$. f can be any degradation process, such as downsampling, blurring or adding noise. So our approach can be used for image restoration for a known degradation process.

In Figure 4 we show the results of image restoration experiments with a blurred and a downsampled image. Samples are realistic and we report a low error with respect to the target corrupted image (the absolute error per channel is approximately less than 0.015).

[3]Open in the sense of having infective contacts with external individuals, not part of the modelled population.

[4]according to classes assigned by the classifier

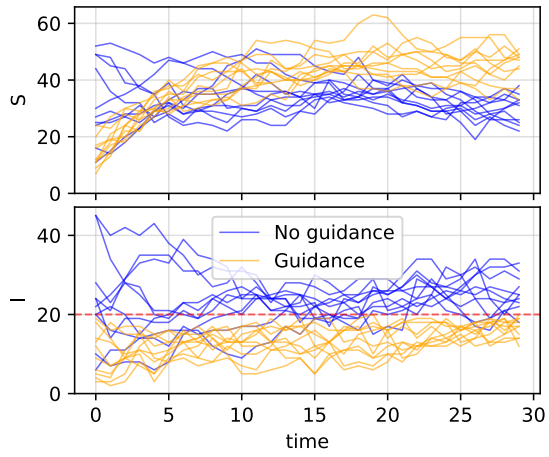

Figure 3: Imposing an inequality on eSIRS time series. We show here a subsample of the 100 time series generated with and without constraint guidance. Additionally to the consistency constraints, that are always met, we imposed $\forall t\ I(t) \leq 20$, that is perfectly met in 99% of the samples.

Figure 4: Restoration experiments with CelebA. The first image on the left is a sample image from the CelebA dataset. Each row shows the corrupted image followed by samples generated imposing the restoration constraint.

In general, we observed the effectiveness of our method to depend on the task. For most experiments involving tabular data, keeping a large constraint strength k (as $k = 30$) was sufficient. However, for some experiments, as most tasks involving images, we needed to tune the constraint strength until a satisfactory trade-off between constraint satisfaction and quality of samples was reached.

Comparison with Universal Guidance

We compared our conditioning method with Universal Guidance, introduced in Bansal et al. (2023). Universal Guidance was successfully used in the context of image generation and it is arguably the state of the art for zero-shot conditional generation. Their method is based on three improvements over the standard guidance technique, that they call *forward universal guidance*, *backward universal guidance* and *per-step self-recurrence*. *Forward universal guidance* consists in using the gradient of the constraint with respect to the predicted clean data point $\hat{\mathbf{x}}_0$ (prediction based on the trained denoising net), instead of the current data point $\mathbf{x}_t$. *Backward universal guidance* involves an optimization of $\hat{\mathbf{x}}_0$ according to the constraint and modifying the score used for the reverse process as a consequence. Applying these techniques on tabular data and time series we observed little effect of *forward universal guidance*, and a significantly detrimental effect of *backward universal guidance*. Metrics reported in Table 2 show the superiority of our method for two of the previously discussed tasks. We think this significant difference is due to the fact that Universal Guidance, as most recent plug-and-play techniques, modify the sampling algorithm by introducing an optimization phase.

Method	White wine		eSIRS bridging	
	Avg L1	**Max L1**	**Avg L1**	**Max L1**
Ours	**0.1**	**0.15**	**0.13**	**0.25**
Univ. Guidance	0.29	0.85	0.47	1.0

Table 2: Comparison with Universal Guidance. On the first white wine and eSIRS bridging tasks our method performs significantly better in terms of l1 histogram distance (we report average and maximum distance over the marginals).

On the other side, we noticed a sensible improvement in some of our image-based conditional generation tasks. For example, relatively to the MNIST digits' sum experiment, universal guidance allowed us to generate all possible pairs, that always summed up to ten. We infer that universal guidance can be useful when the objective is to generate high-quality samples that satisfy a given constraint, but it can lead to large biases when modelling the conditional distribution, that is of primary interest in some contexts, as with tabular data and time series. Moreover, with these experiments we verified that our *Log-probabilistic logic* is effective also within other guidance techniques.

6 Conclusion

We have shown how we can exploit pre-trained unconditional score-based generative models to sample under user-defined logical constraints, without the need for additional training. Our experiments demonstrate the effectiveness in several contexts, such as tabular and high-dimensional data. Nevertheless, in some high dimensional settings as images, we had to trade-off between sample quality and constraint satisfaction by tuning the constraint strength. More sophisticated methods specifically designed for images are probably necessary, still these could fail to model the true conditional distribution. In fact, we show the superiority of our method in approximating conditional distributions for tabular and time-series data with respect to a state of the art method for zero-shot conditioning. Future work will aim at finding better approximation schemes, starting from works focused on general plug-and-play guidance for images.

For a deeper discussion about the experiments, hyperparameters and other technical details we refer readers to the extended version of this article (Scassola et al. 2024).

Acknowledgements

This research was supported by Aindo, which has funded the PhD of the first author and provided the computational resources. Additionally, this study was carried out within the PNRR research activities of the consortium iNEST funded by the European Union Next-GenerationEU (PNRR, Missione 4 Componente 2, Investimento 1.5 – D.D. 1058 23/06/2022, ECS_00000043).

References

Badreddine, S.; d'Avila Garcez, A. S.; Serafini, L.; and Spranger, M. 2020. Logic Tensor Networks. *CoRR*, abs/2012.13635.

Badreddine, S.; Serafini, L.; and Spranger, M. 2023. logLTN: Differentiable Fuzzy Logic in the Logarithm Space. *ArXiv*, abs/2306.14546.

Bansal, A.; Chu, H.-M.; Schwarzschild, A.; Sengupta, S.; Goldblum, M.; Geiping, J.; and Goldstein, T. 2023. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 843–852.

Becker, B.; and Kohavi, R. 1996. UCI Machine Learning Repository: Adult Dataset. https://doi.org/10.24432/C5XW20. Accessed: 2024-08-08.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34: 8780–8794.

Engel, J.; Hoffman, M.; and Roberts, A. 2017. Latent constraints: Learning to generate conditionally from unconditional generative models. *arXiv preprint arXiv:1711.05772*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Graikos, A.; Malkin, N.; Jojic, N.; and Samaras, D. 2022. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35: 14715–14728.

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

Hyvärinen, A.; and Dayan, P. 2005. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).

Kadkhodaie, Z.; and Simoncelli, E. 2021. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Information Processing Systems*, 34: 13242–13254.

Kermack, W. O.; and McKendrick, A. G. 1927. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772): 700–721.

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

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, 3730–3738.

Nair, N. G.; Cherian, A.; Lohit, S.; Wang, Y.; Koike-Akino, T.; Patel, V. M.; and Marks, T. K. 2023. Steered Diffusion: A Generalized Framework for Plug-and-Play Conditional Image Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 20850–20860.

Parisi, G. 1981. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3): 378–384.

Paulo, C.; A., C.; F., A.; T., M.; and J., R. 2009. UCI Machine Learning Repository: Wine Quality Dataset. https://doi.org/10.24432/C56S3T. Accessed: 2024-08-08.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Scassola, D.; Saccani, S.; Carbone, G.; and Bortolussi, L. 2024. Zero-Shot Conditioning of Score-Based Diffusion Models by Neuro-Symbolic Constraints. arXiv:2308.16534.

Sohl-Dickstein, J.; Weiss, E. A.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *CoRR*, abs/1503.03585.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y.; and Ermon, S. 2020. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33: 12438–12448.

Song, Y.; Garg, S.; Shi, J.; and Ermon, S. 2020. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, 574–584. PMLR.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

van Krieken, E.; Acar, E.; and van Harmelen, F. 2022. Analyzing differentiable fuzzy logic operators. *Artificial Intelligence*, 302: 103602.

Vincent, P. 2011. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674.

Welling, M.; and Teh, Y. W. 2011. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *International Conference on Machine Learning*.