# Conservative objective models are a special kind of contrastive divergence-based energy model

**Christopher Beckham** [1 2] **Christopher Pal** [1 2 3 4]

## Abstract

In this work we theoretically show that conservative objective models (COMs) for offline model-based optimisation (MBO) are a special kind of contrastive divergence-based energy model, one where the energy function represents both the unconditional probability of the input and the conditional probability of the reward variable. While the initial formulation only samples modes from its learned distribution, we propose a simple fix that replaces its gradient ascent sampler with a Langevin MCMC sampler. This gives rise to a special probabilistic model where the probability of sampling an input is proportional to its predicted reward. Lastly, we show that better samples can be obtained if the model is decoupled so that the unconditional and conditional probabilities are modelled separately.

## 1. Introduction

Model-based optimisation (MBO) is concerned with the use of generative models for design problems, where the input $\boldsymbol{x}$ specifies the design and the desirability of any design (i.e. the reward) is a black box function $y = f(\boldsymbol{x})$ called the *ground truth oracle* which is prohibitively expensive to evaluate. For instance, if we are dealing with designing drugs to target disease then the oracle is a real world process that involves synthesising and testing the drug in a wet lab, which is expensive. Because each evaluation of the 'real world' $f(\boldsymbol{x})$ is expensive, we would like to use machine learning to construct a reliable proxy of the oracle $f_\theta(\boldsymbol{x})$ and exploit that instead. (This is one discernible difference to more traditional derivative-free black box optimisation, which assumes that $f$ can be queried at will.) In addition, we are also interested in *extrapolation*: we would like to find designs $\boldsymbol{x}$ that have as high of a reward as possible, possibly even higher than what has been observed so far. Generally speaking, we would like to generate a candidate set $\mathcal{S} = \{\boldsymbol{x}_j\}_{j=1}^M$ such that:

$$\mathcal{S} = \text{argmax}_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_M} \sum_{j=1}^{M} f(\boldsymbol{x}_j). \tag{1}$$

Since we do not have access to $f$ during training, we must resort to training an approximation of it, which we denote $f_\theta(\boldsymbol{x})$. This is usually called an *approximate* or *surrogate* model or 'oracle'. We note that because $f_\theta(\boldsymbol{x})$ is approximate and is a discriminative model, it is vulnerable to over-scoring inputs or even assigning rewards greater than zero to implausible inputs,[1] and these are commonly referred to as *adversarial examples*. In the context of our aforementioned drug example, an implausible input would be a drug whose chemical configuration (that is, its configuration of atoms) is physically impossible. How these problems are addressed depends on whether one approaches MBO from a discriminative modelling point of view (Fu & Levine, 2021; Trabucco et al., 2021) or a generative modelling one (Brookes et al., 2019; Fannjiang & Listgarten, 2020; Kumar & Levine, 2020; Beckham et al., 2022). In this work we will exclusively discuss *conservative objective models* (Trabucco et al., 2021), whose work comes from the discriminative perspective. However, we will show that this model essentially falls under a particular class of generative model called an *energy-based model*, and in this work we perform a theoretical and empirical analysis of that model from this perspective.

[1]Mila - Quebec Artificial Intelligence Institute [2]Polytechnique Montreal [3]ServiceNow Research [4]CIFAR AI Chair. Correspondence to: Christopher Beckham <first.last@mila.quebec>.

Preliminary work.

[1]By 'implausible' inputs, we simply mean any input for which $p(\boldsymbol{x}) = 0$.

Lastly, for the sake of clarification, we note that MBO methods can be categorised into whether they are online or offline. In the online case, we assume that the ground truth oracle can be queried during training to obtain additional labels, which essentially becomes active learning. In the offline case, we only assume a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ and must make do with this data to train the best possible proxy model $f_\theta(\boldsymbol{x})$.[2] For the remainder of this paper we will only consider offline MBO, and simply refer to it as MBO unless otherwise stated.

We lay out our contributions in this work as follows:

- We theoretically show that conservative objective models (COMs) are extremely similar to an energy-based model (EBM) that is trained via contrastive divergence, albeit with a modified MCMC sampler that can only sample from the *modes* of its distribution (Section 2.1). This special form of EBM is parameterised such that the *negative energy* of an input $-E_\theta(\boldsymbol{x})$ is equivalent to the predicted reward $f_\theta(\boldsymbol{x})$ of that same input. In other words, the energy is trained to be both predictive of the *likelihood* of an example, as well as its *reward*, with a training hyperparameter $\alpha$ introduced to balance the trade-off between how much model capacity should be allocated between the two. These two components can be seen as inducing a *joint density* $p_\theta(\boldsymbol{x}, y; \alpha) \propto p_\theta(y|\boldsymbol{x})p_\theta(\boldsymbol{x})^\alpha$ over the data.[3]
- COMs uses gradient ascent for its MCMC sampler, which can only sample modes from its distribution. If it is modified to properly sample from the distribution, then the model becomes a special instance of a contrastive divergence EBM, and we call these ***Stochastic COMs*** (Section 2.2). ***Stochastic COMs*** have the special property that the probability of sampling an input is proportional to its predicted reward, i.e. $p_\theta(\boldsymbol{x}) \propto f_\theta(\boldsymbol{x})$. We illustrate the effect of $\alpha$ on a toy spiral dataset in Section 3 as well as visualise generated samples between the both COMs variants.
- We show that COMs fail to generate desirable samples on a simple toy spiral dataset because the same network is being used to parameterise both the likelihood of an example and its score, and this subsequently degrades sample quality. To alleviate this, we propose ***decoupled COMs***, where a separate classifier is trained and its gradients are used at sampling time (Section 2.3). ***Decoupled COMs*** can be thought of as a contrastive divergence-based EBM which leverages an external classifier (regression model) as a form of conditional guidance.

### 1.1. Energy-based generative models

In EBMs we wish to learn a probability distribution without any specific modelling assumptions. This is done by defining the unnormalised probability of an input as the *negative* of an energy $E\theta$ that is parameterised with a nn:

$$p_\theta(\boldsymbol{x}) = \frac{\exp(-E_\theta(\boldsymbol{x}))}{Z_\theta}, \tag{2}$$

where $Z_\theta$ is the (usually intractable) normalising constant, which can be seen as a function of the energy model parameters $\theta$. Ignoring the intractability issue for a brief moment, the log likelihood for one example $\boldsymbol{x}$ can be expressed as:

$$\log p_\theta(\boldsymbol{x}) = -E_\theta(\boldsymbol{x}) - \log \underbrace{\int_{\boldsymbol{x}} \exp(-E_\theta(\boldsymbol{x}))d\boldsymbol{x}}_{Z_\theta}, \tag{3}$$

where we have re-written $Z_\theta$ as an integral. While this seems virtually impossible to handle, an interesting identity from Song & Kingma (2021) says the score of $Z_\theta$ :

$$\nabla_\theta \log Z_\theta = \mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})}[-\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x})]. \tag{4}$$

In other words, the integral can be approximated via Monte Carlo by simply computing the score over each example inside the expected value. This means that we can define a loss $\mathcal{L}_\theta(\boldsymbol{x})$ such that, when we take the gradient of it, it becomes equivalent to $\nabla_\theta \log p_\theta(\boldsymbol{x})$:

$$\begin{aligned} \mathcal{L}_\theta(\boldsymbol{x}) &= -E_\theta(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})} E_\theta(\boldsymbol{x}) \\ \implies \nabla_\theta \mathcal{L}_\theta(\boldsymbol{x}) &= \nabla_\theta \log p_\theta(\boldsymbol{x}) = \nabla_\theta[-E_\theta(\boldsymbol{x})] + \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})}[-\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x})]}_{\text{Eqn. 4}}. \end{aligned} \tag{5}$$

[2] The difference between offline and online MBO pertains to just the training of the generative model. Even with 'offline' MBO, in a real world setting that model still has to be validated against the ground truth oracle by generating novel inputs and scoring them.

[3] Code for this paper will be made available here: https://github.com/christopher-beckham/coms-are-energy-models

It is expensive to approximate $Z_\theta$ term because it requires us to draw samples from the generative model $p_\theta(\boldsymbol{x})$ which is a costly process. For example, we would have to run Langevin MCMC (Neal et al., 2011; Welling & Teh, 2011; Song & Kingma, 2021) by drawing an initial $\boldsymbol{x}_0$ from some simple prior distribution and running the Markov chain for a sufficiently long number of time steps $T$ such that $\boldsymbol{x}_T \approx p_\theta(\boldsymbol{x})$:

$$\begin{aligned}\boldsymbol{x}_{t+1} &:= \boldsymbol{x}_t + \frac{\epsilon_t^2}{2}\nabla_{\boldsymbol{x}_t}\log p_\theta(\boldsymbol{x}_t) + \epsilon \boldsymbol{z}_t \\ &\boxed{= \boldsymbol{x}_t + \frac{\epsilon_t^2}{2}\nabla_{\boldsymbol{x}_t}[-E_\theta(\boldsymbol{x}_t)] + \epsilon_t \boldsymbol{z}_t,}\end{aligned} \tag{6}$$

where $\boldsymbol{z}_t \sim \mathcal{N}(0, \boldsymbol{I})$, and $\epsilon_T \to 0$.

For reasons that will become clear shortly, we prefer to define $p_\theta(\boldsymbol{x})$ more explicitly such that it is obvious that sampling involves an $\boldsymbol{x}_0$ that is drawn from a simple prior distribution (e.g. a Gaussian or uniform distribution), which we will call $p_\pi(\boldsymbol{x})$. That is, in order to sample from our generative model we first sample $\boldsymbol{x}_0 \sim p_\pi(\boldsymbol{x}_0)$ and then run Langevin MCMC on $\boldsymbol{x}_0$, which we can write simply as a sample from the conditional distribution $\boldsymbol{x} \sim p_\theta(\boldsymbol{x}|\boldsymbol{x}_0)$. Both of these distributions define a joint distribution $p_{\theta,\pi}(\boldsymbol{x}, \boldsymbol{x}_0) = p_\theta(\boldsymbol{x}|\boldsymbol{x}_0)p_\pi(\boldsymbol{x}_0)$, and therefore the marginal over $\boldsymbol{x}$ itself can simply be written as:

$$p_{\theta,\pi}(\boldsymbol{x}) = \int_{\boldsymbol{x}_0} p_\theta(\boldsymbol{x}|\boldsymbol{x}_0)p_\pi(\boldsymbol{x}_0)d\boldsymbol{x}_0. \tag{7}$$

Therefore, we can write a more explicit form of Equation 3 that uses $p_{\theta,\pi}$ instead:

$$\begin{aligned}\mathcal{L}_{\theta,\pi}(\boldsymbol{x}) = \log p_{\theta,\pi}(\boldsymbol{x}) &= -E_\theta(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}' \sim p_{\theta,\pi}(\boldsymbol{x})}E_\theta(\boldsymbol{x}') \\ &= -E_\theta(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}' \sim p_\theta(\boldsymbol{x}|\boldsymbol{x}_0), \boldsymbol{x}_0 \sim p_\pi(\boldsymbol{x}_0)}E_\theta(\boldsymbol{x}').\end{aligned} \tag{8}$$

We now explain the reason for this reformulation: a widely known algorithm used to train these models is called *contrastive divergence* (Hinton, 2002), a modification of the Langevin MCMC procedure. Contrastive divergence proposes two modifications to make it more computationally viable: run MCMC for $k$ iterations instead (where $k$ is extremely small, such as a few steps), and let $p_\pi$ be the *actual data distribution*, so the chain is initialised from a real data point. (To keep notation simple, whenever $p_\theta(\boldsymbol{x})$ is used, we really mean $p_{\theta,\pi}(\boldsymbol{x})$ where $p_\pi(\boldsymbol{x}) = p(\boldsymbol{x})$, the real data distribution.) While running the sampling chain for $k$ iterations introduces some bias into the gradient, it appears to work well in practice (Bengio & Delalleau, 2009). Concretely, if we use CD for $k$ iterations then we will write our objective:

$$\begin{aligned}\mathcal{L}_\theta^{CD-k}(\boldsymbol{x}) &:= -E_\theta(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}' \sim p_\theta^k(\boldsymbol{x}'|\boldsymbol{x}_0)p(\boldsymbol{x}_0)}E_\theta(\boldsymbol{x}') \\ &\approx \mathcal{L}_\theta(\boldsymbol{x}).\end{aligned} \tag{9}$$

We will denote this style of energy-based model as a ***contrastive divergence-based EBM***, or simply ***CD-EBM***.

## 2. COMs

Before continuing, we make an important distinction between the approximate oracle $f_\theta(\boldsymbol{x})$ itself and its *statistical* interpretation, $p_\theta(y|\boldsymbol{x})$. The approximate oracle $f_\theta(\boldsymbol{x})$ is a regression model trained to predict $y$ from $\boldsymbol{x}$ but the precise loss function used imbues a specific probabilistic interpretation relating to that model. For instance, if the *mean squared error loss* is used during training, then $p_\theta(y|\boldsymbol{x})$ has the interpretation of being a Gaussian distribution whose $f_\theta(\boldsymbol{x})$ parameterises the mean and $\sigma^2 = 1$. While the choice of probabilistic model is up to the user, we will assume a Gaussian model here as it is the most commonly used for regression tasks and is the probabilistic model used in the paper. Given some training pair $(\boldsymbol{x}, y) \in \mathcal{D}$ we can write out its conditional likelihood:

$$\log p_\theta(y|\boldsymbol{x}) = \log \mathcal{N}(y; f_\theta(\boldsymbol{x}), \sigma) = -\frac{1}{\sigma\sqrt{2\pi}}(y - f_\theta(\boldsymbol{x}))^2, \tag{10}$$

and since we assumed $\sigma^2 = 1$ we get $-(y - f_\theta(\boldsymbol{x}))^2$ times a constant term. Since the mse loss is typically minimised, the negative sign disappears.

Conservative objective models (COMs) are a recently proposed method (Trabucco et al., 2021) for MBO. Conceptually, the method can be thought of as simply training an approximate oracle $f_\theta(\boldsymbol{x})$ but with the model subjected to an extra regularisation term that penalises predictions for samples that have been generated with $f_\theta$, which are assumed to be adversarial examples. In order to mitigate the issue of adversarial examples and over-scoring, the authors propose a regularisation term that penalises the magnitude of samples that have been generated in the vicinity of $\boldsymbol{x}$:

$$\mathcal{L}_\theta^{\text{sup}}(\boldsymbol{x}, y; \alpha) := \log p_\theta(y|\boldsymbol{x}) + \underbrace{\alpha\big[ - \mathbb{E}_{\boldsymbol{x}' \approx p_\theta(\boldsymbol{x}'|\boldsymbol{x}_0), \boldsymbol{x}_0 \sim p(\boldsymbol{x})} f_\theta(\boldsymbol{x}') + f_\theta(\boldsymbol{x})\big]}_{\text{COMs regulariser}}. \tag{11}$$

The following sampler is used for $p_\theta(\boldsymbol{x}|\boldsymbol{x}_0)$:

$$\begin{aligned} \boldsymbol{x}_{t+1} &:= \boldsymbol{x}_t + \epsilon \nabla_{\boldsymbol{x}_t}[-E_\theta(\boldsymbol{x}_t)] \\ &\boxed{= \boldsymbol{x}_t + \epsilon \nabla_{\boldsymbol{x}_t} f_\theta(\boldsymbol{x}_t),} \end{aligned} \tag{12}$$

where $\epsilon$ is constant for each time step. What is interesting is that this procedure does not inject any noise; because of this, samples will instead converge to a *maximum a posteriori* solution, i.e. one of the modes of the distribution $p_\theta(\boldsymbol{x})$ (Welling & Teh, 2011) (hence the use of the approximate symbol $\approx$ in the expectation of Equation 12). This can be problematic if there is very little inter-sample diversity amongst generated samples, as they will be less robust as a whole to the ground truth oracle.

### 2.1. Relationship to EBMs

Furthermore, we note that the regularisation term inside $\alpha$ in Equation 11 is actually *equivalent* to Equation 8 if we define $f_\theta(\boldsymbol{x}) = -E_\theta(\boldsymbol{x})$, and this in turn is equivalent to $\log p_\theta(\boldsymbol{x})$. This, combined with the classification loss $\log p_\theta(y|\boldsymbol{x})$ defines a *joint distribution* $p_\theta(\boldsymbol{x}, y)$, which is precisely the loss proposed in the original paper (Trabucco et al., 2021). Let us propose a special joint density $p(\boldsymbol{x}, y; \alpha)$ where $\alpha$ controls the trade-off between the two likelihood terms:

$$\begin{aligned} & p_\theta(\boldsymbol{x}, y; \alpha) \propto p_\theta(y|\boldsymbol{x}) p_\theta(\boldsymbol{x})^\alpha \\ \implies & \log p_\theta(\boldsymbol{x}, y; \alpha) \propto \log p_\theta(y|\boldsymbol{x}) + \alpha \log p_\theta(\boldsymbol{x}) \qquad (13) \\ & = \underbrace{-\frac{1}{2}(y - f_\theta(\boldsymbol{x}))^2}_{\log p_\theta(y|\boldsymbol{x})} + \alpha\big[ - \mathbb{E}_{\boldsymbol{x}' \sim p_\theta(\boldsymbol{x}'|\boldsymbol{x}_0), \boldsymbol{x}_0 \sim p(\boldsymbol{x})} f_\theta(\boldsymbol{x}') + f_\theta(\boldsymbol{x})\big] \\ & = -\frac{1}{2}(y - \underbrace{f_\theta(\boldsymbol{x})}_{-E_\theta(\boldsymbol{x})})^2 + \alpha \underbrace{\big[\mathbb{E}_{\boldsymbol{x}' \sim p_\theta(\boldsymbol{x})} E_\theta(\boldsymbol{x}') - E_\theta(\boldsymbol{x})\big]}_{\log p_\theta(\boldsymbol{x}), \text{ Eqn. } 5} \qquad (14) \end{aligned}$$

Setting aside for now the fact that Equation 12 is not properly sampling from $p_\theta(\boldsymbol{x})$, what we are observing is a special type of *CD-EBM* where the *negative energy* $-E_\theta(\boldsymbol{x})$ is equivalent to the predicted reward $f_\theta(\boldsymbol{x})$. In other words, $p_\theta(\boldsymbol{x})$ is *proportional* to $f_\theta(\boldsymbol{x})$. The coefficient $\alpha$ dictates the balance between the classification loss and the marginal likelihood over $\boldsymbol{x}$. For instance, if $\alpha = 0$ then $\log p(y|\boldsymbol{x})$ remains and no density estimation is being done over $\boldsymbol{x}$. Conversely, if $\alpha$ was extremely large then it would not be a good predictor of the actual reward of $\boldsymbol{x}$, but good at modelling the distribution of $\boldsymbol{x}$'s since the model is heavily skewed to favour that task. Intuitively then, it would seem that one would want to choose an $\alpha$ such that both tasks are performed well, but this may be difficult to achieve. We will return to this issue in Section 2.3.

### 2.2. *Stochastic* COMs using Langevin MCMC

Previously, we mentioned that COMs' sampling procedure is not actually drawing samples from the generative model $p_\theta(\boldsymbol{x})$; instead, it is simply finding a *maximum a posteriori* solution (i.e. a mode of the distribution). To fix this, we simply need to replace the gradient ascent sampler with Langevin MCMC algorithm in Equation 6:

$$\begin{aligned} \boldsymbol{x}_{t+1} &:= \boldsymbol{x}_t + \frac{\epsilon_t^2}{2} \nabla_{\boldsymbol{x}_t} \log p_\theta(\boldsymbol{x}_t) + \epsilon_t \boldsymbol{z}_t \\ &\boxed{= \boldsymbol{x}_t + \frac{\epsilon_t^2}{2} \nabla_{\boldsymbol{x}_t} \underbrace{f_\theta(\boldsymbol{x}_t)}_{-E_\theta(\boldsymbol{x})} + \epsilon_t \boldsymbol{z}_t,} \end{aligned} \tag{15}$$

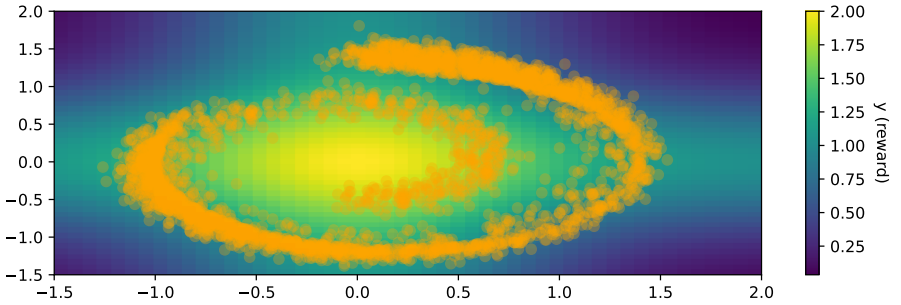



*Figure 1.* 2D spiral dataset. Orange points are samples from the ground truth marginal $p(\boldsymbol{x})$, and background colours correspond to values of $y$ for the ground truth oracle $f(\boldsymbol{x}) = \sum_{i=1}^{2} \exp((\boldsymbol{x}_i - 0)^2)$.

### 2.3. *Decoupled* COMs

In Section 2.1 we showed that COM's training objective induces a special joint density $p_\theta(\boldsymbol{x}, y; \alpha)$ where $\alpha$ controls the trade-off between modelling $p_\theta(\boldsymbol{x})$ and also the classifier $p_\theta(y|\boldsymbol{x})$. If $\alpha$ were to be carefully tuned then we would hope that samples drawn from the model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$ would not only be plausible (i.e. lie on the data distribution) but also comprise high reward on average. One concern is that since the same model $E_\theta$ is parameterising both distributions, achieving this might be cumbersome. Here we propose an alternative, one that decouples the training of both. Let us denote $p_\theta(\boldsymbol{x})$ any learned energy-based model[4] on $\boldsymbol{x}$, and also introduce an independently trained oracle $f_\omega(\boldsymbol{x})$ which is a standard regression model trained to predict $y$ from $\boldsymbol{x}$. We propose the following tilted density (Asmussen & Glynn, 2007; O'Donoghue et al., 2020):

$$p_{\theta,\omega}(\boldsymbol{x}; w) = p_\theta(\boldsymbol{x}) \exp(w f_\omega(\boldsymbol{x}) - \kappa(1/w)) \tag{16}$$

$$\implies \log p_{\theta,\omega}(\boldsymbol{x}; w) = \log p_\theta(\boldsymbol{x}) + w f_\omega(\boldsymbol{x}) - \text{const.}, \tag{17}$$

where $w$ is a hyperparameter weighting our preference for $\boldsymbol{x}$'s with high reward (with respect to $f_\omega$) and $\kappa$ is a normalising constant and does not depend on $\boldsymbol{x}$. To sample, we simply use the following Langevin MCMC sampler:

$$\begin{aligned} \boldsymbol{x}_{t+1} &:= \boldsymbol{x}_t + \frac{\epsilon^2}{2} \nabla_{\boldsymbol{x}_t} \Big[ w f_\omega(\boldsymbol{x}) + \log p_\theta(\boldsymbol{x}_t) \Big] + \epsilon \boldsymbol{z}_t \\ &\boxed{= \boldsymbol{x}_t + \frac{\epsilon^2}{2} \Big[ w \nabla_{\boldsymbol{x}_t} f_\omega(\boldsymbol{x}) + \nabla_{\boldsymbol{x}_t} f_\theta(\boldsymbol{x}_t) \Big] + \epsilon \boldsymbol{z}_t.} \end{aligned} \tag{18}$$

## 3. Experiments and Discussion

**Dataset** We consider a simple 2D spiral dataset that has been modified to also introduce a reward variable $y$. The ground truth function for this reward variable is $f(\boldsymbol{x}) = \sum_{i=1}^{2} \exp((\boldsymbol{x}_i - 0)^2)$, which means that the largest reward is found at the origin $(\boldsymbol{x}_1, \boldsymbol{x}_2) = (0, 0)$. This is illustrated in Figure 1. In the context of MBO, we would like to learn a generative model which is able to sample *valid* points that are as close to the center as possible, since points that are closest to the center will have a larger reward. Here, a 'valid' point is one that lies on the spiral, i.e. some $\boldsymbol{x}$ for which $p(\boldsymbol{x}) > 0$ for the ground truth distribution $p(\boldsymbol{x})$. As we can see in the figure, the point $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2) = (0, 0)$ which lies at the center would not be valid.

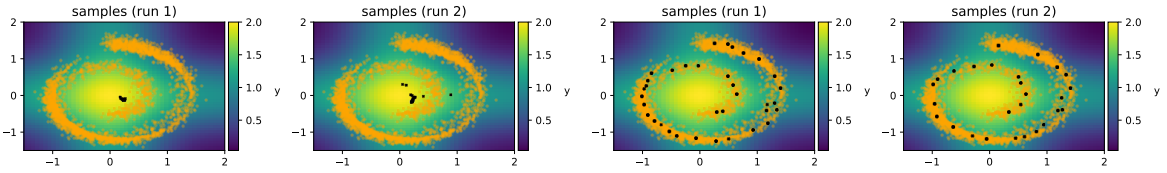
**Training** The energy model is a one hidden layer MLP of 256 units, and we use $k$-contrastive divergence during training for $k = 100$ time steps (see Equation 9). Its associated variance schedule for Langevin MCMC is a geometric sequence from $0.02 \to 0.001$ over those $k$ intervals.[5]. At generation time, we run Langevin MCMC for 50k timesteps with prior distribution $p_\pi(\boldsymbol{x}_0) = \text{Uniform}(-1.5, 2)$ and use a geometric schedule from $0.1 \to 10^{-5}$.

**Results** We train each of the three variants of COMs, and their results are shown in Figures 2, 3, and 4, respectively. For the first two, we train two variants: one where $\alpha = 0$ and the model reduces down to just a regression model (classifier), and

[4]We can even use a COM for which $\alpha$ is large enough such that most of the model is spent on modelling the data distribution. In fact, we found that the training dynamics of this was more stable than the training of a CD-EBM.

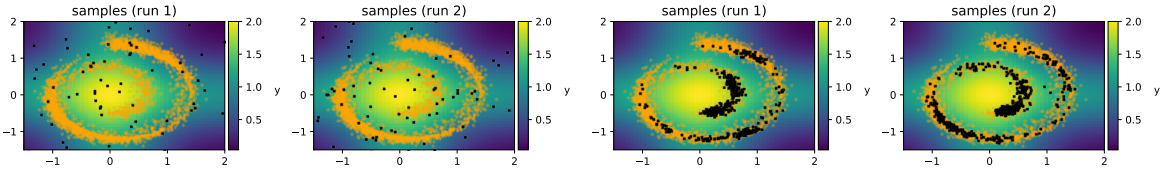
[5]This can be generated easily in Numpy via `numpy.geomspace(a=0.02, b=0.001, n)`

one where $\alpha = 50$ where the model is heavily weighted to model the data distribution instead. Since the original COM uses gradient ascent as its sampler, samples are heavily biased towards seeking modes and sample diversity suffers as a result. In the stochastic variant this is fixed, however we found it difficult to choose an $\alpha$ such that samples were simultaneously concentrated near the center but *on* the spiral, which would constitute the best samples for this dataset. As we mentioned in Section 2.1, we believe it is because we're using the same energy model to model both $p_\theta(\boldsymbol{x})$ and $f_\theta(\boldsymbol{x})$, and therefore either task is not able to be learned sufficiently well. In decoupled COMs however (Figure 4) the energy $E_\theta(\boldsymbol{x})$ and $f_\omega(\boldsymbol{x})$ are separate models and the latter is weighted by hyperparameter $w$. We can see that for modest values of $w$ we obtain samples that progressively become more heavily concentrated at the center, but are still lying on the spiral.



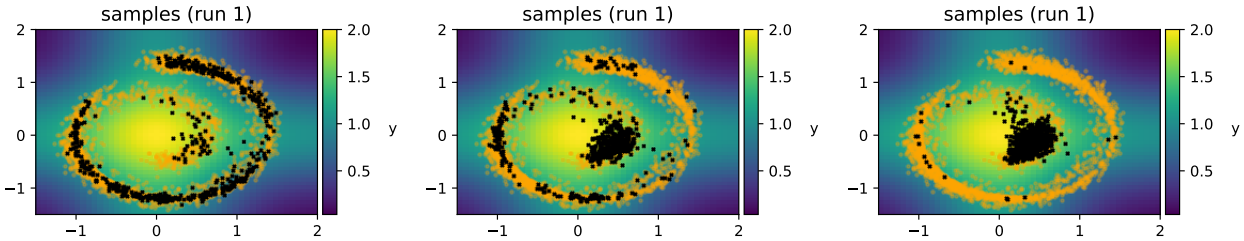

(a) ***Original COMs***: $\alpha = 0$ (b) $\alpha = 50$

*Figure 2.* Generated samples (shown as black crosses) for the ***original COMs*** formulation (Sec. 2). Orange points are those from the real distribution $p(\boldsymbol{x})$, and the colourbar denotes the ground truth $y$. In 2a and 2b we show samples for $\alpha = 0$ and $\alpha = 50$ respectively, for two separate training runs (seeds). For $\alpha = 0$, only the conditional distribution $p_\theta(y|\boldsymbol{x})$ modelled, rather than $\boldsymbol{x}$ and $y$ jointly. For the $\alpha = 50$ case, the energy loss is heavily weighted in favour of modelling $p_\theta(\boldsymbol{x})$. Because the original COMs formulation uses a gradient ascent MCMC sampler, only modes can be sampled from the distribution, and sample diversity suffers as a consequence. This issue is addressed with ***Stochastic COMs*** (Fig. 3), which uses Equation 15 to properly sample from the distribution.




(a) ***Stochastic COMs***: $\alpha = 0$ (b) $\alpha = 50$

*Figure 3.* Generated samples (shown as black crosses) for the ***stochastic COMs*** formulation (Sec. 2.2). Orange points are those from the real distribution $p(\boldsymbol{x})$, and the colourbar denotes the ground truth $y$. In 3a and 3b we show samples for $\alpha = 0$ and $\alpha = 50$ respectively, for two separate training runs (seeds). For $\alpha = 0$, only the conditional distribution $p_\theta(y|\boldsymbol{x})$ modelled, rather than $\boldsymbol{x}$ and $y$ jointly. For the $\alpha = 50$ case, the energy loss is heavily weighted in favour of modelling $p_\theta(\boldsymbol{x})$. While samples for both $\alpha$'s are more diverse, it is difficult to select for the 'good' samples, i.e. those that are close to the center while still lying on the spiral (see Figure S6) for additional enumerations of $\alpha$). This is because the same energy function is being used to parameterise both distributions. We resolve this issue with the decoupled COMs variant, which is shown in Figure 4.




(a) ***Decoupled COMs***: $w = 0$ (b) $w = 5$ (c) $w = 10$

*Figure 4.* Generated samples (shown as black crosses) for ***decoupled COMs*** (Sec. 2.3). Like stochastic COMs, Langevin MCMC is also used here but we also leverage the gradient of an externally trained regression model $f_\omega(\boldsymbol{x})$ as shown in Equation 18. Here, we achieve the desired behaviour: a modest value of $w$ gives us samples that mostly lie on the part of the spiral closest to the center.

*Table 1.* Summary of the three variants of COMs in this work. *Stochastic* (Sec. 2.2) and *decoupled* (Sec. 2.3) variants are proposed methods that address particular issues in the original formulation. Note that for all generated samples shown in Figures 2, 3 and 4, either $p_\theta(\boldsymbol{x})$ is used (original, *stochastic*) or the exponentially tilted variant $p_\theta(\boldsymbol{x})\exp(f_\omega(\boldsymbol{x})^w)$ (*decoupled*). † = samples converge to a maximum a posteriori solution, so we are not truly sampling from $p_\theta(\boldsymbol{x})$ (this is addressed in the stochastic variant).

| Method | Joint density | Training algorithm for $p_\theta(\boldsymbol{x})$ | Sampling algorithm for $p_\theta(\boldsymbol{x})$ |
|---|---|---|---|
| COMs (Sec. 2) | $p_\theta(\boldsymbol{x}, y; \alpha) \propto p_\theta(\boldsymbol{x})^\alpha p_\theta(y\|\boldsymbol{x})$ | Contrastive divergence (approximate†) (Eqn 11) | Gradient ascent† (Eqn. 12) |
| Stochastic COMs (Sec. 2.2) | $p_\theta(\boldsymbol{x}, y; \alpha) \propto p_\theta(\boldsymbol{x})^\alpha p_\theta(y\|\boldsymbol{x})$ | Contrastive divergence | Langevin MCMC (Eqn. 15) |
| Decoupled COMs (Sec. 2.3) | $p_{\theta,\omega}(\boldsymbol{x}, y; w) \propto p_\theta(\boldsymbol{x})\exp(f_\omega(\boldsymbol{x})^w)p_\theta(y\|\boldsymbol{x})$ | Contrastive divergence | Langevin MCMC (Eqn. 18) |

## 4. Related work

Recently, *score-based generative models (SBGMs)* have been in wide use (Song & Ermon, 2019; 2020), and this also includes the diffusion class of models since they are theoretically very similar (Sohl-Dickstein et al., 2015; Ho et al., 2020). Due to space constraints we defer an extended discussion to Section A.3, though we heavily conjecture that this class is model is significantly more robust than COMs and therefore CD-EBMs. This is for the following reasons:

- SBGMs sidesteps the issue of having to generate samples from the distribution during training with MCMC, which significantly speeds up training. This is because the training objective used is score matching (matching derivatives), as opposed to contrastive divergence which requires negative samples be generated.
- SBGMs model the gradient directly $s_\theta(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x})$. Not only does this bypass the need to compute gradients at generation time with autograd, it also means that the energy function can model more information about its input because it is now a mapping from $\mathbb{R}^d \to \mathbb{R}^d$ (where $d$ is the input data dimension), as opposed to $E_\theta(\boldsymbol{x})$ which is a mapping from $\mathbb{R}^d \to \mathbb{R}$ (Salimans & Ho, 2021). Furthermore, this mapping from $\mathbb{R}^d \to \mathbb{R}^d$ allows one to use specialised encoder-decoder models such as the U-Net (Ronneberger et al., 2015), which leverages skip connections to combine information at various resolutions of the input.
- Modern SBGMs also propose score matching over *many* different noise scales. Both large and small are important, since larger ones make it easier to cover all modes and smaller ones are closer to the score of the actual data distribution. All of these noise scales are learned within the same network $s_\theta(\boldsymbol{x})$. At generation time, these noise scales are combined to give rise to an annealed version of Langevin MCMC which iterates from larger noise scales to smaller ones.

We note that a modern SBGM can be constructed by simply replacing the contrastive-based formulation of $p_\theta(\boldsymbol{x})$ in the decoupled COMs variant (Section 2.3) with one that has been trained with score matching as per Song & Ermon (2019). This combined with an external classifier $f_\omega$ becomes very reminiscent to the 'classifier guidance' style of techniques introduced in Dhariwal & Nichol (2021); Ho & Salimans (2022).

## 5. Conclusion

In this work, we showed that COMs, a highly performant algorithm for offline model-based optimisation, is essentially an energy-based model trained via contrastive divergence. COMs use the same energy model to parameterise both the unconditional and conditional parts of the data distribution ($p_\theta(\boldsymbol{x})$ and $p(y|\boldsymbol{x})$, respectively), and this also means that the model has a special property in which the probability of sampling an input is *proportional* to its predicted reward. In this work we identified two shortcomings with the original formulation: firstly, a gradient ascent sampler is used which limits sample diversity; and secondly the parameterisation of both distributions hinders conditional sampling quality, as demonstrated on a toy 2D spiral dataset. We address both of these issues with a 'decoupled' variant of COMs which models the conditional and unconditional parts of the joint distribution separately, as well as use a Langevin MCMC sampler which correctly samples from the learned distibution. Lastly, we contribute a brief discussion comparing the training dynamics of COMs with more recent energy-based models which are trained with score matching.

## References


Asmussen, S. and Glynn, P. W. *Stochastic simulation: algorithms and analysis*, volume 57. Springer, 2007.

Beckham, C., Piche, A., Vazquez, D., and Pal, C. Towards good validation metrics for generative models in offline model-based optimisation. *arXiv preprint arXiv:2211.10747*, 2022.

Bengio, Y. and Delalleau, O. Justifying and generalizing contrastive divergence. *Neural computation*, 21(6):1601–1621, 2009.

Brookes, D., Park, H., and Listgarten, J. Conditioning by adaptive sampling for robust design. In *International conference on machine learning*, pp. 773–782. PMLR, 2019.

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

Fannjiang, C. and Listgarten, J. Autofocused oracles for model-based design. *Advances in Neural Information Processing Systems*, 33:12945–12956, 2020.

Fu, J. and Levine, S. Offline model-based optimization via normalized maximum likelihood estimation. *arXiv preprint arXiv:2102.07970*, 2021.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

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

Imai, K. Pol 571: Expectation and functions of random variables. *Princeton University*, 2006.

Kumar, A. and Levine, S. Model inversion networks for model-based optimization. *Advances in Neural Information Processing Systems*, 33:5126–5137, 2020.

Neal, R. M. et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

O'Donoghue, B., Osband, I., and Ionescu, C. Making sense of reinforcement learning and probabilistic inference. *arXiv preprint arXiv:2001.00805*, 2020.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Salimans, T. and Ho, J. Should ebms model the energy or the score? In *Energy Based Models Workshop-ICLR 2021*, 2021.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

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

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

Song, Y. and Kingma, D. P. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Trabucco, B., Kumar, A., Geng, X., and Levine, S. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, pp. 10358–10368. PMLR, 2021.

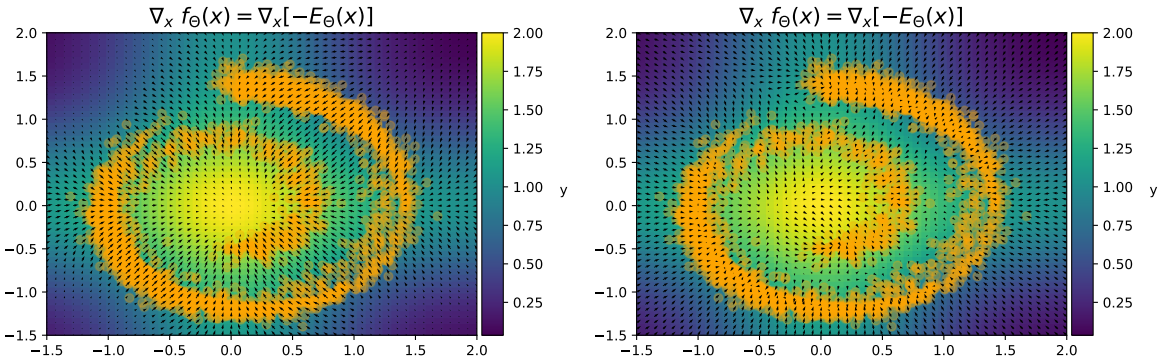
Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

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.

# A. Appendix

## A.1. Additional figures

We can visualise the gradient of the energy function $\nabla_{\boldsymbol{x}}[-E_\theta(\boldsymbol{x})] = \nabla_{\boldsymbol{x}} f_\theta(\boldsymbol{x})$ by plotting their values (vectors) over the entire 2D grid, creating a vector field. This is shown in Figure S5. We can see that when $\alpha = 0$, the energy function points to the center of the spiral since it is only trained to predict the reward of $\boldsymbol{x}$ and this is where the predicted reward is largest. Conversely, when $\alpha = 50$ the energy model is heavily weighted in favour of modelling the (unconditional) distribution of $\boldsymbol{x}$ during training, and the vector field points towards examples *on* the spiral.




(a) $\alpha = 0$ (b) $\alpha = 100$

*Figure S5.* Stochastic COMs (Section 2.2). Vector field plots for the learned energy function, for $\alpha = 0$ (S5a) and $\alpha = 50$ (S5b). Best viewed with a PDF viewer at a higher zoom level. Training details are specified in Section 3.

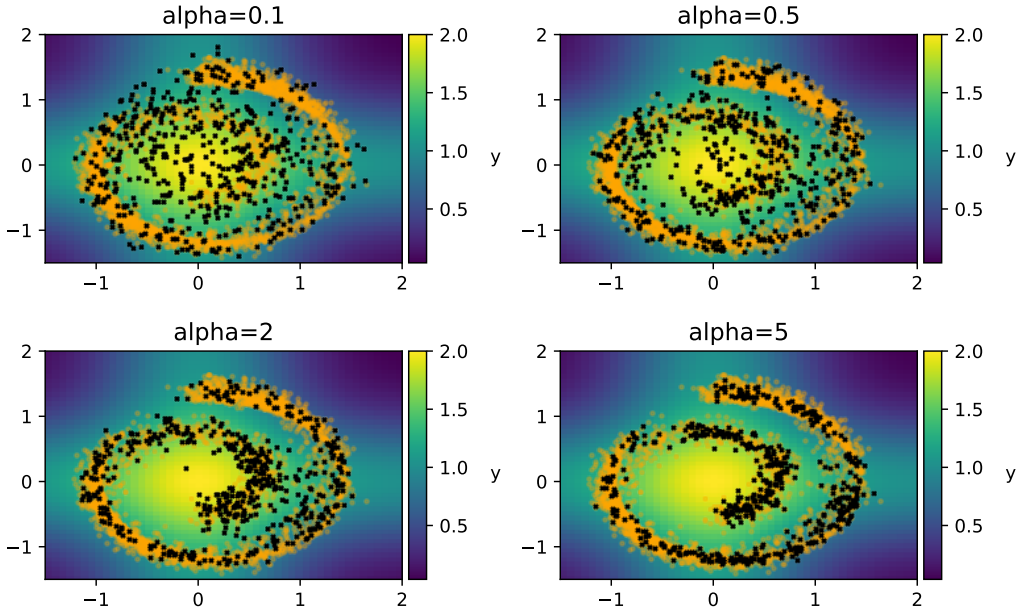



*Figure S6.* Additional plots to complement Figure 3. Values of $\alpha$ are denoted in each subplot. Similar to Figure 3, even with small values of $\alpha$ we are unable to generate samples such that they both lie on the spiral *and* are close to the center, where the ground truth reward $y$ is largest.

### A.2. Additional math

This proof is used for Equation 5. From Thereom 6 of Imai (2006) we have that, satisfying some conditions, the derivative of the expected value is the expected value of the derivative:

$$\nabla_\theta \big[\mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})} E_\theta(\boldsymbol{x})\big] = \nabla_\theta \big[\int_{\boldsymbol{x}} p_\theta(\boldsymbol{x}) E_\theta(\boldsymbol{x}) d\boldsymbol{x}\big] \tag{19}$$

$$= \int_x -\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x}) p_\theta(\boldsymbol{x}) d\boldsymbol{x} \;\; \text{(thm. 6)} \tag{20}$$

$$= \mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})}\big[-\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x})\big]. \tag{21}$$

### A.3. Score-matching EBMs (SM-EBMs)

A recent class of generative model that has enjoyed immense success is the *score-matching EBM* (SM-EBM). Score matching refers to the minimisation of the *Fisher* divergence between the real and generative distributions, which is equivalent to measuring the difference between their respective log derivatives (Hyvärinen & Dayan, 2005):

$$D_F(p(\boldsymbol{x})||p_\theta(\boldsymbol{x})) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\Big[\frac{1}{2}\|\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x})\|^2\Big]. \tag{22}$$

Here, the score refers to the the derivative of the log density with respect to the input, i.e. $\nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x})$. As we have seen already, this can be parameterised with an energy model $E_\theta(\boldsymbol{x})$. Unfortunately, in its current form Equation 22 is intractable since it is assumed the score of the data distribution is known. While an equivalent form can be written that only relies $p_\theta(\boldsymbol{x})$, it relies on computing the Hessian and has quadratic time complexity in the dimension of $\boldsymbol{x}$ (Hyvärinen & Dayan, 2005). To address these issues (and other theoretical assumptions about the $p(\boldsymbol{x})$), denoising score matching (Vincent, 2011) was proposed, where the Fisher divergence is computed with respect to a 'noisy' version of the data distribution, $q(\boldsymbol{x})$. This can be expressed as the following marginalisation over a conditional noising distribution $q_\sigma(\boldsymbol{x}|\boldsymbol{x}_0)$ and the actual data distribution $p_\theta(\boldsymbol{x})$:

$$q_\sigma(\boldsymbol{x}) = \int_{\boldsymbol{x}_0} q_\sigma(\boldsymbol{x}|\boldsymbol{x}_0) p(\boldsymbol{x}_0) d\boldsymbol{x}, \tag{23}$$

where $q_\sigma(\boldsymbol{x}|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}_0, \sigma^2)$. While this no longer becomes an estimate of $D_F$ when $p(\boldsymbol{x})$ is replaced with $q(\boldsymbol{x})$, the difference is negligible for small values of $\sigma$.

Recently, *score-based generative models (SBGMs)* were proposed (Song & Ermon, 2019; 2020), which can be thought of as an improved version of the denoising score matching EBM but with the added intention of generating samples, which comes in the form of a modified Langevin MCMC sampler. To avoid a deluge of acronyms and terms, let us simply refer to these as 'modern' SM-EBMs. Modern SM-EBMs are currently state-of-the-art, in no small part due to some tricks which improve the training dynamics of the original score matching algorithm (Vincent, 2011). Firstly, instead of modelling an energy function and then having to backprop through it to obtain the actual score, the score is parameterised directly, i.e. $s_\theta(\boldsymbol{x}) \approx \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$ instead of $\nabla_{\boldsymbol{x}}[-E_\theta(\boldsymbol{x})] \approx \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$. This means that $s_\theta : \mathbb{R}^d \to \mathbb{R}^d$ and encoder-decoder-style architectures must be used. Secondly, denoising score matching has issues with modelling modes of the data distribution when they are separated by low (or zero) density regions. While larger noise perturbations make this easier, the noise distribution $q(\boldsymbol{x})$ would become less representative of the actual data distribution $p(\boldsymbol{x})$. To resolve this dilemma, a series of noise distributions are used instead. For some $t \in \{1, \dots, T\}$:

$$q_t(\boldsymbol{x}|\boldsymbol{x}_0) = q_{\sigma_t}(\boldsymbol{x}|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}_0, \sigma_t^2) \tag{24}$$

$$\implies q_t(\boldsymbol{x}) = \int_{\boldsymbol{x}_0} q_{\sigma_t}(\boldsymbol{x}|\boldsymbol{x}_0) p(\boldsymbol{x}_0) d\boldsymbol{x}_0, \tag{25}$$

where the sequence of $\sigma_t$'s follows a positive geometric sequence, and $T$ is large enough such that $\sigma_T \approx 0$ (so $q_T \approx p(\boldsymbol{x})$). With this in mind, we must also modify the score predictor to also condition on a timestep, i.e. $s_\theta(\boldsymbol{x}; t)$. $s_\theta(\boldsymbol{x}; t)$ is trained to estimate the score $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x})$. We defer training details to Song & Ermon (2019; 2020), though it suffices to say that at generation time an annealed version of Langevin MCMC is used where the noise magnitude $\sigma$ progressively becomes smaller as the number of timesteps increases.