Contrastive Energy Prediction for Exact Energy-Guided Diffusion Sampling in Offline Reinforcement Learning

Cheng Lu [* 1]   Huayu Chen [* 1]   Jianfei Chen [1]   Hang Su [1]   Chongxuan Li [2]   Jun Zhu [1 3]

Abstract

Guided sampling is a vital approach for applying diffusion models in real-world tasks that embeds human-defined guidance during the sampling procedure. This paper considers a general setting where the guidance is defined by an (unnormalized) energy function. The main challenge for this setting is that the intermediate guidance during the diffusion sampling procedure, which is jointly defined by the sampling distribution and the energy function, is unknown and is hard to estimate. To address this challenge, we propose an exact formulation of the intermediate guidance as well as a novel training objective named contrastive energy prediction (CEP) to learn the exact guidance. Our method is guaranteed to converge to the exact guidance under unlimited model capacity and data samples, while previous methods can not. We demonstrate the effectiveness of our method by applying it to offline reinforcement learning (RL). Extensive experiments on D4RL benchmarks demonstrate that our method outperforms existing state-of-the-art algorithms. We also provide some examples of applying CEP for image synthesis to demonstrate the scalability of CEP on high-dimensional data. Code is available at https://github.com/thu-ml/CEP-energy-guided-diffusion.

1. Introduction

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b; Karras et al., 2022) have demonstrated incredible success. A key for applying diffusion

*Equal contribution [1]Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center, Tsinghua-Bosch Joint ML Center, THBI Lab, Tsinghua University [2]Gaoling School of AI, Renmin University of China; Beijing Key Lab of Big Data Management and Analysis Methods, Beijing, China [3]Pazhou Lab (Huangpu), Guangzhou, China. Correspondence to: Jun Zhu <dcszj@tsinghua.edu.cn>.

models in real-world tasks is to embed human controllability in the sampling procedure. A common paradigm for introducing human preference in diffusion models is *guided sampling*, which includes *classifier guidance* (Dhariwal & Nichol, 2021), *classifier-free guidance* (Ho & Salimans, 2021) and other guidance methods (Nichol et al., 2021; Ho et al., 2022c; Zhao et al., 2022). By leveraging guided sampling, diffusion models can realize amazing text-to-image generation (Saharia et al., 2022b), video generation (Ho et al., 2022c;a; Yang et al., 2022; Zhou et al., 2022), controllable text generation (Li et al., 2022), inverse molecular design (Bao et al., 2022b) and reinforcement learning (Janner et al., 2022; Chen et al., 2022; Ajay et al., 2022).

Both classifier and classifier-free guidance deal with a conditional sampling problem where there exists paired data with additional conditioning variables during the training procedure (Dhariwal & Nichol, 2021; Graikos et al., 2022; Rombach et al., 2022). However, sometimes it is difficult to describe human preference through a conditioning variable and we can only embed our preference through a scalar function. Examples of such a function include a reward function or pretrained Q-function in reinforcement learning (Janner et al., 2022; Chen et al., 2022), human feedback in dialogue systems (Ziegler et al., 2019), cosine similarity between sample features and designated features in image synthesis (Kwon & Ye, 2022), or L_2-distance between the sampled frame and the previous frame in video synthesis (Ho et al., 2022c). In such cases, we aim to leverage human preference to manipulate the desired distribution and draw samples by diffusion sampling with additional guidance, while it is hard to directly use classifier or classifier-free guidance since no actual condition is provided.

We consider a general form that subsumes all the above cases. Let $q(\boldsymbol{x})$ be an unknown data distribution in $\mathcal{X} \subseteq \mathbb{R}^d$. We aim to sample from the following distribution:

$$p(\boldsymbol{x}) \propto q(\boldsymbol{x})e^{-\beta\mathcal{E}(\boldsymbol{x})}, \qquad (1)$$

where $\mathcal{E}(\cdot)$ is an *energy function* from $\mathcal{X} \in \mathbb{R}^d$ to $\mathbb{R}$ and we can compute $\mathcal{E}(\boldsymbol{x})$ for each datum. $\beta \geq 0$ is the inverse temperature for controlling the energy strength. The high-density region of $p(\boldsymbol{x})$ is approximately the intersection of both the high-density regions of $q(\boldsymbol{x})$ and $e^{-\beta\mathcal{E}(\boldsymbol{x})}$. As a

result, we can insert controllability by embedding the desired properties into the energy function $\mathcal{E}(\cdot)$. The choice of the energy function $\mathcal{E}(\cdot)$ is highly flexible: we only need to ensure that the integral of $q(\boldsymbol{x})e^{-\beta\mathcal{E}(\boldsymbol{x})}$ over $\boldsymbol{x} \in \mathcal{X}$ is finite. We can also introduce additional conditioning variables c by an energy function $\mathcal{E}(\cdot, c)$. In particular, let $\beta = 1$ and $\mathcal{E}(\boldsymbol{x}, c) = -\log q(c|\boldsymbol{x})$, the target distribution $p(\boldsymbol{x})$ becomes a conditional distribution $q(\boldsymbol{x}|c)$, which recovers the classic conditional sampling problem as a special case.

Sampling from $p(\boldsymbol{x})$ is difficult in general as $p(\boldsymbol{x})$ is unnormalized. Existing attempts (Janner et al., 2022; Zhao et al., 2022; Ho et al., 2022c; Chung et al., 2022; Kawar et al., 2022) leverage a pretrained diffusion model $q_g(\boldsymbol{x}) \approx q(\boldsymbol{x})$ and apply diffusion sampling with an additional guidance term related to $\mathcal{E}(\cdot)$ called *energy guidance*. However, all previously proposed energy guidance is either manually or arbitrarily defined across the diffusion process, and it is unstudied whether the final samples follow the desired distribution $p(\boldsymbol{x})$. In fact, we show that previous energy-guided samplers are all inexact in the sense that they cannot guarantee convergence to $p(\boldsymbol{x})$. To the best of our knowledge, how to use the pretrained diffusion model $q_g(\boldsymbol{x})$ to draw samples from the exact $p(\boldsymbol{x})$ remains largely open.

In this work, we analyze and derive an exact formulation of the desired guidance for diffusion sampling from $p(\boldsymbol{x})$ in Eq. (1). In contrast with previous work, we show that the exact guidance in the diffused data space during sampling is completely determined by the energy function $\mathcal{E}(\cdot)$ in the original data space. The exact energy guidance is in an intractable form which cannot be computed directly, so we propose a novel training method named *contrastive energy prediction (CEP)* to estimate the guidance using samples from $q(\boldsymbol{x})$ as the training data. CEP trains the guidance model by comparing the energy $\mathcal{E}(\cdot)$ within a set of noise-perturbed data samples and using their soft energy labels as supervising signals. We theoretically prove that the gradient of the optimal learned model is exactly the desired energy guidance, and thus the final samples are guaranteed to follow $p(\boldsymbol{x})$. In a special formulation of $\mathcal{E}(\cdot)$ which corresponds to the classic conditional sampling case, we additionally show that CEP could be understood as an alternative contrastive approach to the classifier guidance method.

To verify the effectiveness and scalability of CEP, we take two important applications of Eq. (1): offline reinforcement learning (RL) and image synthesis. For offline RL, we formulate the classic constrained policy optimization problem (Peters et al., 2010; Peng et al., 2019) as Q-guided policy optimization, and evaluate our method in mainstream D4RL (Fu et al., 2020) benchmarks. Extensive experiments demonstrate that our method outperforms existing state-of-the-art algorithms in most tasks, especially in hard tasks such as AntMaze. For image synthesis, we evaluate con-

ditional sample quality by CEP against classic classifier guidance (Dhariwal & Nichol, 2021) both quantitatively and qualitatively on ImageNet and find two methods almost equally well-performing. We also provide an example of energy-guided image synthesis to affect the color appearance of sampled images and validate the flexibility of CEP.

2. Background

We first present preliminary knowledge of diffusion models as well as offline RL that serves as an important motivation and application of sampling from distribution (1).

2.1. Diffusion (Probabilistic) Models

Diffusion (probabilistic) models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b) are powerful generative models. Given a dataset $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$ with N samples of D-dimensional random variable $\boldsymbol{x}_0$ from an unknown data distribution $q_0(\boldsymbol{x}_0)$, diffusion models gradually add Gaussian noise from $\boldsymbol{x}_0$ at time 0 to $\boldsymbol{x}_T$ at time $T > 0$. The transition distribution $q_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0)$ satisfies

$$q_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\alpha_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I}), \qquad (2)$$

where $\alpha_t, \sigma_t > 0$. Denote $q_t(\boldsymbol{x}_t)$ as the marginal distribution of $\boldsymbol{x}_t$ at time t. The transition distribution at time T satisfies $q_T(\boldsymbol{x}_T|\boldsymbol{x}_0) \approx q_T(\boldsymbol{x}_T) \approx \mathcal{N}(\boldsymbol{x}_T|0, \tilde{\sigma}^2\boldsymbol{I})$ for some $\tilde{\sigma} > 0$ and is independent of $\boldsymbol{x}_0$. Thus, starting from $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{x}_T|0, \tilde{\sigma}^2\boldsymbol{I})$, diffusion models aim to recover the original data $\boldsymbol{x}_0$ by solving a reverse process from T to 0. The reverse process can alternatively be the *diffusion ODE* (Song et al., 2021b):

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t), \qquad (3)$$

where $f(t) = \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}, g^2(t) = \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\sigma_t^2$ (Kingma et al., 2021) and the only unknown term is the *score function* $\nabla_{\boldsymbol{x}_t}\log q_t(\cdot)$ of the distribution q_t at each time t. Thus, diffusion models train a neural network $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ parameterized by θ to estimate the scaled score function: $-\sigma_t\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t)$, and the training objective is (Ho et al., 2020; Song et al., 2021b)

$$\min_\theta \mathbb{E}_{t,\boldsymbol{x}_t}\left[\omega(t)\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \sigma_t\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t)\|_2^2\right]$$
$$\Leftrightarrow \min_\theta \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\omega(t)\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}\|_2^2\right] \qquad (4)$$

where $\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, $t \sim \mathcal{U}([0, T])$, $\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$, and $\omega(t)$ is a weighting function and usually set to $\omega(t) \equiv 1$ (Ho et al., 2020). Thus, after training a diffusion model, we usually have $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) \approx -\sigma_t\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t)$. By replacing the score function with $\boldsymbol{\epsilon}_\theta$, we can fastly sample $\boldsymbol{x}_0$ by solving the corresponding diffusion ODEs with some dedicated solvers (Song et al., 2021a; Lu et al., 2022b).

2.2. Constrained Policy Optimization in Offline Reinforcement Learning

Consider a Markov Decision Process (MDP), described by the tuple $\langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$. $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space. $P(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ and $r(\boldsymbol{s}, \boldsymbol{a})$ are respectively the transition and reward functions. $\gamma \in (0, 1]$ is a constant discount factor. In offline RL, a static dataset $\mathcal{D}^\mu$ containing some interacting history $\{\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}'\}$ between a behavior agent $\mu(\boldsymbol{a}|\boldsymbol{s})$ and the environment is given. The goal is to maximize the expected accumulated rewards of a model policy $\pi_\theta(\boldsymbol{a}|\boldsymbol{s})$ in the above MDP by solely utilizing the knowledge learned from the dataset.

Previous work ([Peters et al., 2010](#); [Peng et al., 2019](#)) formulate offline RL as constrained policy optimization:

$$\max_\pi \mathbb{E}_{\boldsymbol{s} \sim \mathcal{D}^\mu}\left[\mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})} Q_\psi(\boldsymbol{s}, \boldsymbol{a}) - \frac{1}{\beta} D_{\mathrm{KL}}\left(\pi(\cdot|\boldsymbol{s}) \| \mu(\cdot|\boldsymbol{s})\right)\right], \tag{5}$$

where Q_ψ is an action evaluation model which indicates the quality of decision $(\boldsymbol{s}, \boldsymbol{a})$ by estimating the Q-function $Q^\pi(\boldsymbol{s}, \boldsymbol{a}) := \mathbb{E}_{\boldsymbol{s}_1=\boldsymbol{s}, \boldsymbol{a}_1=\boldsymbol{a}; \pi}[\sum_{n=1}^\infty \gamma^n r(\boldsymbol{s}_n, \boldsymbol{a}_n)]$ of the current policy π. β is an inverse temperature coefficient. The first term in Eq. (5) intends to perform policy optimization, while the second term stands for policy constraint.

It is shown that the optimal policy π^* in (5) satisfies:

$$\pi^*(\boldsymbol{a}|\boldsymbol{s}) \propto \mu(\boldsymbol{a}|\boldsymbol{s})\, e^{\beta Q_\psi(\boldsymbol{s}, \boldsymbol{a})}, \tag{6}$$

which falls into the general family of distributions (1). Therefore, sampling from the optimal policy π^* can be implemented by energy-guided sampling with a pretrained diffusion behavior model $\mu_g(\boldsymbol{a}|\boldsymbol{s}) \approx \mu(\boldsymbol{a}|\boldsymbol{s})$, which motivates us to propose an exact energy-guided sampling method.

3. Exact Energy-Guided Sampling

To perform energy-guided sampling for Eq. (1), the guidance during the sampling procedure needs to guarantee that final samples follow the desired distribution $p(\boldsymbol{x})$. In this section, we propose an exact formulation of such energy guidance and propose a novel training objective to estimate the guidance. All the proofs can be found in Appendix D.

3.1. Exact Formulation of Intermediate Energy Guidance

Below we formally analyze how to sample from Eq. (1) by diffusion models.

Rewrite $p_0 := p$ and $q_0 := q$. The target distribution is

$$p_0(\boldsymbol{x}_0) \propto q_0(\boldsymbol{x}_0) e^{-\beta \mathcal{E}(\boldsymbol{x}_0)}. \tag{7}$$

Let $q_t(\boldsymbol{x}_t)$ be the marginal distribution of the forward diffusion process at time t defined in Eq. (2) starting from
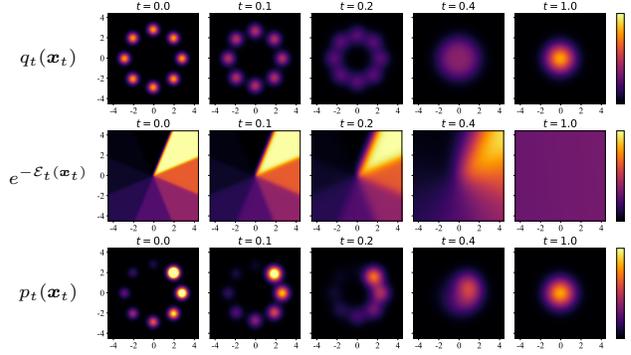

Figure 1. A 2-D mixtures-of-Gaussians example of the density functions (unnormalized) for $q_t(\boldsymbol{x}_t)$, $e^{-\mathcal{E}_t(\boldsymbol{x}_t)}$ and $p_t(\boldsymbol{x}_t)$ during the diffusion process, where $p_t(\boldsymbol{x}_t) \propto q_t(\boldsymbol{x}_t) e^{-\mathcal{E}_t(\boldsymbol{x}_t)}$.

$q_0(\boldsymbol{x}_0)$. Suppose we have a pretrained diffusion model $q_0^\theta(\boldsymbol{x}_0) \approx q_0(\boldsymbol{x}_0)$ by learning a noise prediction model $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) \approx -\sigma_t \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$ parameterized by θ at each time $t \in [0, T]$. By Eq. (3), drawing samples from p_0 requires the corresponding score functions $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ at each intermediate time t of the diffusion process starting from p_0 instead of q_0. Our first key result is on revealing the relationship between the corresponding score functions during the diffusion processes starting from q_0 and p_0, as summarized below:

Theorem 3.1 (Intermediate Energy Guidance). *Suppose q_0 and p_0 are defined as in Eq. (7). For $t \in (0, T]$, let*

$$p_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0) := q_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\alpha_t \boldsymbol{x}_0, \sigma_t^2 \boldsymbol{I}). \tag{8}$$

Denote $q_t(\boldsymbol{x}_t) := \int q_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0) q_0(\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0$ and $p_t(\boldsymbol{x}_t) := \int p_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0) p_0(\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0$ as the marginal distributions at time t, and define

$$\mathcal{E}_t(\boldsymbol{x}_t) := \begin{cases} \beta \mathcal{E}(\boldsymbol{x}_0), & t = 0, \\ -\log \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[e^{-\beta \mathcal{E}(\boldsymbol{x}_0)}\right], & t > 0. \end{cases} \tag{9}$$

Then q_t and p_t satisfy

$$p_t(\boldsymbol{x}_t) \propto q_t(\boldsymbol{x}_t) e^{-\mathcal{E}_t(\boldsymbol{x}_t)}, \tag{10}$$

and their score functions satisfy

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) = \underbrace{\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)}_{\approx -\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)/\sigma_t} - \underbrace{\nabla_{\boldsymbol{x}_t} \mathcal{E}_t(\boldsymbol{x}_t)}_{\substack{\text{energy guidance} \\ \textit{\textcolor{red}{(intractable)}}}}. \tag{11}$$

Theorem 3.2 reveals a previously unnoticed exact form of the intermediate distributions p_t: though p_t is defined as an intractable marginal distribution for all $t > 0$, they could still be written *in the same form* as Eq. (7), proportional to the product of the (diffused) data distribution q_t and an exponential energy term $e^{-\mathcal{E}_t(\boldsymbol{x}_t)}$. Since such energy is defined during the diffusion process, we name $\mathcal{E}_t(\cdot)$ as *intermediate energy*. According to Eq. (9), the intermediate energy is completely determined by the energy function

$\mathcal{E}(\boldsymbol{x}_0)$ at time 0. An illustration is given in Figure 1, from which we draw several observations: (1) $p_t(\boldsymbol{x}_t)$ prefers areas with both high data density $q_t(\boldsymbol{x}_t)$ and high energy density $e^{-\mathcal{E}_t(\boldsymbol{x}_t)}$; (2) Through the forward diffusion process, both p_t and q_t gradually become standard Gaussian, which guarantees that we can reverse the diffusion process starting from the same noise distribution $p_T \approx q_T \approx \mathcal{N}(0, \tilde{\sigma}^2 \boldsymbol{I})$; (3) The energy function $\mathcal{E}_0(\boldsymbol{x}_0)$ is also "diffused" into intermediate energy functions $\mathcal{E}_t(\boldsymbol{x}_t)$ as t increases. In particular, $\mathcal{E}_T(\boldsymbol{x}_T)$ is almost equal to a constant function and thus $\nabla_{\boldsymbol{x}_T} \mathcal{E}_T(\boldsymbol{x}_T) \approx 0$.

The result in Eq. (11) directly defines a principled method to perform guided sampling from $p_0(\boldsymbol{x}_0)$. Namely, we can sample with Eq. (3) as along as we know both $\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$ and $\nabla_{\boldsymbol{x}_t} \mathcal{E}_t(\boldsymbol{x}_t)$. The former score is already given by the pretrained diffusion model ϵ_θ. The remaining problem is to estimate the latter score $\nabla_{\boldsymbol{x}_t} \mathcal{E}_t(\boldsymbol{x}_t)$, which we name as *intermediate energy guidance*. An unbiased estimation of $\nabla_{\boldsymbol{x}_t} \mathcal{E}_t(\boldsymbol{x}_t)$ is generally non-trivial due to the log-expectation formulation and a potentially complex form of $\mathcal{E}(\boldsymbol{x}_0)$ in Eq. (9). To the best of our knowledge, it is still an open problem for estimating the exact intermediate energy guidance. We present a first attempt by developing a novel learning-based method to learn the energy guidance by comparing energy of samples from q_t, as detailed below.

3.2. Learning Energy Guidance by Contrastive Energy Prediction

Let $K > 1$ be a positive integer. Let $\boldsymbol{x}_0^{(1)}, \ldots, \boldsymbol{x}_0^{(K)}$ be K i.i.d. samples from $q_0(\boldsymbol{x}_0)$ and $\boldsymbol{\epsilon}^{(1)}, \ldots, \boldsymbol{\epsilon}^{(K)}$ be K i.i.d. Gaussian samples following $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{I})$. Let $t \sim \mathcal{U}(0, T)$ be a randomly sampled time step. For each $i = 1, \ldots, K$, let $\boldsymbol{x}_t^{(i)} := \alpha_t \boldsymbol{x}_0^{(i)} + \sigma_t \boldsymbol{\epsilon}^{(i)}$, where α_t, σ_t are defined in Eq. (8). Assume that the intermediate energy $\mathcal{E}_t(\cdot)$ is approximated by a network $f_\phi(\cdot, t) : \mathbb{R}^d \to \mathbb{R}$ parameterized by ϕ. We propose to solve the following problem to learn f_ϕ, whose solution is characterized in Theorem 3.2:

$$\min_\phi \mathbb{E}_{p(t)} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[-\sum_{i=1}^{K} \underbrace{e^{-\beta \mathcal{E}(\boldsymbol{x}_0^{(i)})}}_{\text{soft energy label}} \log \underbrace{\frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, t)}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)}, t)}}}_{\text{predicted label}} \right]. \quad (12)$$

Theorem 3.2. *Given unlimited model capacity and data samples, For all $K > 1$ and $t \in [0, T]$, the optimal f_{ϕ^*} in problem (12) satisfies $\nabla_{\boldsymbol{x}_t} f_{\phi^*}(\boldsymbol{x}_t, t) = \nabla_{\boldsymbol{x}_t} \mathcal{E}_t(\boldsymbol{x}_t)$.*

According to Theorem 3.2, we indeed can train the energy guidance model f_ϕ by solving problem (12) and use the gradients of f_ϕ for guided sampling by estimating the energy guidance in Eq. (11). Here we give an intuitive explanation

of Theorem 3.2. To estimate the energy guidance $\nabla_{\boldsymbol{x}_t} \mathcal{E}_t(\cdot)$, we only need to ensure f_ϕ to be a relative proportional value of $\mathcal{E}_t(\cdot)$, so it is enough to relatively compare f_ϕ within K samples instead of directly train f_ϕ with the absolute values. Built upon such an observation, we leverage a cross-entropy loss in problem (12), where the energy $\mathcal{E}(\boldsymbol{x}_0^{(i)})$ of K clean samples $\boldsymbol{x}_0^{(i)}$ are soft supervising labels and the softmax of energy predictions $f_\phi(\boldsymbol{x}_t^{(i)}, t)$ of K noisy samples $\boldsymbol{x}_t^{(i)}$ are predicted labels. Due to the contrastive manner of this objective, we name our proposed method in Eq. (12) as *Contrastive Energy Prediction (CEP)*.

Although the optimal solution in problem (12) is exact, sometimes we may suffer from numerical issues during training because the exponential term $e^{-\beta \mathcal{E}(\boldsymbol{x}_0^{(i)})}$ is unnormalized. For example, suppose $\mathcal{E}(\boldsymbol{x}_0)$ is a complex function that might contain "spikes" at some data point, the exponential term will greatly amplify such instability during training. To address this issue, we further use a self-normalized energy label by normalizing the energy function across the K samples and define the optimization problem as:

$$\min_\phi \mathbb{E}_{p(t)} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[-\sum_{i=1}^{K} \underbrace{\frac{e^{-\beta \mathcal{E}(\boldsymbol{x}_0^{(i)})}}{\sum_{j=1}^{K} e^{-\beta \mathcal{E}(\boldsymbol{x}_0^{(j)})}}}_{\text{self-normalized energy label}} \log \underbrace{\frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, t)}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)}, t)}}}_{\text{predicted label}} \right]. \quad (13)$$

Such a self-normalized objective can ensure the soft energy label is within $[0, 1]$ and the sum of each item is exactly 1. Although this objective may be biased against the original one in problem (12), we empirically find that it can greatly improve the numerical stability and help to achieve a good converged result. Moreover, we can reduce bias in the objective by increasing K. For $K \to \infty$, the objective in (13) is equivalent to that in (12) because $\sum_{\boldsymbol{x}_0} e^{-\beta \mathcal{E}(\boldsymbol{x}_0)} = \mathbb{E}_{q_0(\boldsymbol{x}_0)}[e^{-\beta \mathcal{E}(\boldsymbol{x}_0)}]$ is the normalizing constant of p_0. Therefore, a larger K is preferred in practice given enough computation and memory budget.

4. Comparison with Previous Methods for Guided Sampling

Below we compare CEP with previous methods for guided sampling. We show that all previous energy-guided samplers are inexact; and for a special case of the energy function which corresponds to the conditional sampling problem, CEP is a contrastive alternative to classifier guidance.

4.1. Previous Energy-Guided Samplers are Inexact

In this section, we show that previous energy-guided samplers for Eq. (7) are all inexact and do not guarantee conver-
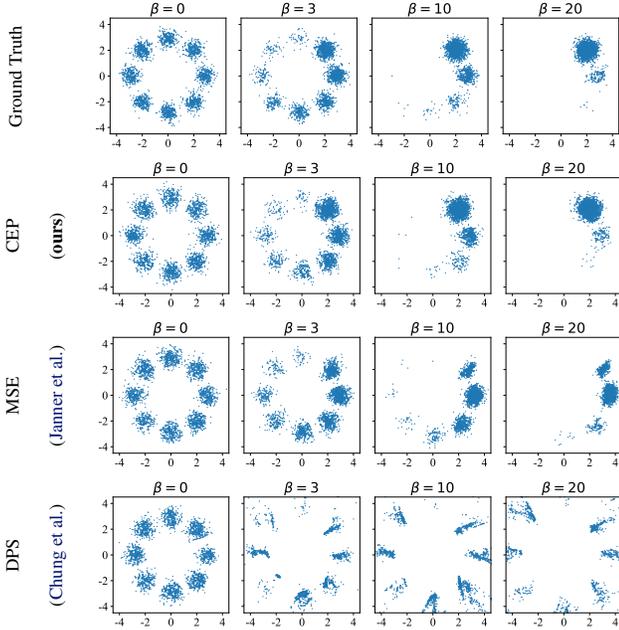
Figure 2. A 2-D example for comparing different energy-guided sampling algorithms, varying different inverse temperature β.

Table 1. Comparison between energy-guided sampling algorithms.

Method	Optimal Solution of Energy	Exact Guidance
CEP (ours)	$-\log \mathbb{E}_{q_{0t}(\boldsymbol{x}_0\vert\boldsymbol{x}_t)}\left[e^{-\mathcal{E}_0(\boldsymbol{x}_0)}\right]$	✓
MSE	$\mathbb{E}_{q_{0t}(\boldsymbol{x}_0\vert\boldsymbol{x}_t)}[\mathcal{E}_0(\boldsymbol{x}_0)]$	✗
DPS	$\mathcal{E}_0\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0\vert\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right)$	✗

gence to p_0. Without loss of generality, we focus on a fixed time $t \in (0, T]$. We summarize the relationship in Table 1.

MSE for Predicting Energy. Many existing energy guidance methods (Janner et al., 2022; Bao et al., 2022b) use a mean-square-error (MSE) objective to train an energy model $f_\phi(\boldsymbol{x}_t, t)$ and use its gradient for energy guidance. The training objective is:

$$\min_\phi \mathbb{E}_{q_{0t}(\boldsymbol{x}_0, \boldsymbol{x}_t)}\left[\|f_\phi(\boldsymbol{x}_t, t) - \mathcal{E}_0(\boldsymbol{x}_0)\|_2^2\right] \quad (14)$$

Given unlimited model capacity, the optimal f_ϕ satisfies:

$$f_\phi^{\text{MSE}}(\boldsymbol{x}_t, t) = \mathbb{E}_{q_{0t}(\boldsymbol{x}_0\vert\boldsymbol{x}_t)}\left[\mathcal{E}_0(\boldsymbol{x}_0)\right]$$

However, by Eq. (9), the true energy function satisfies

$$\mathcal{E}_t(\boldsymbol{x}_t) = -\log \mathbb{E}_{q_{0t}(\boldsymbol{x}_0\vert\boldsymbol{x}_t)}\left[e^{-\mathcal{E}_0(\boldsymbol{x}_0)}\right]$$
$$\geq \mathbb{E}_{q_{0t}(\boldsymbol{x}_0\vert\boldsymbol{x}_t)}\left[\mathcal{E}_0(\boldsymbol{x}_0)\right] = f_\phi^{\text{MSE}}(\boldsymbol{x}_t, t),$$

and the equality only holds when $t = 0$. Therefore, the MSE energy function f_ϕ^{MSE} is inexact for all $t > 0$. Moreover, we show in Appendix E that the gradient of f_ϕ^{MSE} is also inexact against the true guidance $\nabla_{\boldsymbol{x}_t}\mathcal{E}(\boldsymbol{x}_t)$.

Diffusion Posterior Sampling. There also exist some training-free algorithms for energy-guided sampling, such as reconstruction guidance (Ho et al., 2022c) and diffusion posterior sampling (Chung et al., 2022). The basic idea in these methods is to reuse the pretrained diffusion model in the data prediction formulation (Kingma et al., 2021):

$$\mathbb{E}_{q_{0t}(\boldsymbol{x}_0\vert\boldsymbol{x}_t)}[\boldsymbol{x}_0] \approx \hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t, t) := \frac{\boldsymbol{x}_t - \sigma_t\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\alpha_t},$$

and then define the intermediate energy function by:

$$f_\theta^{\text{DPS}}(\boldsymbol{x}_t, t) := \mathcal{E}_0(\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t, t)) \approx \mathcal{E}_0(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0\vert\boldsymbol{x}_t)}[\boldsymbol{x}_0]).$$

However, we show in Appendix E that $\nabla_{\boldsymbol{x}_t} f_\theta^{\text{DPS}} \neq \nabla_{\boldsymbol{x}_t}\mathcal{E}_t$ and thus it is also inexact.

2-D Example. We further compare the different methods for energy-guided sampling on a 2-D example, as shown in Fig. 2, and provide more 2-D results in Appendix L. Experiments show that our method outperforms all referred methods, especially when the inverse temperature β is large.

4.2. Relationship with Contrastive Learning and Classifier Guidance

In this section, we consider a special case of our method in which the energy function $\mathcal{E}_0(\boldsymbol{x}_0)$ is defined as negative log-likelihood $-\log q_0(c\vert\boldsymbol{x}_0)$ for a given conditioning variable c with $\beta = 1$. In such case, the desired distribution is:

$$p_0(\boldsymbol{x}_0) \propto q_0(\boldsymbol{x}_0)q(c\vert\boldsymbol{x}_0) \propto q(\boldsymbol{x}_0\vert c).$$

Different from the problem we consider in Eq. (7) that $p_0(\boldsymbol{x}_0)$ is hard to draw samples from, here we assume that we can draw samples from $p_0(\boldsymbol{x}_0) = q_0(\boldsymbol{x}_0\vert c)$. Following such an assumption, we prove in Appendix F that our proposed CEP in Eq. (12) is equivalent to

$$\mathbb{E}_{t,\boldsymbol{\epsilon}^{(1:K)}}\mathbb{E}_{\prod_{i=1}^K q_0(\boldsymbol{x}_0^{(i)}, c^{(i)})}\left[-\sum_{i=1}^K \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, c^{(i)}, t)}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)}, c^{(i)}, t)}}\right],$$
$$(15)$$

where $(\boldsymbol{x}_0^{(i)}, c^{(i)})$ are K paired data samples from $q_0(\boldsymbol{x}_0, c)$. Note that the inner expectation has the same form as the InfoNCE objective (Oord et al., 2018) and is widely used in contrastive learning for multi-modal data, such as CLIP (Radford et al., 2021) (where f_ϕ represents cosine similarity). Furthermore, Nichol et al. (2021) uses the above objective and trains a CLIP at each t for text-image pairs and uses its gradient as guidance for text-to-image sampling by diffusion models. Therefore, such guidance can be considered as a special case of CEP in Eq. (12), under the assumption that we can draw samples from $p_0(\boldsymbol{x}_0) = q_0(\boldsymbol{x}_0\vert c)$.

Moreover, if c is a discrete variable with a total of M possible values (classes). An alternative guided sampling method

for sampling from $q_0(\boldsymbol{x}_0|c)$ is classifier guidance (Dhariwal & Nichol, 2021), which optimize the following objective:

$$\mathbb{E}_{t,\boldsymbol{\epsilon}^{(1:K)}}\mathbb{E}_{\prod_{i=1}^{K}q_0(\boldsymbol{x}_0^{(i)},c^{(i)})}\left[-\sum_{i=1}^{K}\log\frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},c^{(i)},t)}}{\sum_{j=1}^{M}e^{-f_\phi(\boldsymbol{x}_t^{(i)},c^{(j)},t)}}\right],$$

$$(16)$$

The most notable difference between Eq. (15) and Eq. (16) is the normalizing axes in the training objective's denominator. Classifier guidance aims to *classify conditions* for a given data $\boldsymbol{x}_t^{(i)}$, so the objective could be understood as a classification loss which is normalized across all $c^{(j)}$. CEP is trying to *compare within data* for a specified condition $c^{(i)}$, so the objective could be understood as a contrastive loss which is normalized across all $\boldsymbol{x}_t^{(j)}$.

Theoretically, both classifier guidance and CEP could guarantee exact guidance given unlimited data and model capacity. Experimentally, we show in Sec. 6 that the two methods have quite similar performance. However, traditional classifier guidance cannot be applied to energy-guided sampling because there does not exist a set of conditions c across which we could normalize, whereas our proposed method can. We thus conclude that CEP could be considered as a contrastive alternative to classifier guidance for conditional sampling, but is in a more general form that could transfer to the energy-guided sampling problem.

5. Q-Guided Policy Optimization for Offline Reinforcement Learning

In this section, we showcase how our method can be applied in offline RL, including problem formulation in Section 5.1, algorithm method in Section 5.2 and Section 5.3, and experimental results in Section 5.4. A pseudocode is provided in Appendix I.1.

5.1. Problem Formulation

Recall that from Eq. (6), our desired policy π^* follows $\pi^*(\boldsymbol{a}|\boldsymbol{s}) \propto \mu(\boldsymbol{a}|\boldsymbol{s}) \, e^{\beta Q_\psi(\boldsymbol{s},\boldsymbol{a})}$, where the behavior policy $\mu(\boldsymbol{a}|\boldsymbol{s})$ is a diffusion model. In order to sample actions from π^* by diffusion sampling, we denote $\mu_0 := \mu$, $\pi_0 := \pi$, $\boldsymbol{a}_0 := \boldsymbol{a}$ at time $t = 0$. Then we construct a forward diffusion process to simultaneously diffuse μ_0 and π_0 into the same noise distribution, where $\pi_{t0}(\boldsymbol{a}_t|\boldsymbol{a}_0, \boldsymbol{s}) := \mu_{t0}(\boldsymbol{a}_t|\boldsymbol{a}_0, \boldsymbol{s}) = \mathcal{N}(\boldsymbol{a}_t|\alpha_t\boldsymbol{a}_0, \sigma_t^2\boldsymbol{I})$.

According to Theorem 3.1, by replacing the distribution q with μ, p with π, and the energy function $\mathcal{E}$ with $-Q$ following conventions in offline RL literature, we have the marginal distributions μ_t and π_t of the noise-perturbed action $\boldsymbol{a}_t$ satisfy:

$$\pi_t(\boldsymbol{a}_t|\boldsymbol{s}) \propto \mu_t(\boldsymbol{a}_t|\boldsymbol{s})e^{\mathcal{E}_t(\boldsymbol{s},\boldsymbol{a}_t)}. \qquad (17)$$

$\mathcal{E}_t(\boldsymbol{s}, \boldsymbol{a}_t)$ is an intermediate energy function determined by the learned action evaluation model $Q_\psi(\boldsymbol{s}, \boldsymbol{a}_0)$. Specifically $\mathcal{E}_t(\boldsymbol{s}, \boldsymbol{a}_t) = \log\mathbb{E}_{\mu_{0t}(\boldsymbol{a}_0|\boldsymbol{a}_t,\boldsymbol{s})}\left[e^{\beta Q_\psi(\boldsymbol{s},\boldsymbol{a}_0)}\right]$ and $\mathcal{E}_0(\boldsymbol{s}, \boldsymbol{a}_0) = \beta Q_\psi(\boldsymbol{s}, \boldsymbol{a}_0)$.

We now consider how to estimate the score function of $\pi_t(\boldsymbol{a}|\boldsymbol{s})$ such that we can sample actions from π_0 following Eq. (3). By Eq. (17), we have:

$$\nabla_{\boldsymbol{a}_t}\log\pi_t(\boldsymbol{a}_t|\boldsymbol{s}) = \underbrace{\nabla_{\boldsymbol{a}_t}\log\mu_t(\boldsymbol{a}_t|\boldsymbol{s})}_{\approx-\boldsymbol{\epsilon}_\theta(\boldsymbol{a}_t|\boldsymbol{s},t)/\sigma_t}+\underbrace{\nabla_{\boldsymbol{a}_t}\,\mathcal{E}_t(\boldsymbol{s},\boldsymbol{a}_t)}_{\approx f_\phi(\boldsymbol{s},\boldsymbol{a}_t,t)}\,.$$

$$(18)$$

To this end, we have formulated the classic constrained policy optimization problem (5) as energy-guided sampling, with $\nabla_{\boldsymbol{a}_t}\mathcal{E}_t(\boldsymbol{s}, \boldsymbol{a}_t)$ being the desired guidance. Because such guidance is determined by the Q function, we name this approach as *Q-guided policy optimization (QGPO)*. QGPO requires training a total of three neural networks in order to estimate the targeted score function $\nabla_{\boldsymbol{a}_t}\log\pi_t(\boldsymbol{a}_t|\boldsymbol{s})$: (1) a state-conditioned diffusion model $\boldsymbol{\epsilon}_\theta(\boldsymbol{a}_t|\boldsymbol{s}, t)$ to model the behavior policy $\mu(\boldsymbol{a}|\boldsymbol{s})$, for which we completely follow Chen et al. (2022); (2) an action evaluation model $Q_\psi(\boldsymbol{s}, \boldsymbol{a})$ to define the intermediate energy function $\mathcal{E}_t$ when $t = 0$ (Section 5.3); and (3) an energy model $f_\phi(\boldsymbol{s}, \boldsymbol{a}_t, t)$ to estimate $\mathcal{E}_t(\boldsymbol{s}, \boldsymbol{a}_t, t)$ and guide the diffusion sampling process when $t > 0$ (Section 5.2).

5.2. In-Support Contrastive Energy Prediction

Suppose we already have an action evaluation model $Q_\psi(\boldsymbol{s}, \boldsymbol{a})$ to estimate the Q-function $Q^\pi(\boldsymbol{s}, \boldsymbol{a})$. According to Theorem 3.2, $f_\phi(\boldsymbol{s}, \boldsymbol{a}_t, t)$ can be trained via our proposed CEP. Rewriting Eq. (12) to condition all distributions on state $\boldsymbol{s}$, the problem for learning f_ϕ becomes:

$$\min_\phi\mathbb{E}_{p(t)}\mathbb{E}_{\mu(\boldsymbol{s})}\mathbb{E}_{\prod_{i=1}^{K}\mu(\boldsymbol{a}^{(i)}|\boldsymbol{s})p(\boldsymbol{\epsilon}^{(i)})}\left[\vphantom{\sum_{i=1}^{K}}\right.$$

$$\left.-\sum_{i=1}^{K}\frac{e^{\beta Q_\psi(\boldsymbol{s},\boldsymbol{a}^{(i)})}}{\sum_{j=1}^{K}e^{\beta Q_\psi(\boldsymbol{s},\boldsymbol{a}^{(j)})}}\log\frac{e^{f_\phi(\boldsymbol{s},\boldsymbol{a}_t^{(i)},t)}}{\sum_{j=1}^{K}e^{f_\phi(\boldsymbol{s},\boldsymbol{a}_t^{(j)},t)}}\right],$$

$$(19)$$

where $t \sim \mathcal{U}(0, T)$, $\boldsymbol{a}_t = \alpha_t\boldsymbol{a} + \sigma_t\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$.

One difficulty in solving the above problem is that we have no access to the true distribution $\mu(\boldsymbol{a}|\boldsymbol{s})$ for a specified $\boldsymbol{s}$. Although we can sample data from the joint distribution $\mu(\boldsymbol{s}, \boldsymbol{a})$ or the marginal distribution $\mu(\boldsymbol{s})$ given the offline dataset $\mathcal{D}^\mu$, such data samples cannot be directly used to estimate the objective in problem (19). This is because we require $K > 1$ independent action samples from $\mu(\boldsymbol{a}|\boldsymbol{s})$ for a single $\boldsymbol{s}$ for contrastive learning, whereas we only have one such action in $\mathcal{D}^\mu$ given that $\boldsymbol{s}$ is a continuous variable.

To address this issue, we propose to pre-generate a support action set $\mathcal{D}^{\mu_\theta}$ using the already learned behavior model $\mu_\theta(\boldsymbol{a}_t|\boldsymbol{s}, t)$. Concretely, for each state $\boldsymbol{s}$ in the behavior

Dataset	Environment	CQL	BCQ	IQL	SfBC	DD	Diffuser	D-QL	D-QL@1	QGPO (ours)
Medium-Expert	HalfCheetah	62.4	64.7	86.7	**92.6**	90.6	79.8	**96.1**	**94.8**	93.5 ± 0.3
Medium-Expert	Hopper	98.7	100.9	91.5	**108.6**	**111.8**	**107.2**	**110.7**	100.6	$\textbf{108.0} \pm 2.5$
Medium-Expert	Walker2d	**111.0**	57.5	**109.6**	**109.8**	**108.8**	**108.4**	**109.7**	**108.9**	$\textbf{110.7} \pm 0.6$
Medium	HalfCheetah	44.4	40.7	47.4	45.9	49.1	44.2	50.6	47.8	$\textbf{54.1} \pm 0.4$
Medium	Hopper	58.0	54.5	66.3	57.1	79.3	58.5	82.4	64.1	$\textbf{98.0} \pm 2.6$
Medium	Walker2	79.2	53.1	78.3	77.9	**82.5**	79.7	**85.1**	82.0	$\textbf{86.0} \pm 0.7$
Medium-Replay	HalfCheetah	**46.2**	38.2	44.2	37.1	39.3	42.2	**47.5**	44.0	$\textbf{47.6} \pm 1.4$
Medium-Replay	Hopper	48.6	33.1	94.7	86.2	**100.0**	96.8	**100.7**	63.1	$\textbf{96.9} \pm 2.6$
Medium-Replay	Walker2d	26.7	15.0	73.9	65.1	75.0	61.2	**94.3**	75.4	84.4 ± 4.1
Average (Locomotion)		63.9	51.9	76.9	75.6	81.8	75.3	**86.3**	75.6	**86.6**
Default	AntMaze-umaze	74.0	78.9	87.5	**92.0**	-	-	68.6	69.4	$\textbf{96.4} \pm 1.4$
Diverse	AntMaze-umaze	**84.0**	55.0	62.2	**85.3**	-	-	53.0	56.4	74.4 ± 9.7
Play	AntMaze-medium	61.2	0.0	71.2	**81.3**	-	-	0.0	1.0	$\textbf{83.6} \pm 4.4$
Diverse	AntMaze-medium	53.7	0.0	70.0	**82.0**	-	-	18.4	14.8	$\textbf{83.8} \pm 3.5$
Play	AntMaze-large	15.8	6.7	39.6	59.3	-	-	10.6	15.8	$\textbf{66.6} \pm 9.8$
Diverse	AntMaze-large	14.9	2.2	47.5	45.5	-	-	4.2	1.6	$\textbf{64.8} \pm 5.5$
Average (AntMaze)		50.6	23.8	63.0	74.2	-	-	25.8	26.5	**78.3**
# Action candidates		1	100	1	32	1	1	50	1	1
# Diffusion steps		-	-	-	15	100	100	5	5	15

Table 2. Evaluation numbers of D4RL benchmarks (normalized as suggested by Fu et al. (2020)). We report mean and standard deviation of algorithm performance across 5 random seeds at the end of training. Numbers within 5 percent of the maximum in every individual task are highlighted. We rerun the experiments of Diffusion-QL to ensure a consistent evaluation metric. See Appendix I for details.

dataset $\mathcal{D}^\mu$, we sample K support actions $\{\hat{\boldsymbol{a}}^{(i)}\}_K$ from $\mu_\theta(\cdot|\boldsymbol{s})$ and store these actions in pair with the state $\boldsymbol{s}$ in $\mathcal{D}^{\mu_\theta}$. We then estimate the objective in (19) with $\mathcal{D}^{\mu_\theta}$:

$$\min_\phi \mathbb{E}_{t,\boldsymbol{s},\boldsymbol{\epsilon}} -\sum_{i=1}^{K} \frac{e^{\beta Q_\psi(\boldsymbol{s},\hat{\boldsymbol{a}}^{(i)})}}{\sum_{j=1}^{K} e^{\beta Q_\psi(\boldsymbol{s},\hat{\boldsymbol{a}}^{(j)})}} \log \frac{e^{f_\phi(\boldsymbol{s},\hat{\boldsymbol{a}}_t^{(i)},t)}}{\sum_{j=1}^{K} e^{f_\phi(\boldsymbol{s},\hat{\boldsymbol{a}}_t^{(j)},t)}}$$
$$(20)$$

where $\hat{\boldsymbol{a}}^{(i)}$, $\hat{\boldsymbol{a}}^{(j)}$ are support actions for $\mathcal{D}^{\mu_\theta}(\boldsymbol{s})$. Since problem (20) is optimized in a support action set instead of the true dataset, we refer to it as in-support CEP.

5.3. In-support Softmax Q-Learning

We now discuss in detail how the action evaluation model $Q_\psi \approx Q^\pi$ could be trained. Ideally, we can use a typical Bellman-style bootstrapping method to calculate the mean square error (MSE) training target of Q_ψ (Wang et al., 2022b; Goo & Niekum, 2022):

$$\mathcal{T}^\pi Q_\psi(\boldsymbol{s},\boldsymbol{a}) = r(\boldsymbol{s},\boldsymbol{a}) + \gamma \mathbb{E}_{\boldsymbol{s}' \sim P(\cdot|\boldsymbol{s},\boldsymbol{a}), \boldsymbol{a}' \sim \pi(\cdot|\boldsymbol{s}')} Q_\psi(\boldsymbol{s}',\boldsymbol{a}').$$
$$(21)$$

However, calculating $\mathcal{T}^\pi Q_\psi(\boldsymbol{s},\boldsymbol{a})$ could in practice be time-consuming, because it requires sampling from a diffusion model π during training. We thus leverage the generated support action set $\mathcal{D}^{\mu_\theta}$ to avoid repeated sampling from a diffusion model. Specifically, we estimate $\mathcal{T}^\pi Q_\psi(\boldsymbol{s},\boldsymbol{a})$ via importance sampling:

$$\mathcal{T}^\pi Q_\psi(\boldsymbol{s},\boldsymbol{a}) \approx r(\boldsymbol{s},\boldsymbol{a}) + \gamma \frac{\sum_{\hat{\boldsymbol{a}}'} e^{\beta_Q Q_\psi(\boldsymbol{s}',\hat{\boldsymbol{a}}')} Q_\psi(\boldsymbol{s}',\hat{\boldsymbol{a}}')}{\sum_{\hat{\boldsymbol{a}}'} e^{\beta_Q Q_\psi(\boldsymbol{s}',\hat{\boldsymbol{a}}')}}.$$
$$(22)$$

5.4. Results

We compare the performance of QGPO with several related works in multiple D4RL (Fu et al., 2020) tasks in Table 2. Among them, `MuJoCo locomotion` tasks are popular benchmarks in offline RL and mainly aim to drive different robots moving forward as fast as possible. The dataset might contain a mixture of expert-level and medium-level policies' decision data (Medium-Expert), decision data generated by a single medium-level policy (Medium), and diverse decision data generated by a large set of medium-level policies (Medium-Replay). `Antmaze` tasks are typically considered to be hard tasks for RL-based methods. They aim to navigate an ant robot in several prespecified mazes (Umaze, Medium, Large). The learned policy directly outputs an eight-dimensional motor torque to control the motor motion of the ant robot at each degree of freedom. As a result, Antmaze tasks require policies to perform both low-level motion control and high-level navigation.

From Table 2, we can see that in most tasks, our method outperforms referenced baselines, especially in difficult tasks such as Antmaze-Large. Baselines include traditional state-of-the-art algorithms like CQL (Kumar et al., 2020), BCQ (Fujimoto et al., 2019), and IQL (Kostrikov et al., 2022), which adopt Gaussian-like policies. We also include recent advances in offline RL that adopt diffusion-based policies. Diffusers (Janner et al., 2022) considers using an energy guidance method that ensembles the MSE-based method as described in Eq. (14). Decision Diffuser (DD, Ajay et al. (2022)), on the other hand, explores using the classifier-free

Conditional	Resolution	Diffusion Steps	FID	sFID	Precision	Recall
✓	128×128	250	3.17 / 2.97	5.17 / 5.09	0.78 / 0.78	0.59 / 0.59
✓	128×128	25	6.15 / 5.98	6.97 / 7.04	0.79 / 0.78	0.51 / 0.51
✓	256×256	250	4.74 / 4.59	5.23 / 5.25	0.82 / 0.82	0.52 / 0.52
✓	256×256	25	5.58 / 5.44	5.25 / 5.32	0.82 / 0.81	0.48 / 0.49
✗	256×256	250	32.53 / 33.03	7.23 / 6.99	0.56 / 0.56	0.65 / 0.65

Table 3. Effect of CEP guidance (left) on image sample quality compared with classifier guidance (right).

guidance (Ho & Salimans, 2021). Diffusion-QL (D-QL, Wang et al. (2022b)) tracks the gradients of the actions sampled from the behavior diffusion policy to guide generated actions to high Q-value area. SfBC (Chen et al., 2022) simply resamples actions from multiple behavior action candidates using the predicted Q-value as sampling weights. Note that such a resampling trick is also shared by Diffusion-QL. We also study a variant of Diffusion-QL (D-QL@1) where the resampling trick is removed to better reflect the quality of decisions generated by the diffusion policy.

6. Image Synthesis Examples with CEP

6.1. Results in Class-Conditional Image Synthesis

We quantitatively evaluate our proposed method (Eq. (15)) in image synthesis tasks on ImageNet as is shown in Table 3. Our method achieves results that are roughly on par with classic classifier guidance (Dhariwal & Nichol, 2021). We also qualitatively compare sampled images guided by our proposed CEP guidance and classifier guidance with fixed random seeds in Appendix O, which shows that they generate samples that are almost visually identical. Besides the training objective, our method uses exactly the same network architecture, training pipeline, and evaluation methods as Dhariwal & Nichol (2021), without any kind of hyperparameter tuning. These results empirically indicate that our method is an almost equally well-performing alternative to classifier guidance in ImageNet image synthesis tasks.

6.2. Energy-Guided Image Synthesis

This section showcases how image synthesis can be controlled through a continuous energy function as described in Eq. (1) instead of a discrete class condition as in Section 6.1. We define the energy function at $t = 0$ data space to indicate the overall color appearance of an image:

$$\mathcal{E}(\boldsymbol{x}) := -\overline{\|h(\boldsymbol{x}) - h_{\mathbf{tar}}\|_1}, \qquad (23)$$

where $h(\boldsymbol{x})$ represents the hue value for each pixel in an image $\boldsymbol{x}$, and can be calculated via Hue-Saturation-Intensity (HSI) decomposition (Shapiro et al., 2001). $h(\boldsymbol{x})$ is defined in an angular space of range $[0, 2\pi]$, where red is at angle 0, green at $2\pi/3$, blue at $4\pi/3$, and red again at 2π. As a result, by setting the target hue $h_{\mathbf{tar}}$ to corresponding angular values, we can evaluate how an image is visually close to a
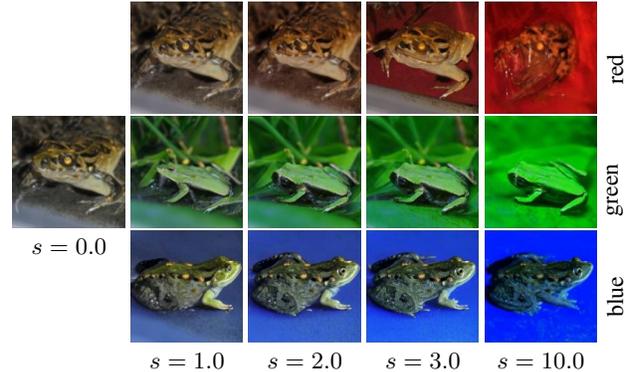
Figure 3. Samples by color guidance with red, green, and blue, varying the guidance scale s (under a fixed random seed).

"pure color" image.

With such a definition of the energy function, we train three energy guidance models to control the overall color appearance of sampled images. An illustration is given in Figure 3. By switching among different guidance models and tuning guidance scales to control the guidance effect similar to Dhariwal & Nichol (2021); Ho & Salimans (2021), we can effectively control the color appearance of an image. If the diffusion prior is a conditional model, generated images might have different backgrounds (e.g., desert, forest, and sky) to meet different preferences of color appearance while ensuring fidelity. See more examples in Appendix N.

7. Conclusion

In this work, we formally consider the problem of sampling by diffusion models pretrained on a data distribution but the target sampling distribution is edited by an energy function. We show that this can be achieved by adding additional energy guidance to the original sampling procedure. We further propose a novel training objective named contrastive energy prediction for training an energy model to estimate such guidance. Our proposed CEP guidance is exact compared with previous energy guidance methods in the sense that it can guarantee convergence to the desired distribution. We apply our proposed method to several downstream tasks in order to demonstrate its effectiveness and scalability. Experimental results show that our method outperforms existing guidance methods in offline RL and is roughly on par with the classic classifier guidance in conditional image synthesis.

Acknowledgements

References

Ajay, A., Du, Y., Gupta, A., Tenenbaum, J., Jaakkola, T., and Agrawal, P. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Bao, F., Li, C., Zhu, J., and Zhang, B. Analytic-DPM: An analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022a.

Bao, F., Zhao, M., Hao, Z., Li, P., Li, C., and Zhu, J. Equivariant energy-guided sde for inverse molecular design. *arXiv preprint arXiv:2209.15408*, 2022b.

Chen, H., Lu, C., Ying, C., Su, H., and Zhu, J. Offline reinforcement learning via high-fidelity generative behavior modeling. *arXiv preprint arXiv:2209.14548*, 2022.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021a.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., Dehak, N., and Chan, W. Wavegrad 2: Iterative refinement for text-to-speech synthesis. In *International Speech Communication Association*, pp. 3765–3769, 2021b.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

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

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/fujimoto18a.html.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 09–15 Jun 2019.

Goo, W. and Niekum, S. Know your boundaries: The necessity of explicit behavioral cloning in offline rl. *arXiv preprint arXiv:2206.00695*, 2022.

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.

Graikos, A., Malkin, N., Jojic, N., and Samaras, D. Diffusion models as plug-and-play priors. *arXiv preprint arXiv:2206.09012*, 2022.

Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., and Guo, B. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10696–10706, 2022.

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. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

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

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022b.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022c.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3D. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Janner, M., Du, Y., Tenenbaum, J., and Levine, S. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

Kawar, B., Elad, M., Ermon, S., and Song, J. Denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793*, 2022.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. In *Advances in Neural Information Processing Systems*, 2021.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

Kumar, A., Fu, J., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *CoRR*, abs/1906.00949, 2019. URL http://arxiv.org/abs/1906.00949.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative Q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

Kwon, G. and Ye, J. C. Diffusion-based image translation using disentangled style and content representation. *arXiv preprint arXiv:2209.15264*, 2022.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. B. Diffusion-LM improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N. M. O., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *ICLR*, 2016.

Liu, J., Li, C., Ren, Y., Chen, F., and Zhao, Z. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11020–11028, 2022.

Lu, C., Zheng, K., Bao, F., Chen, J., Li, C., and Zhu, J. Maximum likelihood training for score-based diffusion odes by high order denoising score matching. In *International Conference on Machine Learning*, pp. 14429–14460. PMLR, 2022a.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022b.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022c.

Meng, C., Gao, R., Kingma, D. P., Ermon, S., Ho, J., and Salimans, T. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022a.

Meng, C., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. SDEdit: Image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022b.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Nie, W., Vahdat, A., and Anandkumar, A. Controllable and compositional generation with latent-space energy-based models. *Advances in Neural Information Processing Systems*, 34:13497–13510, 2021.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Pandey, K., Mukherjee, A., Rai, P., and Kumar, A. DiffuseVAE: Efficient, controllable and high-fidelity generation from low-dimensional latents. *arXiv preprint arXiv:2201.00308*, 2022.

Pearce, T., Rashid, T., Kanervisto, A., Bignell, D., Sun, M., Georgescu, R., Macua, S. V., Tan, S. Z., Momennejad, I., Hofmann, K., et al. Imitating human behaviour with diffusion models. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Peters, J., Mulling, K., and Altun, Y. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10, 2022a.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022b.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Shapiro, L. G., Stockman, G. C., et al. *Computer vision*, volume 3. Prentice Hall New Jersey, 2001.

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, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

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

Theis, L., Salimans, T., Hoffman, M. D., and Mentzer, F. Lossy compression with gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022.

Wang, T., Zhang, B., Zhang, T., Gu, S., Bao, J., Baltrusaitis, T., Shen, J., Chen, D., Wen, F., Chen, Q., et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. *arXiv preprint arXiv:2212.06135*, 2022a.

Wang, Z., Novikov, A., Zolna, K., Merel, J. S., Springenberg, J. T., Reed, S. E., Shahriari, B., Siegel, N., Gulcehre, C., Heess, N., and de Freitas, N. Critic regularized regression. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7768–7778. Curran Associates, Inc., 2020.

Wang, Z., Hunt, J. J., and Zhou, M. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022b.

Wu, L., Gong, C., Liu, X., Ye, M., and Liu, Q. Diffusion-based molecule generation with informative prior bridges. *arXiv preprint arXiv:2209.00865*, 2022.

Xiao, Z., Yan, Q., and Amit, Y. Exponential tilting of generative models: Improving sample quality by training and sampling from latent energy. *arXiv preprint arXiv:2006.08100*, 2020.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Yang, R., Srivastava, P., and Mandt, S. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022.

Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., and Kreis, K. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022.

Zhang, Q. and Chen, Y. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

Zhang, Q., Tao, M., and Chen, Y. gddim: Generalized denoising diffusion implicit models. *arXiv preprint arXiv:2206.05564*, 2022.

Zhao, M., Bao, F., Li, C., and Zhu, J. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. *arXiv preprint arXiv:2207.06635*, 2022.

Zhong, Y., Liu, H., Liu, X., Bao, F., Shen, W., and Li, C. Deep generative modeling on limited data with regularization by nontransferable pre-trained models. *arXiv preprint arXiv:2208.14133*, 2022.

Zhou, D., Wang, W., Yan, H., Lv, W., Zhu, Y., and Feng, J. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

Zhu, J., Chen, N., and Xing, E. P. Bayesian inference with posterior regularization and applications to infinite latent svms. *The Journal of Machine Learning Research*, 15(1): 1799–1847, 2014.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A. Limitations and broader impacts

Similar to other deep generative modeling methods, energy-guided diffusion sampling can be potentially used to generate harmful contents such as "deepfakes", and might reflect and amplify unwanted social bias existed in the training dataset.

B. Related Work

Diffusion Models and Applications.   Diffusion models (also as known as score-based generative models) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b; Karras et al., 2022) are emerging powerful generative models and have achieved impressive success on various tasks, such as voice synthesis (Liu et al., 2022; Chen et al., 2021a;b), high-resolution image synthesis (Dhariwal & Nichol, 2021; Ho et al., 2022b), image editing (Meng et al., 2022b; Saharia et al., 2022a; Zhao et al., 2022), text-to-image generation (Saharia et al., 2022b; Nichol et al., 2021; Ramesh et al., 2022; Rombach et al., 2022; Gu et al., 2022), molecule generation (Xu et al., 2022; Hoogeboom et al., 2022; Wu et al., 2022), 3-D shape generation (Zeng et al., 2022; Poole et al., 2022; Wang et al., 2022a), video generation (Ho et al., 2022c;a; Yang et al., 2022; Zhou et al., 2022) and data compression (Theis et al., 2022; Kingma et al., 2021; Lu et al., 2022a). The sampling methods for diffusion models include training-free fast samplers (Song et al., 2021a; Bao et al., 2022a; Lu et al., 2022b;c; Zhang & Chen, 2022; Zhang et al., 2022) and distillation-based samplers (Salimans & Ho, 2022; Meng et al., 2022a).

Diffusion Models as Priors.   There exist many existing works that use a pretrained diffusion model as the prior distribution $q(\boldsymbol{x})$ and aim to sample from Eq. (1). Graikos et al. (2022) propose a training-free sampling method which directly use the constraint ($\mathcal{E}(\cdot)$ in Eq. (1)) and can be used for approximately solving traveling salesman problems; Poole et al. (2022) use a pretrained 2-D diffusion model and optimizing the 3-D parameters for 3-D shape generation; Kawar et al. (2022); Chung et al. (2022) use pretrained diffusion models to solve linear and some special non-linear inverse problems, such as image inpainting, deblurring and denoising; Zhao et al. (2022); Bao et al. (2022b) use human-designed intermediate energy guidance for image-to-image translation and inverse molecular design. However, all these existing methods cannot guarantee that the final samples follow the desired $p(\boldsymbol{x})$ in Eq. (1).

Controllable Generation.   To embed human preference and controllability into the sampling procedure of deep generative models, many recent work (Nie et al., 2021; Li et al., 2022; Pandey et al., 2022) manipulate the sampling procedure of the latent space by some learned conditional models, and use the obtained latent code to generate desired samples. The generator include variational auto-encoder (Kingma & Welling, 2014) and generative adversarial networks (Goodfellow et al., 2020). However, existing methods for realizing controllable generation in diffusion models mainly focus on conditional guidance. The problem we considered in Eq. (1) can be alternatively understood as a general formulation for embed human controllability into the sampling procedure of diffusion models.

Offline Reinforcement Learning.   Offline RL typically requires reconciling two conflicting aims: Staying close to the behavior policy while maximizing the expected Q-values. In order to stick with a potentially diverse behavior policy, recent studies (Janner et al., 2022; Goo & Niekum, 2022; Wang et al., 2022b; Chen et al., 2022; Ajay et al., 2022; Pearce et al., 2022) have found diffusion models to be a powerful generative tool, which tends to outperform previous generative methods such as Gaussians (Peng et al., 2019; Wang et al., 2020; Nair et al., 2020) or VAEs (Fujimoto et al., 2019; Kumar et al., 2019) in terms of behavior modeling. In terms of how to generate actions that maximize the learned Q-functions, different methods take different approaches. Diffusers (Janner et al., 2022) intends to mimic the classifier-guidance method (Dhariwal & Nichol, 2021) and propose to use a guidance method as described in Eq. (14), but without a detailed discussion on the convergence property of the proposed method. Decision Diffuser (DD, Ajay et al. (2022)), on the other hand, explores using classifier-free guidance. Diffusion-QL tracks the gradients of the actions sampled from the behavior diffusion policy to guide generated actions to high Q-value area. SfBC (Chen et al., 2022) and Diffusion-QL share the same trick by simply resampling actions from multiple behavior action candidates using the predicted Q-value as sampling weights. Other work (Goo & Niekum, 2022; Pearce et al., 2022) only uses diffusion models for pure behavior cloning, so no Q-value maximizing is required. In contrast with prior work, our work aims to study how an energy guidance model could be exactly trained and used to guide the sampling process in diffusion-based decision-making.

C. Motivation for Eq. (1)

The formulation in Eq. (1) is general and common across various settings, such as the product of experts by Gibbs distribution (Hinton, 2002), posterior-regularized Bayesian inference (Zhu et al., 2014), exponential tilting of generative models (Xiao et al., 2020), training deep generative models on limited data with regularization by pre-trained models (Zhong et al., 2022), constrained policy optimization in reinforcement learning (Peng et al., 2019; Ziegler et al., 2019), and inverse problems (Chung et al., 2022). In this section, we give a motivating example for such an objective.

Let $q(\boldsymbol{x})$ be an unknown data distribution, and $\mathcal{E}(\boldsymbol{x})$ be a loss function that we want to minimize. We want to optimize a generative model $p(\boldsymbol{x})$ such that the samples from $p(\boldsymbol{x})$ have as small loss $\mathcal{E}(\boldsymbol{x})$ as possible; meanwhile, we also use the data distribution $q(\boldsymbol{x})$ to regularize the model $p(\boldsymbol{x})$ to increase the diversity and avoid collapse solutions. The objective can be formulated by:

$$\min_p \mathbb{E}_{p(\boldsymbol{x})}[\mathcal{E}(\boldsymbol{x})] + \frac{1}{\beta} D_{\mathrm{KL}}(p(\boldsymbol{x}) \parallel q(\boldsymbol{x})), \tag{24}$$

where $D_{\mathrm{KL}}(p(\boldsymbol{x}) \parallel q(\boldsymbol{x}))$ is a regularization term and β is a hyperparameter. By simply computing the derivation for p and letting it be zero, we can obtain the optimal p^* satisfies

$$p^*(\boldsymbol{x}) \propto q(\boldsymbol{x}) e^{-\mathcal{E}(\boldsymbol{x})}, \tag{25}$$

which has the exactly same form as Eq. (1).

D. Proofs and Additional Theory

D.1. CEP with Condition Variables

Assume the energy function is $\mathcal{E}(\boldsymbol{x}_0, c)$ with an additional conditioning variable c, which follows a distribution $q(c)$. We aim to learn the intermediate energy guidance by a neural network $f_\phi(\cdot, c, t) : \mathbb{R}^d \to \mathbb{R}$ parameterized by ϕ.

CEP with unconditional prior. Assume the prior distribution $q_0(\boldsymbol{x})$ is unconditional. We aim to sample from

$$p_0(\boldsymbol{x}_0|c) \propto q_0(\boldsymbol{x}_0) e^{-\beta \mathcal{E}(\boldsymbol{x}_0, c)}, \tag{26}$$

for each given c. By taking the expectation of $q(c)$, the objective in Eq. (12) is

$$\min_\phi \mathbb{E}_{q(c)} \mathbb{E}_{p(t)} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[-\sum_{i=1}^K e^{-\beta \mathcal{E}(\boldsymbol{x}_0^{(i)}, c)} \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, c, t)}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)}, c, t)}} \right]. \tag{27}$$

CEP with conditional prior. Assume the prior distribution $q_0(\boldsymbol{x}|c)$ is conditional on c. We aim to sample from

$$p_0(\boldsymbol{x}_0|c) \propto q_0(\boldsymbol{x}_0|c) e^{-\beta \mathcal{E}(\boldsymbol{x}_0, c)}, \tag{28}$$

for each given c. By taking the expectation of $q(c)$, the objective in Eq. (12) is

$$\min_\phi \mathbb{E}_{q(c)} \mathbb{E}_{p(t)} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)}|c)} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[-\sum_{i=1}^K e^{-\beta \mathcal{E}(\boldsymbol{x}_0^{(i)}, c)} \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, c, t)}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)}, c, t)}} \right]. \tag{29}$$

Moreover, we can draw K sample pairs $(\boldsymbol{x}_0^{(i)}, c^{(i)})$ for $i = 1, \ldots, K$, and the above objective becomes

$$\min_\phi \mathbb{E}_{p(t)} \mathbb{E}_{\prod_{i=1}^K q_0(\boldsymbol{x}_0^{(i)}, c^{(i)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[-\sum_{i=1}^K e^{-\beta \mathcal{E}(\boldsymbol{x}_0^{(i)}, c^{(i)})} \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, c^{(i)}, t)}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)}, c^{(i)}, t)}} \right]. \tag{30}$$

Note that the terms in the numerator is $(\boldsymbol{x}_t^{(i)}, c^{(i)}) \sim q_t(\boldsymbol{x}_t^{(i)}, c^{(i)})$ are samples from the joint distribution; while the terms in the denominator is $(\boldsymbol{x}_t^{(j)}, c^{(i)}, t) \sim q_t(\boldsymbol{x}_t^{(j)}) q(c^{(i)})$ are independent samples. Such formulation is highly similar to the contrastive learning objective (Oord et al., 2018), and we discuss the connections in Appendix F. In summary, CEP with conditional prior can be considered as a generalized version of contrastive learning with soft energy labels.

D.2. CEP in Multiple Time Steps

Training the guidance model by Eq. (12) needs K samples of $\boldsymbol{x}_t$ for each time t. If we use M samples of $t \in (0, T]$, the number of total samples of $\boldsymbol{x}_t$ is KM. However, for high-dimensional data such as images, the memory budget is limited and we want to use as many samples at each time t as possible. In this section, we propose an alternative objective that can leverage K samples x_t from different time t. Thus, we only need K samples of t and K samples of $\boldsymbol{x}_t$ to reduce the memory cost. We formally propose the objective below and provide the proof in Appendix D.5.

Theorem D.1 (CEP in Multiple Time Steps). *Let $t^{(1)}, \ldots, t^{(K)}$ be K i.i.d. samples from $p(t)$. For each $i = 1, \ldots, K$, let $\boldsymbol{x}_t^{(i)} := \alpha_{t^{(i)}} \boldsymbol{x}_0^{(i)} + \sigma_{t^{(i)}} \boldsymbol{\epsilon}^{(i)}$, where α_t, σ_t are defined in Eq. (8). Define an objective:*

$$\min_{\phi} \mathbb{E}_{p(t^{(1:K)})} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[- \sum_{i=1}^{K} \underbrace{\left(e^{-\beta \mathcal{E}(\boldsymbol{x}_0^{(i)})} \right)}_{\text{energy label}} \log \underbrace{\frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, t^{(i)})}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)}, t^{(j)})}}}_{\text{predicted label}} \right]. \tag{31}$$

Given unlimited model capacity and data samples, For all $K > 1$ and $t \in [0, T]$, the optimal f_{ϕ^*} in Eq. (31) satisfies

$$\nabla_{\boldsymbol{x}_t} f_{\phi^*}(\boldsymbol{x}_t, t) = \nabla_{\boldsymbol{x}_t} \mathcal{E}_t(\boldsymbol{x}_t). \tag{32}$$

Below we also give the corresponding objectives for energy functions with conditioning variables.

CEP with unconditional prior in multiple time steps. The objective is

$$\min_{\phi} \mathbb{E}_{q(c)} \mathbb{E}_{p(t^{(1:K)})} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[- \sum_{i=1}^{K} e^{-\beta \mathcal{E}(\boldsymbol{x}_0^{(i)}, c)} \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, c, t^{(i)})}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)}, c, t^{(j)})}} \right]. \tag{33}$$

CEP with conditional prior in multiple time steps.

$$\min_{\phi} \mathbb{E}_{p(t^{(1:K)})} \mathbb{E}_{\prod_{i=1}^{K} q_0(\boldsymbol{x}_0^{(i)}, c^{(i)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[- \sum_{i=1}^{K} e^{-\beta \mathcal{E}(\boldsymbol{x}_0^{(i)}, c^{(i)})} \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, c^{(i)}, t^{(i)})}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)}, c^{(i)}, t^{(j)})}} \right]. \tag{34}$$

Also note that the terms in the numerator is $(\boldsymbol{x}_t^{(i)}, c^{(i)}, t^{(i)}) \sim q(\boldsymbol{x}_t^{(i)}, c^{(i)}, t^{(i)})$ are samples from the joint distribution; while the terms in the denominator is $(\boldsymbol{x}_t^{(j)}, c^{(i)}, t^{(j)}) \sim q(\boldsymbol{x}_t^{(j)}, t^{(j)}) q(c^{(i)})$ are independent samples.

D.3. Proof of Theorem 3.1

Proof. Assume the normalizing constant for $p_0(\boldsymbol{x}_0)$ is

$$Z = \int q_0(\boldsymbol{x}_0) e^{-\beta \mathcal{E}(\boldsymbol{x}_0)} \mathrm{d}\boldsymbol{x}_0 = \mathbb{E}_{q_0(\boldsymbol{x}_0)} \left[e^{-\beta \mathcal{E}(\boldsymbol{x}_0)} \right],$$

then we have

$$p_0(\boldsymbol{x}_0) = \frac{q_0(\boldsymbol{x}_0) e^{-\beta \mathcal{E}(\boldsymbol{x}_0)}}{Z}$$

According to the definition, we have

$$
\begin{aligned}
p_t(\boldsymbol{x}_t) &= \int p_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0)p_0(\boldsymbol{x}_0)\mathrm{d}\boldsymbol{x}_0 \\
&= \int p_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0)q_0(\boldsymbol{x}_0)\frac{e^{-\beta\mathcal{E}(\boldsymbol{x}_0)}}{Z}\mathrm{d}\boldsymbol{x}_0 \\
&= \int q_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0)q_0(\boldsymbol{x}_0)\frac{e^{-\beta\mathcal{E}(\boldsymbol{x}_0)}}{Z}\mathrm{d}\boldsymbol{x}_0 \\
&= q_t(\boldsymbol{x}_t)\int q_0(\boldsymbol{x}_0|\boldsymbol{x}_t)\frac{e^{-\beta\mathcal{E}(\boldsymbol{x}_0)}}{Z}\mathrm{d}\boldsymbol{x}_0 \\
&= \frac{q_t(\boldsymbol{x}_t)\mathbb{E}_{q_0(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[e^{-\beta\mathcal{E}(\boldsymbol{x}_0)}\right]}{Z} \\
&= \frac{q_t(\boldsymbol{x}_t)e^{-\mathcal{E}_t(\boldsymbol{x}_t)}}{Z}
\end{aligned}
$$

and then

$$
\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t}\mathcal{E}_t(\boldsymbol{x}_t)
$$

$\square$

D.4. Proof of Theorem 3.2

Proof. As $q_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\alpha_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I})$, we can rewrite Eq. (12) by

$$
\min_\phi \mathbb{E}_{p(t)}\mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})}\mathbb{E}_{q_{t0}(\boldsymbol{x}_t^{(1:K)}|\boldsymbol{x}_0^{(1:K)})}\left[-\sum_{i=1}^K e^{-\beta\mathcal{E}(\boldsymbol{x}_0^{(i)})}\log\frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},t)}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)},t)}}\right].
$$

Rewriting $q_{0t}(\boldsymbol{x}_0, \boldsymbol{x}_t) = q_t(\boldsymbol{x}_t)q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)$ and moving the conditional expectation $q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)$ into the inner part, we have

$$
\min_\phi \mathbb{E}_{p(t)}\mathbb{E}_{q_t(\boldsymbol{x}_t^{(1:K)})}\left[-\sum_{i=1}^K \mathbb{E}_{q_{0t}(\boldsymbol{x}_0^{(i)}|\boldsymbol{x}_t^{(i)})}\left[e^{-\beta\mathcal{E}(\boldsymbol{x}_0^{(i)})}\right]\log\frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},t)}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)},t)}}\right].
$$

By Eq. (9), we have $\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[e^{-\beta\mathcal{E}(\boldsymbol{x}_0)}\right] = e^{-\mathcal{E}_t(\boldsymbol{x}_t)}$, thus the above objective is equivalent to

$$
\min_\phi \mathbb{E}_{p(t)}\mathbb{E}_{q_t(\boldsymbol{x}_t^{(1:K)})}\left[-\sum_{i=1}^K e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(i)})}\log\frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},t)}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)},t)}}\right]. \tag{35}
$$

For each t and $\boldsymbol{x}_t^{(1:K)}$, for $i = 1, \ldots, K$, define

$$
a_i(\boldsymbol{x}_t^{(1:K)}, t) := \frac{e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(i)})}}{\sum_{j=1}^K e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(j)})}},
$$

$$
b_i(\boldsymbol{x}_t^{(1:K)}, t) := \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},t)}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)},t)}},
$$

$$
c(\boldsymbol{x}_t^{(1:K)}, t) := \sum_{j=1}^K e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(j)})} > 0,
$$

Then Eq. (35) is equivalent to

$$
\min_\phi \mathbb{E}_{p(t)}\mathbb{E}_{q_t(\boldsymbol{x}_t^{(1:K)})}\left[-c(\boldsymbol{x}_t^{(1:K)}, t)\sum_{i=1}^K a_i(\boldsymbol{x}_t^{(1:K)}, t)\log\left(b_i(\boldsymbol{x}_t^{(1:K)}, t)\right)\right].
$$

For each fixed t and $\boldsymbol{x}_t^{(!:K)}$, as $\sum_{i=1}^{K} a_i(\boldsymbol{x}_t^{(1:K)}, t) = \sum_{i=1}^{K} b_i(\boldsymbol{x}_t^{(1:K)}, t) = 1$, according to Gibbs' inequality, we have

$$-\sum_{i=1}^{K} a_i(\boldsymbol{x}_t^{(1:K)}, t) \log\left(b_i(\boldsymbol{x}_t^{(1:K)}, t)\right) \geq -\sum_{i=1}^{K} a_i(\boldsymbol{x}_t^{(1:K)}, t) \log\left(a_i(\boldsymbol{x}_t^{(1:K)}, t)\right),$$

so we have

$$\mathbb{E}_{p(t)}\mathbb{E}_{q_t(\boldsymbol{x}_t^{(1:K)})}\left[-c(\boldsymbol{x}_t^{(1:K)}, t)\sum_{i=1}^{K} a_i(\boldsymbol{x}_t^{(1:K)}, t) \log\left(b_i(\boldsymbol{x}_t^{(1:K)}, t)\right)\right]$$

$$\geq \mathbb{E}_{p(t)}\mathbb{E}_{q_t(\boldsymbol{x}_t^{(1:K)})}\left[-c(\boldsymbol{x}_t^{(1:K)}, t)\sum_{i=1}^{K} a_i(\boldsymbol{x}_t^{(1:K)}, t) \log\left(a_i(\boldsymbol{x}_t^{(1:K)}, t)\right)\right],$$

and the equality holds when for each $t \in (0, T]$ and each $\boldsymbol{x}_t^{(1:K)}$ in the supported space of $q_t(\boldsymbol{x}_t^{(1:K)})$,

$$b_i(\boldsymbol{x}_t^{(1:K)}, t) = a_i(\boldsymbol{x}_t^{(1:K)}, t), \quad i = 1, \dots, K,$$

so given unlimited data and model capacity, the optimal ϕ^* satisfies

$$\frac{e^{-f_{\phi^*}(\boldsymbol{x}_t^{(i)}, t)}}{\sum_{j=1}^{K} e^{-f_{\phi^*}(\boldsymbol{x}_t^{(j)}, t)}} = \frac{e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(i)})}}{\sum_{j=1}^{K} e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(j)})}},$$

so for each $i = 1, \dots, K$,

$$\frac{e^{-f_{\phi^*}(\boldsymbol{x}_t^{(i)}, t)}}{e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(i)})}} = \frac{\sum_{j=1}^{K} e^{-f_{\phi^*}(\boldsymbol{x}_t^{(j)}, t)}}{\sum_{j=1}^{K} e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(j)})}},$$

due to the arbitrariness of $\boldsymbol{x}_t^{(1:K)}$ and t, for any $\boldsymbol{x}_t^{(i)}, \boldsymbol{x}_t^{(j)}$, we have

$$\frac{e^{-f_{\phi^*}(\boldsymbol{x}_t^{(i)}, t)}}{e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(i)})}} = \frac{e^{-f_{\phi^*}(\boldsymbol{x}_t^{(j)}, t)}}{e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(j)})}}.$$

Therefore, there exists a constant C_t independent of $\boldsymbol{x}_t$, such that

$$e^{-f_{\phi^*}(\boldsymbol{x}_t, t)} = C_t \cdot e^{-\mathcal{E}_t(\boldsymbol{x}_t)} \propto e^{-\mathcal{E}_t(\boldsymbol{x}_t)},$$

and then $\nabla_{\boldsymbol{x}_t} f_{\phi^*}(\boldsymbol{x}_t, t) = \nabla_{\boldsymbol{x}_t} \mathcal{E}_t(\boldsymbol{x}_t)$. $\qquad\square$

D.5. Proof of Theorem D.1

Intuitively, Theorem D.1 can be similarly proved as Theorem 3.2 by considering $(\boldsymbol{x}_t, t)$ as a whole random variable. We formally give the proof below.

Proof of Theorem D.1. As $q_{t0}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\alpha_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I})$, we can rewrite Eq. (31) by

$$\min_{\phi} \mathbb{E}_{q(t^{(1:K)})}\mathbb{E}_{q(\boldsymbol{x}_0^{(1:K)}, \boldsymbol{x}_t^{(1:K)})}\left[-\sum_{i=1}^{K} e^{-\beta\mathcal{E}(\boldsymbol{x}_0^{(i)})} \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, t^{(i)})}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)}, t^{(j)})}}\right].$$

Rewriting $q(\boldsymbol{x}_0, \boldsymbol{x}_t) = q_t(\boldsymbol{x}_t)q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)$ and moving the conditional expectation $q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)$ into the inner part, we have

$$\min_{\phi} \mathbb{E}_{p(t^{(1:K)})}\mathbb{E}_{q_t(\boldsymbol{x}_t^{(1:K)})}\left[-\sum_{i=1}^{K} \mathbb{E}_{q_{0t}(i)}(\boldsymbol{x}_0^{(i)}|\boldsymbol{x}_t^{(i)})\left[e^{-\beta\mathcal{E}(\boldsymbol{x}_0^{(i)})}\right] \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, t^{(i)})}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)}, t^{(j)})}}\right].$$

By Eq. (9), we have $\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[e^{-\beta\mathcal{E}(\boldsymbol{x}_0)}\right] = e^{-\mathcal{E}_t(\boldsymbol{x}_t)}$, thus the above objective is equivalent to

$$\min_{\phi} \mathbb{E}_{p(t^{(1:K)})}\mathbb{E}_{q_t(\boldsymbol{x}_t^{(1:K)})}\left[-\sum_{i=1}^{K} e^{-\mathcal{E}_{t^{(i)}}(\boldsymbol{x}_t^{(i)})} \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, t^{(i)})}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)}, t^{(j)})}}\right]. \tag{36}$$

For each $t^{(1:K)}$ and $\boldsymbol{x}_t^{(1:K)}$, for $i = 1, \ldots, K$, define

$$a_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) := \frac{e^{-\mathcal{E}_{t^{(i)}}(\boldsymbol{x}_t^{(i)})}}{\sum_{j=1}^K e^{-\mathcal{E}_{t^{(j)}}(\boldsymbol{x}_t^{(j)})}},$$

$$b_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) := \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)}, t^{(i)})}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)}, t^{(j)})}},$$

$$c(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) := \sum_{j=1}^K e^{-\mathcal{E}_{t^{(j)}}(\boldsymbol{x}_t^{(j)})} > 0,$$

Then Eq. (36) is equivalent to

$$\min_\phi \mathbb{E}_{p(t^{(1:K)})} \mathbb{E}_{q_t(\boldsymbol{x}_t^{(1:K)})} \left[-c(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \sum_{i=1}^K a_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \log \left(b_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \right) \right].$$

For each fixed $t^{(1:K)}$ and $\boldsymbol{x}_t^{(!:K)}$, as $\sum_{i=1}^K a_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) = \sum_{i=1}^K b_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) = 1$, according to Gibbs' inequality, we have

$$-\sum_{i=1}^K a_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \log \left(b_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \right) \geq -\sum_{i=1}^K a_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \log \left(a_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \right),$$

so we have

$$\mathbb{E}_{p(t^{(1:K)})} \mathbb{E}_{q_t(\boldsymbol{x}_t^{(1:K)})} \left[-c(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \sum_{i=1}^K a_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \log \left(b_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \right) \right]$$

$$\geq \mathbb{E}_{p(t^{(1:K)})} \mathbb{E}_{q_t(\boldsymbol{x}_t^{(1:K)})} \left[-c(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \sum_{i=1}^K a_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \log \left(a_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) \right) \right],$$

and the equality holds when for each $t^{(1:K)}$ and each $\boldsymbol{x}_t^{(1:K)}$ in the supported space of $q_t(\boldsymbol{x}_t^{(1:K)})$,

$$b_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}) = a_i(\boldsymbol{x}_t^{(1:K)}, t^{(1:K)}), \quad i = 1, \ldots, K,$$

so given unlimited data and model capacity, the optimal ϕ^* satisfies

$$\frac{e^{-f_{\phi^*}(\boldsymbol{x}_t^{(i)}, t^{(i)})}}{\sum_{j=1}^K e^{-f_{\phi^*}(\boldsymbol{x}_t^{(j)}, t^{(j)})}} = \frac{e^{-\mathcal{E}_{t^{(i)}}(\boldsymbol{x}_t^{(i)})}}{\sum_{j=1}^K e^{-\mathcal{E}_{t^{(j)}}(\boldsymbol{x}_t^{(j)})}},$$

so for each $i = 1, \ldots, K$,

$$\frac{e^{-f_{\phi^*}(\boldsymbol{x}_t^{(i)}, t^{(i)})}}{e^{-\mathcal{E}_t(\boldsymbol{x}_t^{(i)})}} = \frac{\sum_{j=1}^K e^{-f_{\phi^*}(\boldsymbol{x}_t^{(j)}, t^{(j)})}}{\sum_{j=1}^K e^{-\mathcal{E}_{t^{(j)}}(\boldsymbol{x}_t^{(j)})}},$$

due to the arbitrariness of $\boldsymbol{x}_t^{(1:K)}$ and $t^{(1:K)}$, for any $\boldsymbol{x}_t^{(i)}, \boldsymbol{x}_t^{(j)}$, we have

$$\frac{e^{-f_{\phi^*}(\boldsymbol{x}_t^{(i)}, t^{(i)})}}{e^{-\mathcal{E}_{t^{(i)}}(\boldsymbol{x}_t^{(i)})}} = \frac{e^{-f_{\phi^*}(\boldsymbol{x}_t^{(j)}, t^{(j)})}}{e^{-\mathcal{E}_{t^{(j)}}(\boldsymbol{x}_t^{(j)})}}.$$

Therefore, there exists a constant C_t independent of $\boldsymbol{x}_t$, such that

$$e^{-f_{\phi^*}(\boldsymbol{x}_t, t)} = C_t \cdot e^{-\mathcal{E}_t(\boldsymbol{x}_t)} \propto e^{-\mathcal{E}_t(\boldsymbol{x}_t)},$$

and then $\nabla_{\boldsymbol{x}_t} f_{\phi^*}(\boldsymbol{x}_t, t) = \nabla_{\boldsymbol{x}_t} \mathcal{E}_t(\boldsymbol{x}_t)$. $\qquad \square$

Table 4. Comparison between energy-guided sampling algorithms.

Method	Optimal Solution of Energy	Optimal Solution of Guidance	Exact Guidance
CEP (ours)	$-\log \mathbb{E}_{q_{0t}(\boldsymbol{x}_0\|\boldsymbol{x}_t)}\left[e^{-\mathcal{E}_0(\boldsymbol{x}_0)}\right]$	$\mathbb{E}_{q_{0t}(\boldsymbol{x}_0\|\boldsymbol{x}_t)}\left[-e^{\mathcal{E}_t(\boldsymbol{x}_t)-\mathcal{E}_0(\boldsymbol{x}_0)}\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0\|\boldsymbol{x}_t)\right]$	✓
MSE	$\mathbb{E}_{q_{0t}(\boldsymbol{x}_0\|\boldsymbol{x}_t)}[\mathcal{E}_0(\boldsymbol{x}_0)]$	$\mathbb{E}_{q_{0t}(\boldsymbol{x}_0\|\boldsymbol{x}_t)}\left[\mathcal{E}_0(\boldsymbol{x}_0)\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0\|\boldsymbol{x}_t)\right]$	✗
DPS	$\mathcal{E}_0\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0\|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right)$	$\mathbb{E}_{q_{0t}(\boldsymbol{x}_0\|\boldsymbol{x}_t)}\left[\left(\left(\nabla\mathcal{E}_0\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0\|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right)\right)^{\top}\boldsymbol{x}_0\right)\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0\|\boldsymbol{x}_t)\right]$	✗

E. Comparison with Existing Energy-Guided Sampling Algorithms

Firstly, we can easily compute the gradients for the energy guidance used in MSE (Janner et al., 2022; Bao et al., 2022b) and DPS (Ho et al., 2022c; Chung et al., 2022), and we summarize the results in Table 4. Below we propose a deeper connection between these methods.

Comparison for Energy. The exact energy function is

$$\mathcal{E}_t(\boldsymbol{x}_t) = -\log \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[e^{-\mathcal{E}_0(\boldsymbol{x}_0)}\right].$$

By exchanging the order of the $\log$ function and the expectation, we can derive the energy used in MSE:

$$\mathcal{E}_t^{\mathrm{MSE}}(\boldsymbol{x}_t) = -\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\log\left(e^{-\mathcal{E}_0(\boldsymbol{x}_0)}\right)\right] = \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\mathcal{E}_0(\boldsymbol{x}_0)\right].$$

By further exchanging the order of $\mathcal{E}_0$ function and the expectation, we can derive the energy used in DPS:

$$\mathcal{E}_t^{\mathrm{DPS}}(\boldsymbol{x}_t) = \mathcal{E}_0\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\boldsymbol{x}_0\right]\right).$$

Intuitively, exchanging the order between a nonlinear function and an expectation will introduce additional approximation errors, and the errors depend on the complexity of the nonlinear function. As $\log(\cdot)$ is a simple concave function but $\mathcal{E}_0(\cdot)$ may be rather complex, the approximation error of DPS may be quite large, which may explain why the sample results of DPS are quite worse than CEP in Fig. 2 and Fig. 6.

Comparison for Guidance. The exact guidance is

$$\nabla_{\boldsymbol{x}_t}\mathcal{E}_t(\boldsymbol{x}_t) = \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[-e^{\mathcal{E}_t(\boldsymbol{x}_t)-\mathcal{E}_0(\boldsymbol{x}_0)}\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)\right]. \tag{37}$$

And the guidance by MSE is

$$\nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{MSE}}(\boldsymbol{x}_t) = \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\mathcal{E}_0(\boldsymbol{x}_0)\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)\right].$$

And the guidance by DPS is

$$\nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{DPS}}(\boldsymbol{x}_t) = \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\left(\left(\nabla\mathcal{E}_0\left(\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right)\right)\right)^{\top}\boldsymbol{x}_0\right)\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)\right].$$

Below we discuss the relationship between these three functions.

By taking Taylor expansion for $e^{\mathcal{E}_t(\boldsymbol{x}_t)-\mathcal{E}_0(\boldsymbol{x}_0)} \approx 1 + \mathcal{E}_t(\boldsymbol{x}_t) - \mathcal{E}_0(\boldsymbol{x}_0)$, we have

$$\begin{aligned}
\nabla_{\boldsymbol{x}_t}\mathcal{E}_t(\boldsymbol{x}_t) &\approx \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[(-1-\mathcal{E}_t(\boldsymbol{x}_t)+\mathcal{E}_0(\boldsymbol{x}_0))\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)\right] \\
&= \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\mathcal{E}_0(\boldsymbol{x}_0)\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)\right] \\
&= \nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{MSE}}(\boldsymbol{x}_t),
\end{aligned}$$

where the penultimate equation follows the fact that $\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)] = 0$. Therefore, the guidance by $\nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{MSE}}(\boldsymbol{x}_t)$ is a first-order approximation of the true energy guidance by assuming $\mathcal{E}_t(\boldsymbol{x}_t) \approx \mathcal{E}_0(\boldsymbol{x}_0)$, which only makes sense for t near to 0. However, as shown in Fig. 1, for t near to T, $\mathcal{E}_t$ is quite different from $\mathcal{E}_0$, so guided sampling by

$\nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{MSE}}(\boldsymbol{x}_t)$ may have large guidance errors near to T, which can explain why MSE is worse than CEP, especially for large β.

By further taking Taylor expansion for $\mathcal{E}_0(\boldsymbol{x}_0)$ at $\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]$, we have

$$\mathcal{E}_0(\boldsymbol{x}_0) \approx \mathcal{E}_0\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right) + \left(\nabla\mathcal{E}_0\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right)\right)^{\top}\left(\boldsymbol{x}_0 - \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right)$$

Then we can further approximate $\nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{MSE}}(\boldsymbol{x}_t)$ by

$$
\begin{aligned}
\nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{MSE}}(\boldsymbol{x}_t) &\approx \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\Big[\left(\nabla\mathcal{E}_0\left(\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right)\right)^{\top}\boldsymbol{x}_0\right)\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)\Big]\\
&\quad + \left(\left(\mathcal{E}_0\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right) - \nabla\mathcal{E}_0\left(\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right)\right)^{\top}\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right)\right)\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)]\\
&= \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\Big[\left(\nabla\mathcal{E}_0\left(\left(\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]\right)\right)^{\top}\boldsymbol{x}_0\right)\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)\Big]\\
&= \nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{DPS}}(\boldsymbol{x}_t),
\end{aligned}
$$

where the penultimate equation follows the fact that $\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\nabla_{\boldsymbol{x}_t}\log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)] = 0$. Therefore, the guidance by $\nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{DPS}}(\boldsymbol{x}_t)$ is a further first-order approximation of $\nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{MSE}}(\boldsymbol{x}_t)$ by assuming $\boldsymbol{x}_0 \approx \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0]$, which also only makes sense for t near to 0. However, as $\nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{MSE}}(\boldsymbol{x}_t)$ is also an approximation for the exact guidance $\nabla_{\boldsymbol{x}_t}\mathcal{E}_t(\boldsymbol{x}_t)$, the difference between $\nabla_{\boldsymbol{x}_t}\mathcal{E}_t^{\mathrm{DPS}}(\boldsymbol{x}_t)$ and $\nabla_{\boldsymbol{x}_t}\mathcal{E}_t(\boldsymbol{x}_t)$ may be rather large.

Additional Experiment Results. We further compare CEP, MSE, and DPS in both 2-D experiments and offline RL experiments. All the results show that empirically, the performance of CEP is significantly better than MSE, and MSE is significantly better than DPS. We refer to Appendix L and Appendix I.3 for details.

F. Relationship with Contrastive Learning

Given a condition variable c, assume $(\boldsymbol{x}_0, c) \sim q_0(\boldsymbol{x}_0, c)$, and we learn the intermediate energy guidance by a neural network $f_\phi(\cdot, c, t) : \mathbb{R}^d \to \mathbb{R}$ parameterized by ϕ. In this section, we prove that for a special energy function ($\beta = 1$ and $\mathcal{E}(\boldsymbol{x}_0) = -\log q_0(c|\boldsymbol{x}_0)$), the objective of CEP in Eq. (12) is equivalent to the traditional contrastive InfoNCE objective (for a fixed t).

Theorem F.1. *If $\beta = 1$ and $\mathcal{E}(\boldsymbol{x}_0) = -\log q_0(c|\boldsymbol{x}_0)$, the objective in Eq. (12) with the sum over all possible c is equivalent to*

$$\mathbb{E}_{t,\boldsymbol{\epsilon}^{(1:K)}}\mathbb{E}_{\prod_{i=1}^K q_0(\boldsymbol{x}_0^{(i)},c^{(i)})}\left[-\sum_{i=1}^K \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},c^{(i)},t)}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)},c^{(i)},t)}}\right], \tag{38}$$

Proof. Firstly, for a fixed c, Eq. (12) becomes

$$\min_\phi \mathbb{E}_{p(t)}\mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})}\mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})}\left[-\sum_{i=1}^K q_0(c|\boldsymbol{x}_0^{(i)})\log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},c,t)}}{\sum_{j=1}^K e^{-f_\phi(\boldsymbol{x}_t^{(j)},c,t)}}\right].$$

By taking the integral over c, it becomes

$$\min_{\phi} \mathbb{E}_{p(t)} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[-\sum_{c} \sum_{i=1}^{K} q_0(c|\boldsymbol{x}_0^{(i)}) \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},c,t)}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)},c,t)}} \right]$$

$$\Leftrightarrow \min_{\phi} \mathbb{E}_{p(t)} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[-\sum_{i=1}^{K} \sum_{c} q_0(c|\boldsymbol{x}_0^{(i)}) \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},c,t)}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)},c,t)}} \right]$$

$$\Leftrightarrow \min_{\phi} \mathbb{E}_{p(t)} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[-\sum_{i=1}^{K} \mathbb{E}_{q(c|\boldsymbol{x}_0^{(i)})} \left[\log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},c,t)}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)},c,t)}} \right] \right]$$

$$\Leftrightarrow \min_{\phi} \mathbb{E}_{p(t)} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[-\sum_{i=1}^{K} \mathbb{E}_{q(c^{(i)}|\boldsymbol{x}_0^{(i)})} \left[\log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},c^{(i)},t)}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)},c^{(i)},t)}} \right] \right]$$

$$\Leftrightarrow \min_{\phi} \mathbb{E}_{p(t)} \mathbb{E}_{q_0(\boldsymbol{x}_0^{(1:K)},c^{(1:K)})} \mathbb{E}_{p(\boldsymbol{\epsilon}^{(1:K)})} \left[-\sum_{i=1}^{K} \log \frac{e^{-f_\phi(\boldsymbol{x}_t^{(i)},c^{(i)},t)}}{\sum_{j=1}^{K} e^{-f_\phi(\boldsymbol{x}_t^{(j)},c^{(i)},t)}} \right]$$

$\square$

Note that the above objective assumes that we can draw samples from $q_0(\boldsymbol{x}_0, c)$, which means that we can draw samples from $p_0(\boldsymbol{x}_0) = q_0(\boldsymbol{x}_0|c)$. However, for general energy functions, such an assumption is hard to ensure and we can not draw samples from $p_0(\boldsymbol{x}_0)$ but only $q_0(\boldsymbol{x}_0)$. Therefore, CEP can be understood as a generalized version of the traditional contrastive objective and is suitable for both conditional sampling and energy-guided sampling in diffusion models.

G. Relationship between Inverse Temperature and Guidance Scale

A widely-used trick in guided sampling is to introduce an additional hyperparameter s called "guidance scale" (Dhariwal & Nichol, 2021), and replace the score function for p_t by $\tilde{p}_t$ during the guided sampling procedure as following:

$$\nabla_{\boldsymbol{x}_t} \log \tilde{p}_t(\boldsymbol{x}_t) := \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t) - s \cdot \nabla_{\boldsymbol{x}_t} \mathcal{E}_t(\boldsymbol{x}_t) \tag{39}$$

According to Eq. (37), the above equation is equivalent to

$$\nabla_{\boldsymbol{x}_t} \log \tilde{p}_t(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t) - s \cdot \mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[-e^{\mathcal{E}_t(\boldsymbol{x}_t) - \mathcal{E}_0(\boldsymbol{x}_0)} \nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t) \right]$$

$$= \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t) + s \cdot \frac{\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[e^{-\beta\mathcal{E}(\boldsymbol{x}_0)} \nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t) \right]}{\mathbb{E}_{q_{0t}(\boldsymbol{x}_0|\boldsymbol{x}_t))} \left[e^{-\beta\mathcal{E}(\boldsymbol{x}_0)} \right]}.$$

Note that the influences of changing s and changing β are different: changing s will linearly affect the guidance strength, but changing β will affect the guidance w.r.t. the exponential term. Thus, s and β are two different hyperparameters and we can tune them together.

Empirically, we find that in simple 2-D experiments, only changing β is enough to guarantee convergence to our desired distribution $p(\boldsymbol{x})$. However, in complex tasks such as image synthesis and reinforcement learning, by only varying β we cannot guarantee a good performance, so a larger s is somewhat necessary. Our hypothesis for explaining this is that the neural network used is not expressive enough, such that when β increases and the task becomes more complex, the model capacity approaches saturation so we must rely on a training-free method in order to amplify the guidance effect (Appendix I.2).

H. E-MSE Energy Guidance

In this section, we propose an alternative way of CEP to learn energy guidance. In order to ensure an exact converged point, we add an exponential activation in the original MSE-based training objective (Eq. (14)), which we name E-MSE:

$$\min_{\phi} \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_t} \left[\| \exp(f_\phi(\boldsymbol{x}_t,t)) - \exp(\beta\mathcal{E}(\boldsymbol{x}_0)) \|_2^2 \right] \tag{40}$$

Such a training objective could also guarantee convergence to the exact energy guidance for sampling from $p(\boldsymbol{x})$. As visualized in Figure 6, we find that both CEP and E-MSE guidance could generate more accurate data samples in 2-D settings compared to the MSE-based guidance method, especially when β is large. However, one main disadvantage of E-MSE guidance is that Eq. (40) is not numerically stable due to the isolated exponential term. In particular, in RL settings where the energy function $\mathcal{E}$ is no longer normalized in the range $[0, 1]$, but is defined by a potentially noisy neural network Q_ψ, E-MSE guidance generally tends to underperform CEP and even MSE guidance (Table 6).

I. Experiment details for offline RL

I.1. Pseudocode of QGPO

Algorithm 1 Q-Guided Policy Optimization (QGPO) for Offline RL

Initialize the diffusion behavior model ϵ_θ, the action evaluation model Q_ψ and the intermediate energy model f_ϕ
// Training the behavior model
for each gradient step **do**
    Sample B data points (s, a) from $\mathcal{D}^\mu$, B Gaussian noises ϵ from $\mathcal{N}(0, I)$ and B time t from $\mathcal{U}(0, T)$
    Perturb a according to $a_t := \alpha_t a + \sigma_t \epsilon$
    Update $\theta \leftarrow \theta - \lambda_\theta \nabla_\theta \sum[\|\epsilon_\theta(a_t|s, t) - \epsilon\|_2^2]$
end for
// Generating the support action set
for each state s in $\mathcal{D}^\mu$ **do**
    Sample K support actions $\hat{a}^{(1:K)}$ from $\mu_\theta(\cdot|s)$ and store them as $\mathcal{D}^{\mu_\theta}(s)$
end for
// Training the action evaluation model and the energy guidance model
for each gradient step **do**
    Sample B data points (s, a, r, s') from $\mathcal{D}^\mu$, B Gaussian noises ϵ from $\mathcal{N}(0, I)$ and B time t from $\mathcal{U}(0, T)$
    Retrieve support action sets $\hat{a}^{(1:K)}$ and $\hat{a}'^{(1:K)}$ respectively from $\mathcal{D}^\mu(s)$ and $\mathcal{D}^\mu(s')$
    Calculate the target Q-value $\mathcal{T}^\pi Q_\psi(s, a) = r(s, a) + \gamma \sum_{\hat{a}'} \left[\frac{e^{\beta_Q Q_\psi(s', \hat{a}')}}{\sum_{\hat{a}'} e^{\beta_Q Q_\psi(s', \hat{a}')}} Q_\psi(s', \hat{a}') \right]$ and detach gradient
    Update $\psi \leftarrow \psi - \lambda_\psi \nabla_\psi \sum[\|Q_\psi(s, a) - \mathcal{T}^\pi Q_\psi(s, a)\|_2^2]$
    Perturb $\hat{a}$ according to $\hat{a}_t := \alpha_t \hat{a} + \sigma_t \epsilon$
    Update $\phi \leftarrow \phi + \lambda_\phi \nabla_\phi \sum_i [\frac{e^{\beta Q_\psi(s, \hat{a}_i)}}{\sum_j e^{\beta Q_\psi(s, \hat{a}_j)}} \log \frac{e^{f_\phi(\hat{a}_{i,t}|s, t)}}{\sum_j e^{f_\phi(\hat{a}_{j,t}|s, t)}}]$
end for

I.2. Experiment Details of QGPO

For offline RL benchmarks, our methods require training three neural networks in total for each task, namely a diffusion-based behavior model s_θ, an action evaluation model Q_ψ, and an energy guidance model f_ϕ. We first provide experiment details in training every component described above and then discuss how to combine these components for policy evaluation.

Training behavior model. The architecture and training method of our behavior model completely follow Chen et al. (2022). The network architecture resembles U-Nets, but with spatial convolutions changed to dense connections, such that it is compatible with a vectorized data representation. A similar network architecture was also adopted by Janner et al. (2022) and Pearce et al. (2022). We train the behavior model for 600k gradient steps, using the Adam optimizer with a learning rate of 1e-4. The batchsize is 4096. As for the data perturbation method, we adopt the default VPSDE setting in (Song et al., 2021b) with a linear schedule. The α_t and σ_t in Eq. (8) are:

$$\alpha_t = -\frac{\beta_1 - \beta_0}{4} t^2 - \frac{\beta_0}{2} t, \quad \sigma_t = \sqrt{1 - \alpha_t^2}, \quad \beta_0 = 0.1, \ \beta_1 = 20. \tag{41}$$

Training action evaluation model. The action evaluation model is a 3-layer MLP with 256 hidden units and ReLU activations. We train the action evaluation model for 500k gradient steps, using the Adam optimizer with a learning rate of 3e-4. The batchsize is 256. We set $\beta_Q = 1$ and $K = 16$ for MuJoCo Locomotion tasks, $\beta_Q = 20$ and $K = 32$ for AntMaze tasks in Eq. (22). Before training, we follow Kostrikov et al. (2022) and normalize task rewards. We also use standard tricks such as soft updates (Lillicrap et al., 2016) and double networks (Fujimoto et al., 2018) to stabilize Q-learning.

Training energy guidance model. The energy guidance model is a 4-layer MLP with 256 hidden units and SiLU activations. We train it for 1M gradient steps, using the Adam optimizer with a learning rate of 3e-4. The batchsize is 256. The size support action set is the same as the one used in training the action evaluation model ($K = 16$ for MuJoCo Locomotion and $K = 32$ for AntMaze), though we set $\beta = 3$ in all tasks.

Evaluation. We run all experiments over 5 independent trials and report their averaged performance. For each trial, we additionally collect the evaluation score averaged on multiple test seeds (10 for MuJoCo Locomotion and 100 for AntMaze).

In order to sample from the learned diffusion-based policy, we adopt a recent advance in diffusion sampling, namely DPM-Solver (Lu et al., 2022b). We use the second-order sampler and report performance scores at a diffusion step of 15. We conduct an ablation study of diffusion steps (Figure 5) and find that a diffusion step of 10 could already yield equally good performance, while a diffusion step of 5 only slightly underperforms 15 diffusion steps. Note that we only ablated diffusion step numbers in evaluation. The support action set is still generated using a diffusion step of 15. We also adopted a widely used trick in guided diffusion sampling (Dhariwal & Nichol, 2021; Ho & Salimans, 2021), which tunes a hyperparameter s to amplify the guidance effect during sampling, by multiplying energy guidance in Eq. (18) with the guidance scale s. To choose the optimal energy guidance scales s for action sampling, we sweep over $[1.0, 2.0, 3.0, 5.0, 8.0, 10.0]$ for MuJoCo Locomotion and $[1.0, 1.5, 2.0, 2.5, 3.0, 4.0]$ for AntMaze tasks during evaluation (Figure 4). Gradient scales used for reported scores are listed in Table 5.
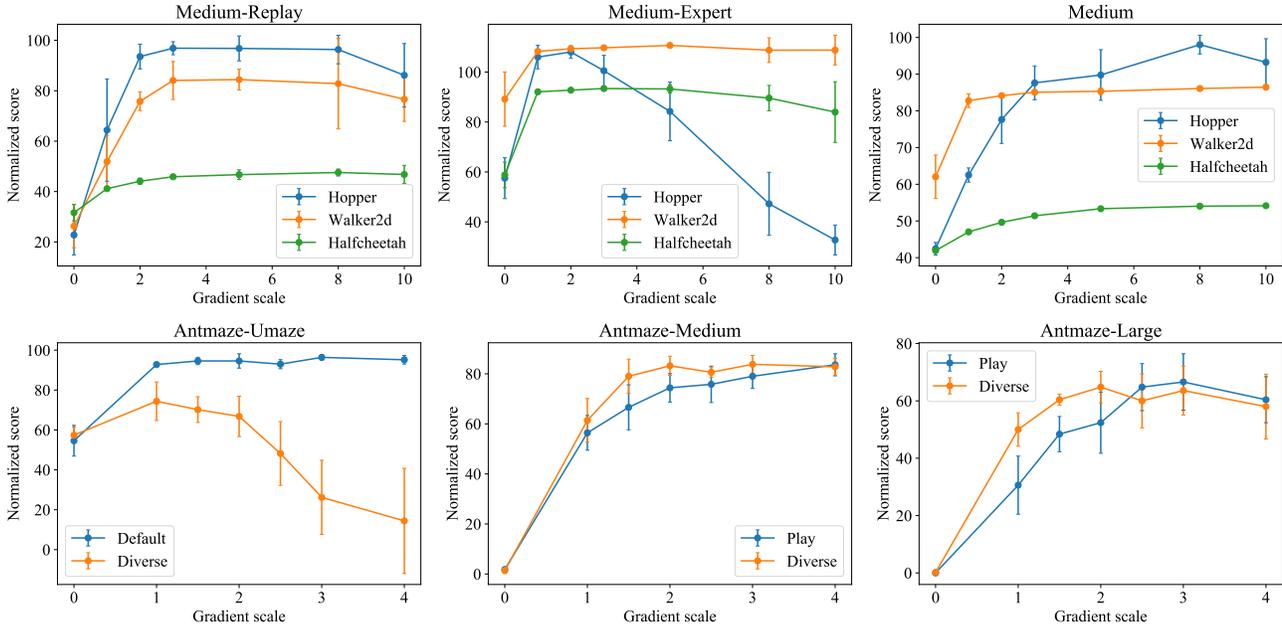
Figure 4. Ablation of gradient scales in D4RL benchmark.
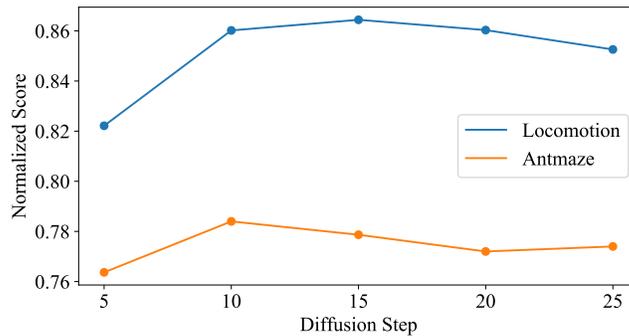
Figure 5. Ablation of diffusion steps in evaluation.

I.3. Experiment Details for Other Baselines

Diffusion-QL. We use the official implementation of Diffusion-QL (`https://github.com/Zhendong-Wang/Diffusion-Policies-for-Offline-RL`) and default hyperparameter settings to rerun all experiments for Diffusion-QL to ensure a consistent evaluation metric. We follow Fu et al. (2020) and report averaged scores across

Locomotion-Medium-Expert	Walker2d 5.0	Halfcheetah 3.0	Hopper 2.0
Locomotion-Medium	Walker2d 10.0	Halfcheetah 10.0	Hopper 8.0
Locomotion-Medium-Replay	Walker2d 5.0	Halfcheetah 8.0	Hopper 3.0
AntMaze-Fixed	Umaze 3.0	Medium 4.0	Large 3.0
AntMaze-Diverse	Umaze 1.0	Medium 3.0	Large 2.0

Table 5. Guidance scale s used across different tasks

Environment	MSE	E-MSE	RS	CEP
Locomotion	68.0	58.1	76.9	86.6
AntMaze	46.4	24.5	63.0	78.3

Table 6. Ablation studies of different energy guidance methods and the resampling technique.

5 independent trails at the end of training for each task, instead of the max performance scores during training as in Wang et al. (2022b). In addition, we notice that Diffusion-QL adopts a resampling technique for evaluation. Specifically, during evaluation, the learned policy first generates 50 different action candidates and then selects one action with the highest Q-value for execution. We empirically found that such a technique is important for good performance in MuJoCo Locomotion tasks. However, this technique makes it hard to reflect the true quality of sampled actions before resampling and is computationally expensive, we thus additionally conduct an ablation study in which we remove the resampling procedure in evaluation, and use a single action candidate (Diffusion-QL@1 in Table 2 and Appendix M).

Ablations of CEP guidance. We study three variations of our proposed guidance method. Specifically, an MSE-based guidance method as described in Eq. (14) (similarly used in Janner et al. (2022)), an E-MSE guidance method as described in Eq. (40), and a resampling-based method following Chen et al. (2022). For MSE and E-MSE guidance, we only change the training objective of the energy guidance model while leaving other hyperparameters untouched. For the resampling-based method, the energy guidance model is not required. During evaluation, at every decision step we first sample 50 random action candidates from the behavior policy model $\mu_\theta(\boldsymbol{a}|\boldsymbol{s})$ and then select one action with the highest predicted Q-value via Q_ψ for execution.

J. Experiment Details for 2-D experiments

To perform unconditional energy-guided sampling in low-dimensional data space. Our method requires training two neural networks independently, specifically, one generative diffusion model and one energy guidance model. In contrast with offline RL, the energy function at $t = 0$ data space is pre-defined (as illustrated in Figure 6) and does not require training. A total of 1M datapoints is generated and used as the training set. Each datapoint contains a two-dimensional data sample $\boldsymbol{x}$ and a float number e representing its energy.

Training diffusion generative models. The generative diffusion model is a 5-layer MLP with hidden sizes of $[512, 512, 512, 512, 256]$ and SiLU activations. The network is trained for 750 epochs, using the Adam optimizer with a learning rate of 1e-4. The batchsize is 1 with $K = 4096$. As for the data perturbation method, we adopt the default VPSDE setting in (Song et al., 2021b) with a linear schedule. The α_t and σ_t in Eq. (8) are:

$$\alpha_t = -\frac{\beta_1 - \beta_0}{4}t^2 - \frac{\beta_0}{2}t, \quad \sigma_t = \sqrt{1 - \alpha_t^2}, \quad \beta_0 = 0.1, \ \beta_1 = 20. \tag{42}$$

Training energy guidance models. The energy guidance model is a 4-layer MLP with 512 hidden units and SiLU activations. We train it for 750 epochs, using the Adam optimizer with a learning rate of 3e-4. The batchsize is also 4096.

Guided sampling. In order to perform guided sampling, we adopt a recent advance in diffusion sampling, namely DPM-Solver (Lu et al., 2022b). We use the second-order sampler and a diffusion step of 25. We fix the guidance scale s to 1 in all experiments.

K. Experiment Details for Image Synthesis

We completely follow Dhariwal & Nichol (2021) to train and evaluate our energy-guided diffusion models for image synthesis, without any kind of hyperparameter tuning or network architecture changes. For the generative diffusion prior, we use the pretrained ImageNet models released at `https://github.com/openai/guided-diffusion`. For the energy guidance model, we adopt the same U-Net architecture as Dhariwal & Nichol (2021) but rewrite the training objective to Eq. (15) for conditional image synthesis, and to Eq. (13) for energy guided image synthesis. Our ImageNet 128 energy guidance model is trained for 300k gradient steps with a batch size of 256 (distributed on 8 GPUs). The ImageNet 256 energy guidance model is trained for 500k steps. During sampling, we use 250 diffusion steps by default except when we use a DDIM (Song et al., 2021a) sampler with 25 steps.

For energy-guided image synthesis tasks, we set $\beta = 50$. A penalty is added to the energy function at defined $t = 0$ data space in Eq. (23) when the image's average saturation is lower than 0.1. This penalization mainly intends to avoid generating images with low saturation (overly bright), such that image samples guided by different color models are more distinguishable. We respectively let h_{tar} be 0 (red), $2\pi/3$ (green) and $4\pi/3$ (blue) for the three guidance models.

L. More 2-D Results
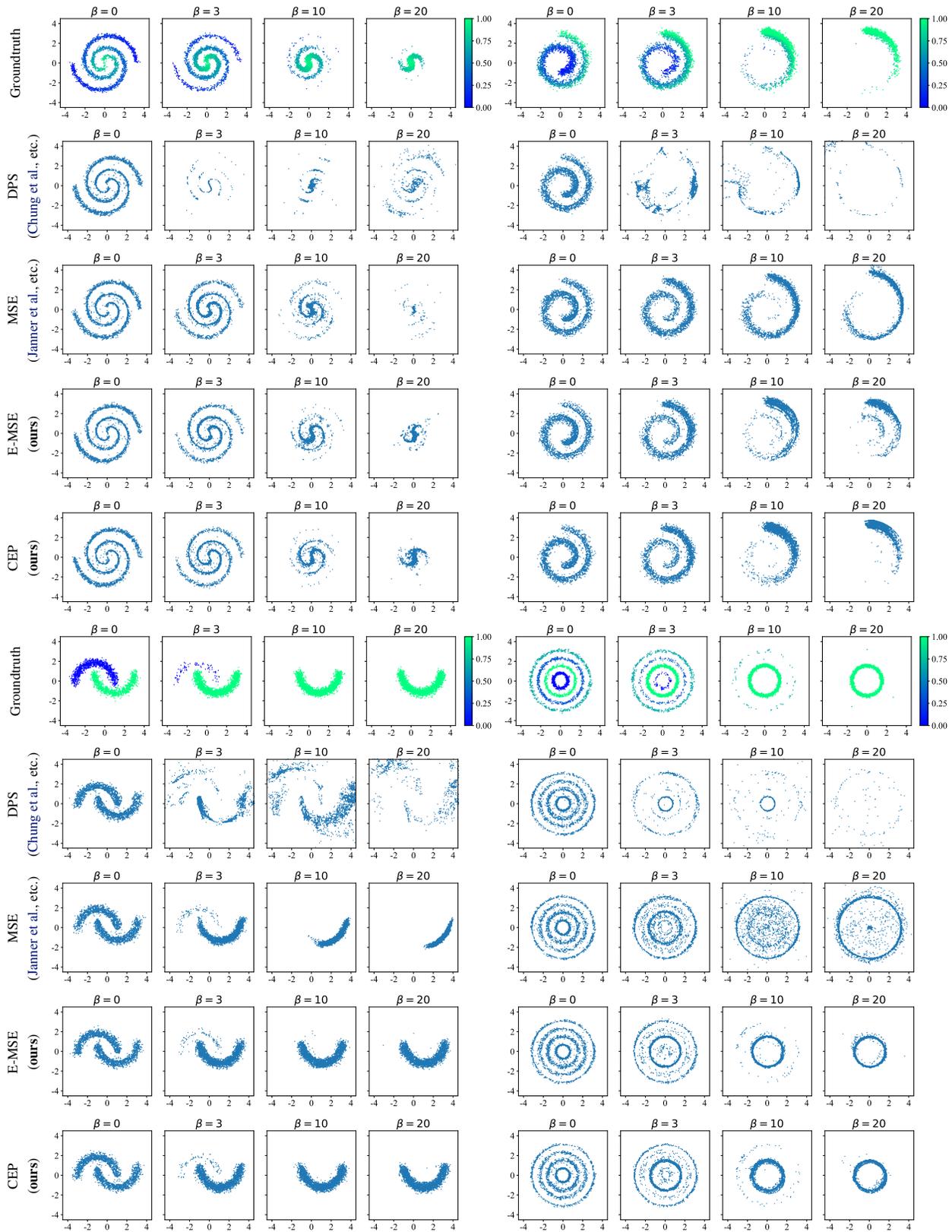
Figure 6. Scatter plots of different energy guidance methods in 2-D experiments. E-MSE is another method we propose as a variant of MSE guidance (Appendix H).

27
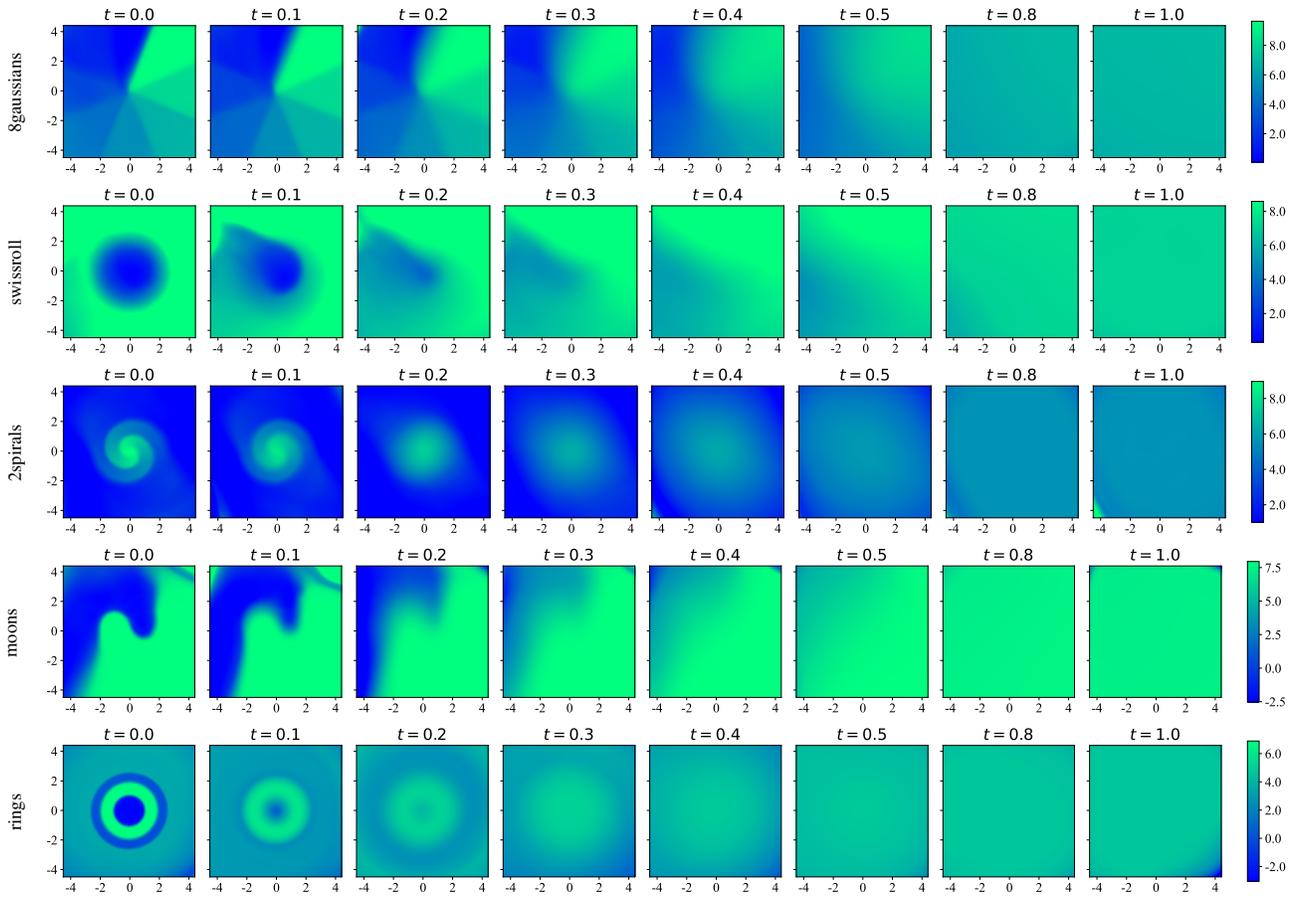
Figure 7. Visualization of the contrastively learned intermediate energy model when $\beta = 10$.

M. Training Curves for Offline Reinforcement Learning
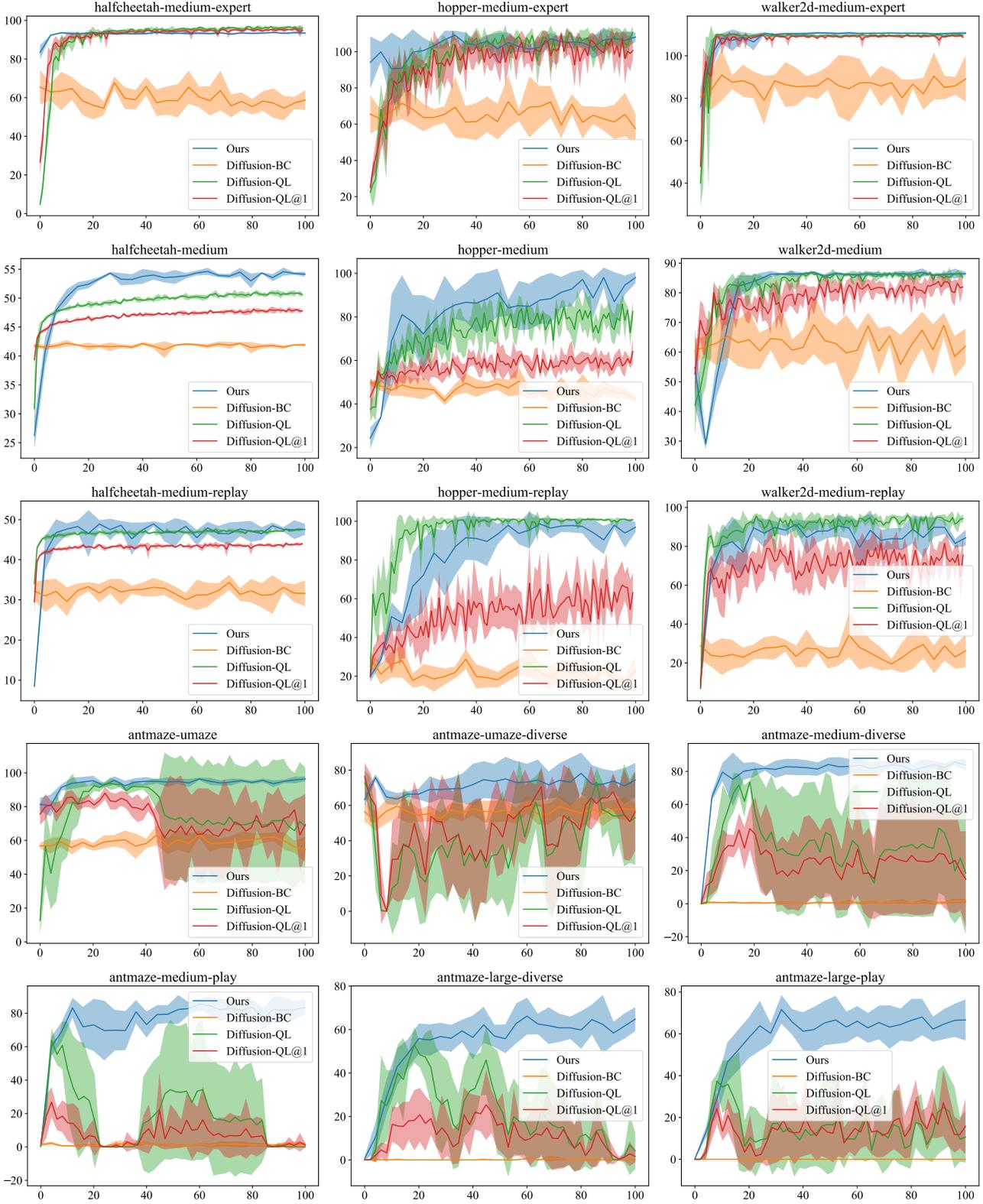
Figure 8. Training curves of QGPO (ours) and several baselines. We plot mean and standard deviation of results across five random seeds. Scores are normalized according to (Fu et al., 2020). Diffusion BC indicates evaluation scores of the learned behavior policy without any guidance ($s = 0$).

N. More Results for Energy-Guided Image Synthesis
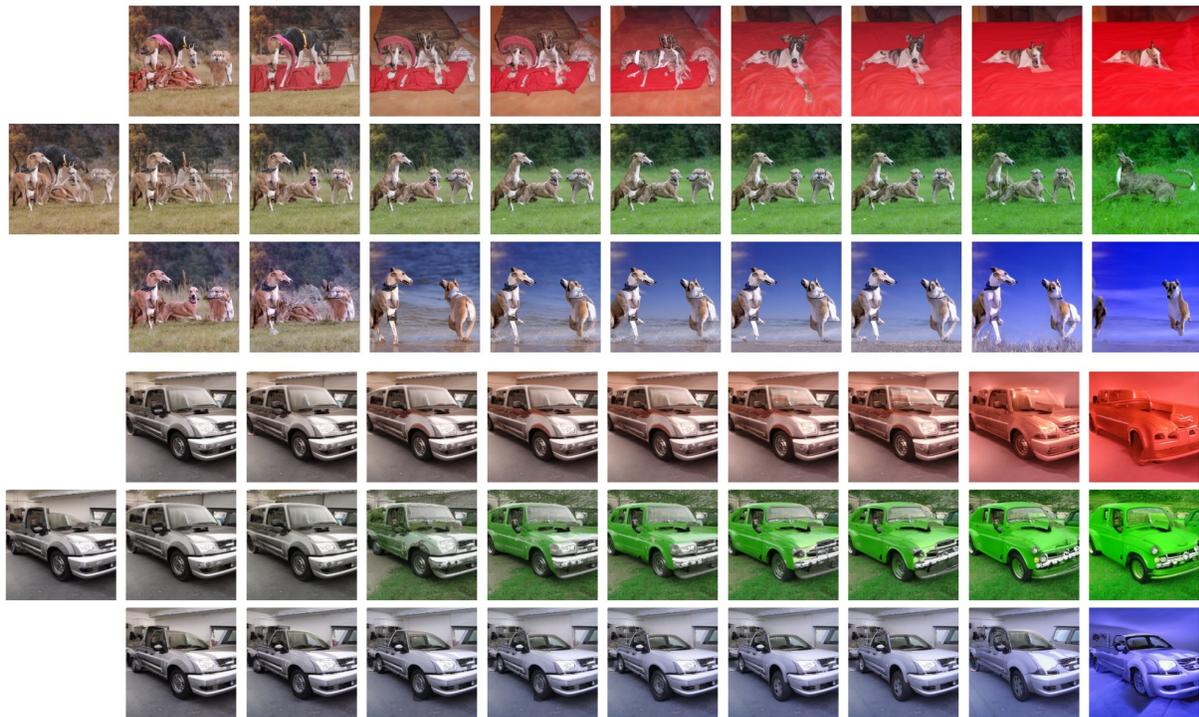
Figure 9. Ablation of color guidance with a conditional diffusion prior. From left to right are samples under an increasing guidance scale in [0.0, 0.25, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 5.0, 10.0].
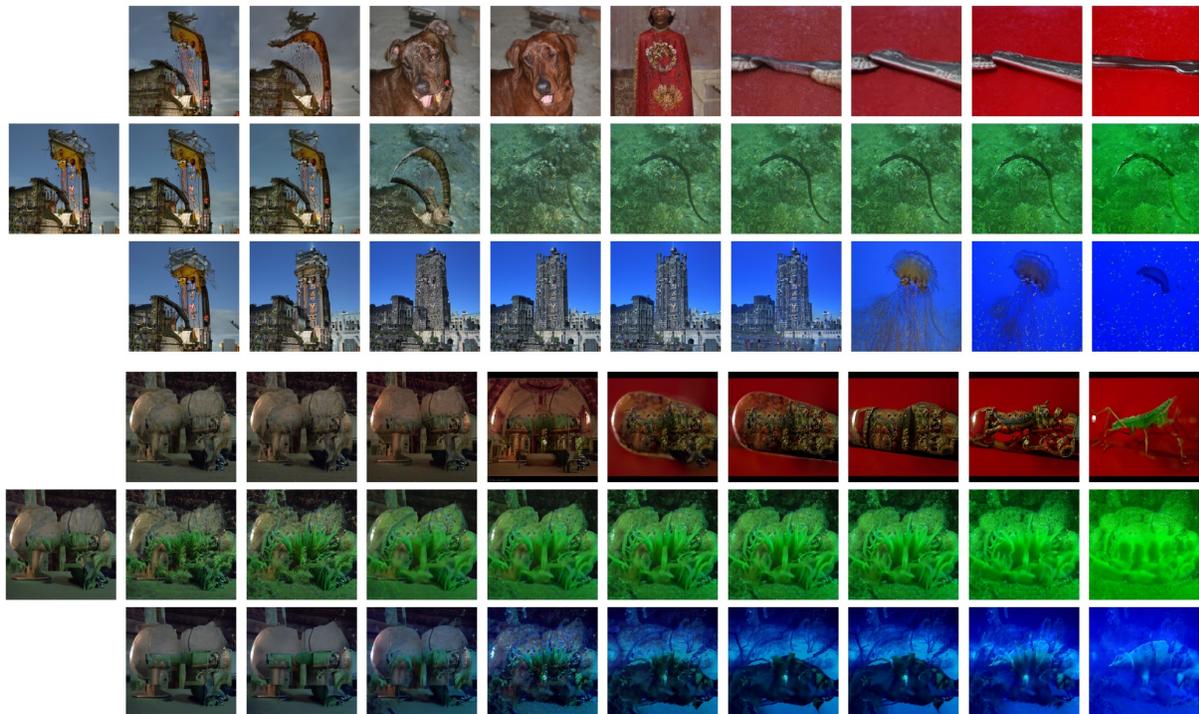

Figure 10. Ablation of color guidance with an unconditional diffusion prior. From left to right are samples under an increasing guidance scale in [0.0, 0.25, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 5.0, 10.0].

O. CEP Guidance vs. Classifier Guidance

<div align="center">
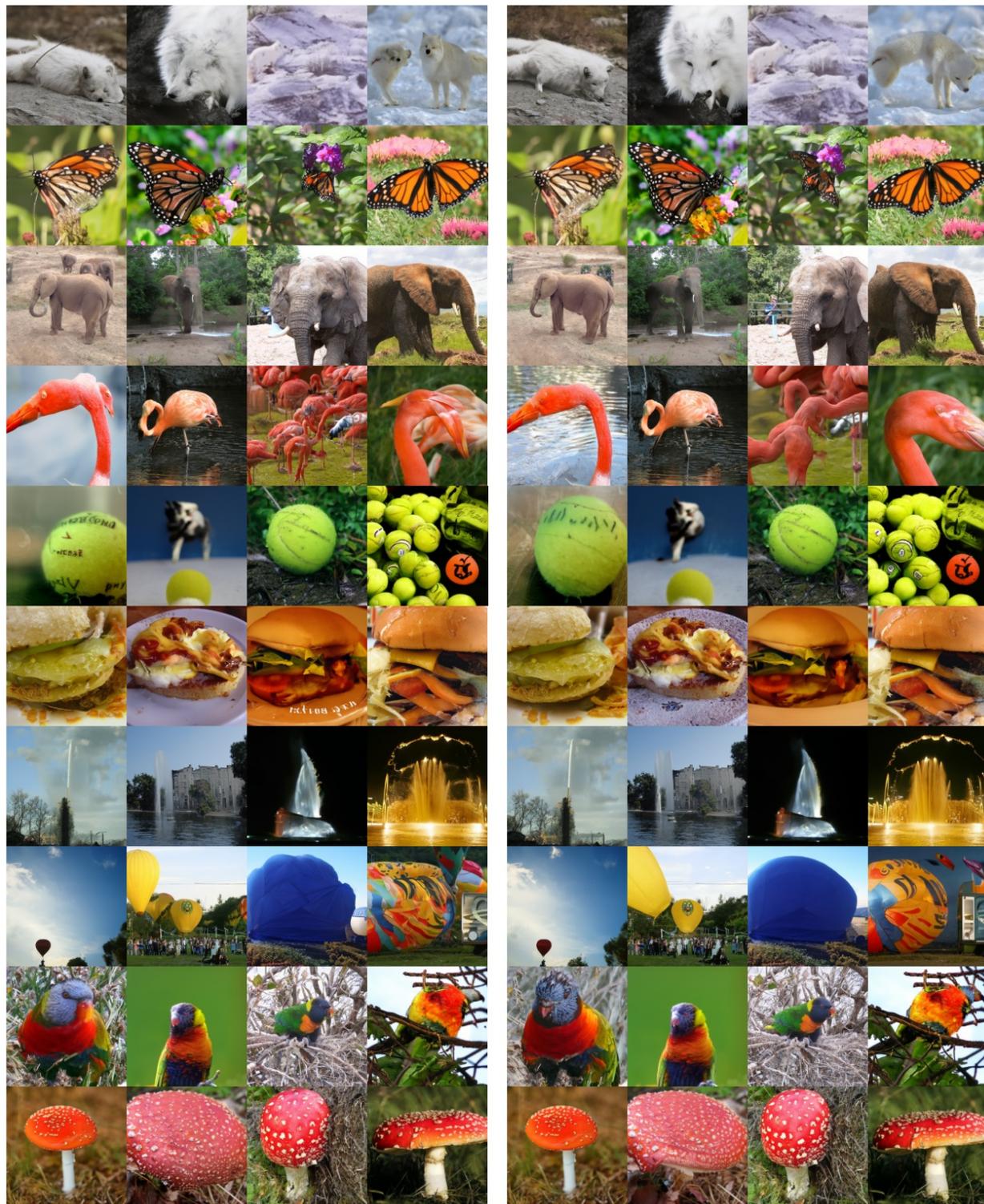
Classifier guidance (Dhariwal & Nichol, 2021)   Energy guidance (**ours**)

</div>

Figure 11. Samples from the conditional 256×256 ImageNet model with different guidance methods. Random seeds are fixed across experiments. Classes are 279: arctic fox, 323: monarch butterfly, 386: african elephant, 130: flamingo, 852: tennis ball, 933: cheeseburger, 562: fountain, 417: balloon, 90: lorikeet, 992: agaric.