DIFFUSION MODEL FINE-TUNING WITH CONSTRAINED OPTIMIZATION

Naicheng He, Nuo Wen Lei, Wanjia Fu, Yixiang Sun
Department of Computer Science
Brown University
Providence, RI 02912, USA
{naicheng_he,nuo_wen_lei,wanjia_fu,yixiang_sun}@brown.edu

ABSTRACT

Diffusion models have been successful in a number of fields in recent years due to their ability to obtain a synthetic probability distribution for a given dataset. However, they are likely trained on general data and can generate undesired results for specific tasks and use cases. In this work, we propose a simple pipeline to fine-tune unconditional diffusion models via a constrained optimization process. We show that this formulation has broad compatibilities with straight-forward score functions in diverse domains of applications by experiments in (i) removing the class of digits in an MNIST data set, (ii) simulating safety constraints for trajectory planning, and (iii) optimizing with pairs of expert preferences on polymer generation tasks. We show that our framework is robust and easy to implement, only requiring an additional coarse penalty term for all of the experiments we demonstrate.

1 INTRODUCTION

Diffusion models are a powerful tool to generate various forms of data, from visual data such as images(Zhang et al. (2024); Ruiz et al. (2023); Li et al. (2023)) and videos(Ho et al. (2022); Xing et al. (2024); Blattmann et al. (2023), to robot learning data such as video policies(Dong et al. (2024); Chen et al. (2024b)).

A diffusion model reflects approximately the underlying distribution of its training data. Such distribution could be unideal. Conditional diffusion models implicitly constrain the output distribution by aligning with a text prompt embedding. Inspired by this, we propose a more general diffusion fine-tuning procedure that leverages constrained optimization, aiming to alter the underlying distribution of the diffusion models with a guiding function.

To provide further motivations, we briefly discuss some potential issues in pre-trained diffusion models which are solvable with our pipeline. In policy diffusion models, a genre of phenomenon named "hallucination" severely impact the execution of video policy by making up objects or scenes inconsistent with history frames Aithal et al. (2024); Betti et al. (2024), leading to risky or hazardous situations. Other than diffusion policy, sometimes we might intend to avoid certain outcomes in the probabilistic distribution, even though they are present in the original dataset. For example, when leveraging a diffusion model to generate possible indoor scenes with furniture and walls, we might want it to not generate blue backgrounds because the end users of the diffusion model do not favor the color blue. For a more practical case of safe robot navigation task, due to unforeseen obstacles, we might want the robot to avoid a certain region in the room when it's reaching towards its goal, even though that region is accessible in the original training data.

To achieve tasks alike the aforementioned, we are inherently trying to shape a pre-trained diffusion model to our desired directions. Re-training the diffusion model from scratch on the desired clean training dataset requires a lot of resources.

It thus becomes natural to fine-tune diffusion models to suit human's preferences. In reinforcement learning, a popular approach is Reinforcement Learning with Human Feedback (RLHF). A reward model is learned to capture what humans care about in the task, which is then used to train the agent.

Compared to designing the reward, a more efficient way would be to directly reflect the constraints in the diffusion process.

In this work, we present a simple pipeline to fine-tune unconditional diffusion models via a constrained optimization process. Given a pre-trained diffusion model and a constraint on the desired outcome , we aim to remove undesirable outputs from its probability distribution. To achieve this, we leverage the penalty-based method, where the constrained diffusion model is formulated as an optimization problem that minimizes the diffusion objective subject to the imposed constraints. Compared to Reinforcement Learning With Human Feedback (RLHF)(Fan et al. (2023)), our method requires much less training and fewer resources, thus offering a simpler framework for diffusion fine-tuning problems than in the context of reinforcement learning. That being said, we do believe that our work can be understood intuitively as a variance of reward shaping in diffusion RLHF.

To provide evidence of our work's functionality, we conducted experiments in at least three tasks. First, We were able to cleanly remove the class of digits 4 from an MNIST dataset without affecting the quality of other digits. Secondly, we show succession in manipulating diffusion policy on mini-grid path finding-problems. Given a mini-grid maze planning problem, we create additional obstacles in the maze, and constrain the agent from going inside a pre-defined epsilon ball of the target. Our last task is to optimize with human preference pairs on polymers generated by pre-trained diffusion model. This task demonstrates our algorithm's compatibility with the Bradly-Terry Model (Bradley & Terry (1952)) and connects this work to RLHF as well as interdisciplinary work.

To sum up, our contributions consist of the following:

- We introduce a simple pipeline to fine-tune unconditional diffusion models via a constrained optimization process, leveraging penalty-based method to generalize constraints to non-convex functions.
- Our framework works across a wide range of tasks, offering a more generalizable approach that can be readily utilized on pre-trained diffusion models with a constraint function.
- We provide an empirical analysis on the convergence of our algorithm, and discuss the relationship between Constrained Fine-tuning and RLHF.

2 Background and Related Works

Constrained Optimization A constrained optimization problem P can be formulated as below:

$$P: \quad \min_{x \in \mathbb{R}^n} f(x) \tag{1}$$

$$s.t. \quad g_i(x) \leq 0, i = 1, \ldots, m \tag{2}$$

$$h_i(x) = 0, i = 1, \ldots, k \tag{3}$$

$$x \in \mathbb{R}^n \tag{4}$$

The equality constraints $h_i(x) = 0$ can be further rewritten into $h_i(x) \leq 0$, $h_i(x) \geq 0$, and therefore added to the inequality constraints. One major class of algorithm to solve such problem is the penalty algorithm Tessema & Yen (2006); Lin (2013); Barbosa & Lemonge (2008) and the barrier algorithm Doyle (2004); Dvurechensky & Staudigl (2024); Bomze et al. (2019), which construct sequence of unconstrained optimization problems that under continuity of f and g_i converge at infinity.

Diffusion Models Diffusion models are generative models where a Markov chain is defined to iteratively add noise to the image in the forward process, and a reverse process is learned to denoise and construct desired image samples from noise Weng.

In the forward process, we are given a data point sampled from the real distribution $\mathbf{x}_0 \sim q(x)$, where we add small amounts of Guassian noise up until timestep T. This gives us samples $\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T$, as specified by the following:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \boldsymbol{I}\right) \tag{5}$$

, where $\{\beta_t \in (0,1)\}_{t=1}^T$ is a variance schedule that controls the step sizes. Let $\alpha_t = 1 - \beta_t$, and $\overline{\alpha}_t = \prod_{i=1}^t \alpha_i$. For $\epsilon \sim \mathcal{N}(0, \mathbf{I})$,

$$\mathbf{x}_t = \sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \overline{\alpha}_t}\epsilon \tag{6}$$

In the backward process, we reverse the forward process and sample from $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ from a random Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$. Because we don't know the ground truth data distribution, we learn p_θ to approximate the conditional probabilities in q, where

$$p_\theta\left(\mathbf{x}_{0:T}\right) = p\left(\mathbf{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) \tag{7}$$

Evidence lower bound (ELBO) is introduced as the lower bound of the log likelihood of observed data, where observed data is the sampled distribution from the diffusion model. Our goal is to maximize the ELBO, which is a proxy objective for optimizing the latent varable model p_θ. The log of the evidence

$$\log q(\mathbf{x}) = \underbrace{\mathbb{E}_{p_\theta(\mathbf{x}_{1:T}|\mathbf{x})}[\log \frac{q(\mathbf{x}_{0:T})}{p_\theta(\mathbf{x}_{1:T}|\mathbf{x})}]}_{\text{(1) ELBO}} + \underbrace{\mathcal{D}_{KL}(p_\theta(\mathbf{x}_{1:T} \mid \mathbf{x}_0)||q(\mathbf{x}_{1:T} \mid \mathbf{x}))}_{\text{(2) KL Divergence } \geq 0} \tag{8}$$

Since $\log q(\mathbf{x})$ is constant with respect to θ, optimizing the ELBO is equivalent to minimizing KL divergence. By Equation 6, minimizing KL divergence can be simplified to minimizing the following mean squared error as prediction loss:

$$\mathcal{L}_t(\theta) = \mathbb{E}_{\mathbf{x}_0,\epsilon,t}[||\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \overline{\alpha}_t}\epsilon, t)||^2] \tag{9}$$

, where ϵ_θ is the noise prediction at timestep t.

Diffusion models can also be guided towards certain outputs by designing score functions $s_\theta(\mathbf{x})$. The gradient of such score function can be integrated into the gradient of the new evidence:

$$\nabla = \nabla_{\mathbf{x}} \log q(\mathbf{x}) + \nabla_{\mathbf{x}} s_\theta(\mathbf{x}) \tag{10}$$

Constrained Diffusion The constrained diffusion problem imposes restrictions on the output probability distribution of diffusion models. This requires adding constraints on its training process. Conditional diffusion models Dhariwal & Nichol (2021); Ho & Salimans (2022); Bansal et al. (2023) restrict generation through conditional information. But such restrictions are usually coarse, and can usually be represented by a simple text labeling.

In most realistic cases, the constraints are far more complicated. In order to work with these complex constraints, Liu et al. (2023); Du et al. (2024) balance generation using equal weights through compositional generation. Also leveraging conditional generation, Power et al. (2023) uses equal hyperparameters. Meanwhile, Friedrich et al. (2023) solves the constrained diffusion problem with fair diffusion. Different from the above work, distribution models are balanced by Lagrange Multipliers instead in Khalafi et al. (2024). They approached constrained diffusion models with dual training, but assume strong convexity in the constraint functions, such as KL divergence Shlens (2014). The constraint is limited to a target data distribution, i.e. the diffusion represented data distribution and the target distribution should be close. Yet we assume that a more general set of constrains usually comes in the form of continuous functions possibly parametrized by neural networks.

In addition, the objective of most current work is to balance data distribution from a biased training dataset, whereas in the real world, a lot of cases require us to avoid certain outputs, i.e. biasing the outcome to suit our preferences instead of the other way around, which makes the task non-trivial. To address this limitation, our method fine-tunes unconditional diffusion models via a constrained optimization process based on penalty-based methods, which doesn't require the strong convexity of constraints.

Reinforcement Learning with Human Feedback For applications in reinforcement learning, we desire the agents to conform to human preferences and values. For this purpose, recent works Lee et al. (2024); Knox (2011); Griffith et al. (2013) employ Reinforcement Learning with Human Feedback (RLHF) for policy shaping, where the agent is allowed to directly receive human feedback, and a reward model is optimized to fine-tune the optimized policy. It has also been applied in the context

of diffusion, where fine-tuning diffusion models is formulated as a reinforcement learning problem Zhao et al. (2025); Lin & Ye. However, learning a reward model requires resources, which can also be time consuming when we want to train an agent to perform a variety of tasks. Swamy et al. (2024) improved upon this by proposing an algorithm that requires a single agent to play against itself but does not require training a reward model. Black et al. (2023); Fan et al. (2023); Yang et al. (2024a); Hiranaka et al. (2024) have been able to incorporate human feedback online. Our method leverages diffusion policy, and directly changes the loss function during the diffusion process to incorporate score functions that reflect human preferences.

3 ALGORITHM

Denote the parameter of a pre-trained diffusion model by θ, our pipeline requires no more information than a set of continuous differentiable constraints $\{g_i, i \in I\}$ on clean images.

First, to prepare the data for fine-tuning, we either sample a batch of clean image output $p_\theta(\mathbf{x}_0|\mathbf{x}_T)$ when the training data is not available, or sample a batch of prepared training data. Then, we feed these training data back to the pre-trained diffusion model as in a standard diffusion training pipeline. The forward process adds Guassian noice according to a schedule as standard, but we formulate the backward steps in a constrained optimization matter. For each backwards training objective(usually for updating the UNet), where the true noise is defined by the constrained diffusion problem P^* as:

$$\boldsymbol{P}^*: \quad \min_{\mathbf{x}\in\mathbb{R}^n} \boldsymbol{L}_t(\theta) = \mathbb{E}_{\mathbf{x}_0,\epsilon,t}[||\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t)||^2] \tag{11}$$

$$s.t. \quad g_i(x) \leq 0, i \in \boldsymbol{I} \tag{12}$$

$$x \in \mathbb{R}^n \tag{13}$$

Given $\boldsymbol{L}_t(\theta)$ strongly convex, assume $\{g_i\}_{i=1}^M$ continuous differentiable functions for all i, we propose to leverage the penalty-based method for the above constrained optimization problem.

Our loss function is thus formulated as follows:

$$\begin{aligned}\boldsymbol{L}_t(\theta, \mathbf{C}) = &\underbrace{\mathbb{E}_{\mathbf{x}_0,\epsilon,t}[||\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t)||^2]}_{\text{(1) prediction loss } f(\mathbf{x}_0,t)} \\ &+ \mathbf{C} \cdot \underbrace{\mathbb{E}_{t<T_\phi}[\text{ReLU}(\mathbf{g}(\hat{\mathbf{x}}(\mathbf{x}_t,t) - \delta)^2]}_{\text{(2) penalty } \mathbf{p}(\mathbf{x}_t,t,\delta)}\end{aligned} \tag{14}$$

, where $\mathbf{C}$ is the penalty parameter, δ is the tolerance threshold, $\mathbf{g}$ is the vector of constraints, $\hat{\mathbf{x}}$ is the clean image constructed from a noised version of it, ϕ represents the task, and T_ϕ is task-dependent timestep threshold for adding the penalty loss. We pass the constraint function output into a ReLU activation to ensure the penalty term is strictly positive. This formulation of the problem P^* enables us to use tools from the optimization literature.

Then, we construct an alternative unconstrained program:

$$\boldsymbol{P}^*(\mathbf{C}): \min_x [f(\mathbf{x}_0, t) + \mathbf{C} \cdot \mathbf{p}(\mathbf{x}_t, t, \delta)] \tag{15}$$

so that $\mathbf{C}$ is an increase sequence and $\mathbf{C} \to \infty$. Let $\mathbf{C}_k \geq 0, k = 1, \ldots, \infty$, let x_k be the exact solution to the program $\boldsymbol{P}^*(\mathbf{C}_k)$, and let x^* be any optimal solution of $\boldsymbol{P}^*$,

The Penalty Convergence Theorem Suppose that $f(\mathbf{x}_0, t)$, $\mathbf{p}(\mathbf{x}_t, t, \delta)$, and $\mathbf{g}(\mathbf{x}_t, t)$ are continuous functions. Let $\{x_k\}, k = 1, \ldots, \infty$, be a sequence of solution to $\boldsymbol{P}^*(\mathbf{C}_k)$. Then any limit point $\bar{x}$ of $\{x_k\}$ solves $\boldsymbol{P}^*$.

In the actual execution of the algorithm, we apply the constraint only when the noise level is not too high, which is controlled by a hyperparameter T_ϕ. We optimize for this T_ϕ in each of our applications. An overview of our algorithm can be found in algorithm 1.

Figure 1: From left to right, above are samples generated after finetuning at epochs 0, 24, and 44 with a dataset of 512 images sampled from the pretrained model. Yellow highlights are non-digit samples. Red highlights are discriminative class samples. As epochs increase, the number of discriminative class samples decrease and the number of non-digit samples slightly increase.

3.1 An alternative interpretation

We offer alternative interpretations of the aforementioned algorithm by linking it with RLHF. The constraint diffusion algorithm is able to inject a form of reward signal at each noise level, instead of only reflecting in the end. The idea intuitively follows that of the direct policy gradient for LLM RLHF. We propose that this algorithm can also be viewed as a form of reward-shaping in RLHF. A full investigation is still needs to bridge the two methods.

Algorithm 1 Practical Implementation of Constrained Diffusion Algorithm

Require: Maximum number of penalty iterations I, diffusion timestep t, initial penalty parameters c, penalty growth factor r, penalty function $\mathbf{g}$, number of iterations per penalty growth n, diffusion for noise prediction ϵ_θ, function to reconstruct original sample with noise prediction diffusion model $\hat{x}$, noise schedule $\overline{\alpha}_t$, ground truth sample x_0, timestep threshold T_ϕ, learning rate η, early stopping threshold τ

for $i = 1$ **to** I **or** until convergence such that $\mathcal{L}_{\text{penalty}} \leq \tau$ **do**
  $\epsilon \sim \mathcal{N}(0, 1)$
  $\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{\mathbf{x}_0,\epsilon,t}[||\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t)||^2]$
  $\mathcal{L}_{\text{penalty}} = \mathbb{E}_{t<T_\phi}[\text{ReLU}(\mathbf{g}(\hat{x}(x_t, t)) - \delta)^2]$
  $\mathcal{L}(\theta) = \mathcal{L}_{\text{diffusion}} + c \cdot r^{\lfloor \frac{i}{n} \rfloor} \cdot \frac{||\mathcal{L}_{\text{diffusion}}||}{||\mathcal{L}_{\text{penalty}}||} \cdot \mathcal{L}_{\text{penalty}}$
  Compute gradient $g_i \leftarrow \nabla_\theta \mathcal{L}(\theta)$
  Update parameters $\theta \leftarrow \theta - \eta \cdot g_i$
end for
return θ

4 Experiments

In this section, we demonstrate experimental results using our approach. We first conduct experiments using the MNIST pap dataset which has a relatively small scale. We then move on to validate our method on maze planning tasks using diffusion policy Chen et al. (2024b) and on Polymer.

4.1 MNIST

MNIST is a dataset which consists of images of digits from 0 to 9 in color white on black backgrounds. Our goal is to remove samples of a particular digit from the unconditional sample distribution of our diffusion model. This is a practical task that can be generalized to other tasks in which the user of the diffusion model aims to remove certain outputs from the unconditioned diffusion distribution. In our example, we choose to remove the digit 4 as it is often associated with misfortune in Chinese culture. It is also a non-trivial task, because when we are trying to maximize the KL-divergence between images of the digit 4 (which we aim to remove) and the generated images, it pushes our sample distribution away from not just the digit 4 but also all digits together, which could cause samples to not look like any digit.

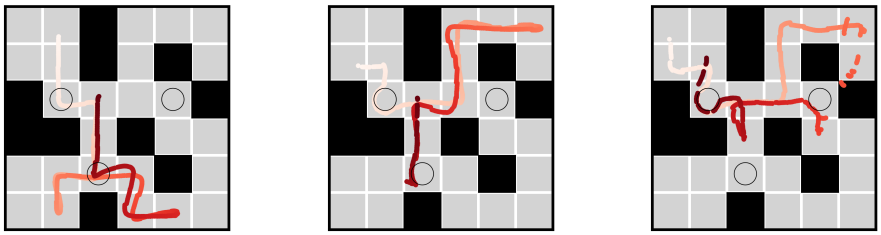

Figure 2: From left to right, above are random walk (unconditional) samples generated from the pretrained diffusion planner, a success case from the finetuned planner, and a failure case from the finetuned planner. The pretrained sample does not avoid the forbidden zones. The finetuned samples both avoid the forbidden zones with the success case retaining the continuity of the planned trajectory while the failure case trajectory loses continuity, becoming more dotted and sometimes ignoring obstacles.

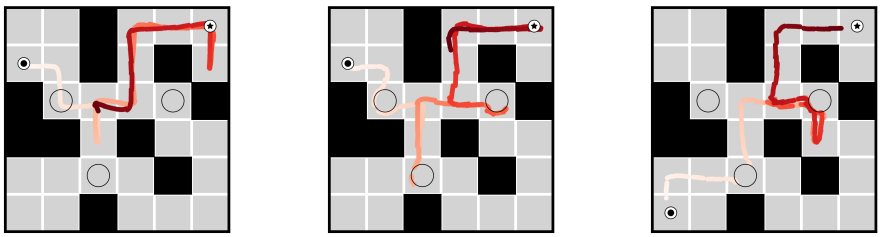

Figure 3: From left to right, above are goal-guided (conditional) samples generated from the pretrained diffusion planner, a success case from the finetuned planner, and a failure case from the finetuned planner. Similar to the results from the random walk, the pretrained sample does not avoid forbidden zones while the finetuned samples avoid the forbidden zones. In the failure case, we showcase a possible failure where the planned trajectory completely jumps over the forbidden zones without planning around it.

We represent our penalty score function as follows:

$$\mathbf{g}(\mathbf{x}) = \text{ReLU}(p_\theta(y \mid \mathbf{x}) - p_\phi) \tag{16}$$

, where y is the label for the class of digit 4, and we want to constrain the probability of generating images to digit 4 to be below p_ϕ. T_ϕ for this task is 800 timesteps.

Given the pretrained diffusion model, we first generate image samples of any digit. We create a dataset of 512 such generated images, and fine-tune the diffusion model on this created dataset with Algorithm 1. Our experimental results are presented in Figure 1. We show our diffusion outputs after finetuning at epochs 0, 24, and 44. Yellow highlights are non-digit samples. Red highlights are samples of digit 4. There's a trade-off between non-digit samples and samples of the discriminative class. As epochs increase, the number of digits 4 samples decrease, and the number of non-digit samples slightly increase.

4.2 Diffusion Planning Trajectory with Safety Constraints

We further performed experiments on fine-tuning diffusion planning models to satisfy safety constraints on the Maze2D Medium benchmark from D4RL (Fu et al., 2021). We start from pretrained **Diffusion Forcing** planners (Chen et al., 2024a) that denoises full trajectories token-wise, achieving state-of-the-art returns on unaided planning tasks. We introduce safety through user-defined forbidden zones defined by coordinates on the grid and a set radius around each point. At each fine-tuning step, we apply our constrained diffusion algorithm denoted by algorithm 1 to constrain the distance between every predicted point on the trajectory and the center of every forbidden zone to be at least δ. The only modification to the original training loop is the inclusion of the weighted penalty term as shown in equation 14; no safety-specific data, reward model, or hand-tune control barrier function is required, unlike policy-guided diffusion (Jackson et al., 2024) or Safe

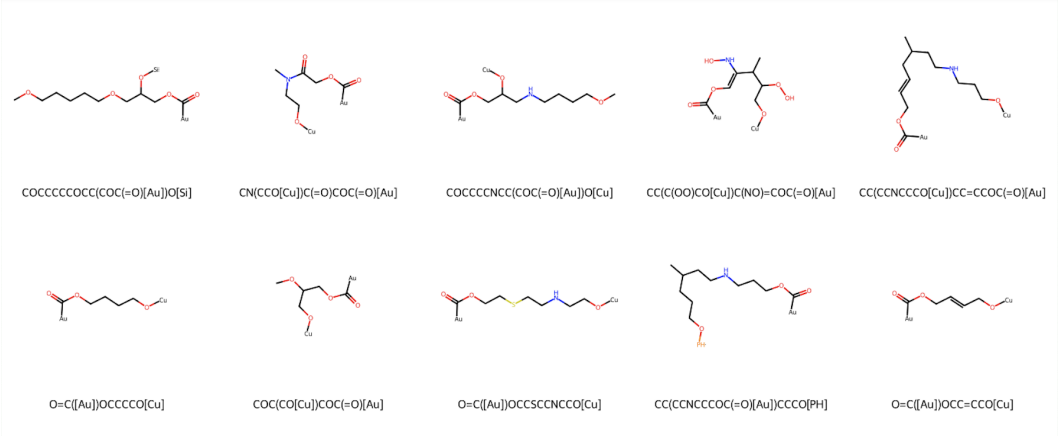

Figure 4: The novel conductive polymer structures and SMILES strings generated by baseline PolyGen with Diffusion1D backbone, trained on over 10000 examples.

Diffuser (Xiao et al., 2023). The fine-tuning experiment was carried out in both an unconditional setting (Fig. 2), where the training data are random walks in the 2D maze, and a conditional setting (Fig. 3), where the model is provided a goal position with fixed initial position and the training data are trajectories linking the initial position to the goal.

We design our score function as follows:

$$\mathbf{g}(\mathbf{x}) = \text{ReLU}(R - d(P, Q)) \tag{17}$$

, where R is the forbidden radius, $P = \{(p_{ix}, p_{iy})\}_{i=1}^{m}$ are the trajectory points, $Q = \{(q_{ix}, q_{iy}\}_{i=1}^{n}$ are the forbidden points, d is the distance between the set of points P and Q.

Results. Fine-tuning for 1 epoch guarantees safety constraints while preserving path smoothness. Training a fresh model with obstacle-augmented data for comparable performance required 7 hours on same devices. Qualitative roll-outs in Fig. 2 (unconditional) and Fig. 3 (conditional) show cohesive trajectories generated from both random walks and goal guidance that skirt the restricted regions.

4.3 (Future Work) Polymer Generation with Human Preference

Generative models have become a central paradigm in molecular design, enabling de-novo discovery of drug-like molecules, catalysts, and functional materials. Polymers are especially attractive because (i) their vast compositional space is only sparsely explored in the laboratory, and (ii) macroscopic properties—ionic conductivity, glass-transition temperature, mechanical strength—hinge on subtle variations in repeat-unit chemistry and chain architecture. Recent work shows that both language-model and diffusion-based generators can propose electrolyte polymers with target conductivities (Yang et al., 2024b) and tailor sequences for advanced composites (Ge et al., 2025). Consequently, polymers provide a realistic, high-impact setting to test algorithms that must **steer** a pretrained generator toward user-defined constraints (e.g. safety, synthesisability, or aesthetic preference).

Model: Diffusion1D backbone from PolyGen. We adopt the Diffusion1D architecture in PolyGen introduced by (Yang et al., 2024b). Diffusion1D treats a polymer repeat unit as a one-dimensional token sequence, applies Gaussian noise in the forward process, and leverages a 1D UNet to learn the reverse denoiser—mirroring 2D DDPMs but with weight-tying and causal convolutions optimised for sequence length ≤ 128 (Yang et al., 2024b). Because the model exposes intermediate hidden states, it remains compatible with external property predictors and our future penalty term. PolyGen also provides evaluation of novelty, uniqueness, sythesibility, validity, similarity and diversity of the generated polymers, making future evaluations comprehensive and adding more possibilities into penalty term shaping.

Following Yang et al. (2024b), we pretrained our baseline model with the conditional setting, where 11409 **Simplified Molecular Input Line Entry System (SMILES)** strings of **conductive polymers only** are used as the training set. We pre-trained our baseline model for 10000 steps, and the generated conductive polymers can be seen in 4. We plan to design a more complex penalty term from human feedbacks described in the following section.

Preference-driven scoring via Bradley–Terry. When expert chemists rank two generated candidates (x_i, x_j), we can fit a Bradley–Terry (BT) model that assigns latent "merit" scores γ_i such that

$$\Pr(x_i \succ x_j) = \gamma_i/(\gamma_i + \gamma_j)$$

(Liu et al., 2024). The BT log-likelihood:

$$\ell(\boldsymbol{\gamma}) = \sum_{(i,j)\in\mathcal{P}} y_{ij} \log \gamma_i - (1 - y_{ij}) \log \gamma_j - \log(\gamma_i + \gamma_j)$$

is differentiable; thus the estimated preference score $s_{\mathrm{BT}}(x) = \log \gamma_x$ serves as a continuous constraint $g(x)$ in Eq. (14). Compared with manually crafted heuristics or black-box RLHF reward models, BT offers (i) statistical efficiency on sparse pairwise data, (ii) an interpretable scale, and (iii) negligible computational overhead. We therefore collect a small pool of $\approx 5\,000$ preference pairs on PolyGen samples and optimise our fine-tuning objective with $g(x) = \delta - s_{\mathrm{BT}}(x)$, where δ encodes the user-specified desirability threshold. This shows that out penalty based method is especially adaptive, being able to take advantage of any score or reward functions that shows generalization ability in recent development of RLHF, with a much simpler finetuning pipeline.

5 Conclusion

We present a simple pipeline to fine-tune unconditional diffusion models via a constrained optimization process. We leverage penalty-based method to monitor loss during the diffusion process, while offering a generalized constraints that removes the strong convexity assumption. Our formulation has broad compatibilities in experiments ranging from computer vision tasks to diffusion policy tasks in reinforcement learning, where the incorporation of score function reflecting human preferences also offers a more straightforward approach than RLHF.

References

Papers with Code - MNIST Dataset — paperswithcode.com. https://paperswithcode.com/dataset/mnist. [Accessed 06-05-2025].

Sumukh K Aithal, Pratyush Maini, Zachary C. Lipton, and J. Zico Kolter. Understanding hallucinations in diffusion models through mode interpolation, 2024. URL https://arxiv.org/abs/2406.09358.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models, 2023. URL https://arxiv.org/abs/2302.07121.

Helio JC Barbosa and Afonso CC Lemonge. An adaptive penalty method for genetic algorithms in constrained optimization problems. Citeseer, 2008.

Federico Betti, Lorenzo Baraldi, Lorenzo Baraldi, Rita Cucchiara, and Nicu Sebe. Optimizing resource consumption in diffusion models through hallucination early detection, 2024. URL https://arxiv.org/abs/2409.10597.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. arXiv preprint arXiv:2305.13301, 2023.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023. URL https://arxiv.org/abs/2311.15127.

Immanuel M. Bomze, Panayotis Mertikopoulos, Werner Schachinger, and Mathias Staudigl. Hessian barrier algorithms for linearly constrained optimization problems. SIAM Journal on Optimization, 29(3):2100–2127, 2019. doi: 10.1137/18M1215682. URL https://doi.org/10.1137/18M1215682.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika, 39(3/4):324–345, 1952. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/2334029.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion, 2024a. URL https://arxiv.org/abs/2407.01392.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion, 2024b. URL https://arxiv.org/abs/2407.01392.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Zibin Dong, Yifu Yuan, Jianye Hao, Fei Ni, Yi Ma, Pengyi Li, and Yan Zheng. Cleandiffuser: An easy-to-use modularized library for diffusion models in decision making, 2024. URL https://arxiv.org/abs/2406.09509.

Maureen Doyle. A barrier algorithm for large nonlinear optimization problems. stanford university, 2004.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc, 2024. URL https://arxiv.org/abs/2302.11552.

Pavel Dvurechensky and Mathias Staudigl. Barrier algorithms for constrained non-convex optimization, 2024. URL https://arxiv.org/abs/2404.18724.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2305.16381.

Felix Friedrich, Manuel Brack, Lukas Struppek, Dominik Hintersdorf, Patrick Schramowski, Sasha Luccioni, and Kristian Kersting. Fair diffusion: Instructing text-to-image generation models on fairness, 2023. URL https://arxiv.org/abs/2302.10893.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2021. URL https://arxiv.org/abs/2004.07219.

Wei Ge, Ramindu Silva, Yanan Fan, Scott Sisson, and Martina Stenzel. Machine learning in polymer research. Advanced Materials, 37, 02 2025. doi: 10.1002/adma.202413695.

Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L. Isbell, and Andrea Thomaz. Policy shaping: integrating human feedback with reinforcement learning. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, pp. 2625–2633, Red Hook, NY, USA, 2013. Curran Associates Inc.

Ayano Hiranaka, Shang-Fu Chen, Chieh-Hsin Lai, Dongjun Kim, Naoki Murata, Takashi Shibuya, Wei-Hsiang Liao, Shao-Hua Sun, and Yuki Mitsufuji. Human-feedback efficient reinforcement learning for online diffusion model finetuning. arXiv preprint arXiv:2410.05116, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. URL https://arxiv.org/abs/2204.03458.

Matthew Thomas Jackson, Michael Tryfan Matthews, Cong Lu, Benjamin Ellis, Shimon Whiteson, and Jakob Foerster. Policy-guided diffusion, 2024. URL https://arxiv.org/abs/2404.06356.

Shervin Khalafi, Dongsheng Ding, and Alejandro Ribeiro. Constrained diffusion models via dual training, 2024. URL https://arxiv.org/abs/2408.15094.

W. B. Knox. Augmenting reinforcement learning with human feedback. 2011. URL https://api.semanticscholar.org/CorpusID:17123490.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. RLAIF: Scaling reinforcement learning from human feedback with AI feedback, 2024. URL https://openreview.net/forum?id=AAxIs3D2ZZ.

Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds, 2023. URL https://arxiv.org/abs/2306.00980.

Chih-Hao Lin. A rough penalty genetic algorithm for constrained optimization. Information Sciences, 241:119–137, 2013. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2013.04.001. URL https://www.sciencedirect.com/science/article/pii/S0020025513002776.

Ying Lin and Fei Ye. Optimizing generative diffusion models with reinforcement learning from human feedback (rlhf) in architectural: A case study of campus layouts.

Jinsong Liu, Dongdong Ge, and Ruihao Zhu. Reward learning from preference with ties, 2024. URL https://arxiv.org/abs/2410.05328.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models, 2023. URL https://arxiv.org/abs/2206.01714.

Thomas Power, Rana Soltani-Zarrin, Soshi Iba, and Dmitry Berenson. Sampling constrained trajectories using composable diffusion models. In IROS 2023 Workshop on Differentiable Probabilistic Robotics: Emerging Perspectives on Robot Learning, 2023. URL https://openreview.net/forum?id=UAylEpIMNE.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation, 2023. URL https://arxiv.org/abs/2208.12242.

Jonathon Shlens. Notes on kullback-leibler divergence and likelihood, 2014. URL https://arxiv.org/abs/1404.2000.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Zhiwei Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback, 2024. URL https://arxiv.org/abs/2401.04056.

B. Tessema and G.G. Yen. A self adaptive penalty function based algorithm for constrained optimization. In 2006 IEEE International Conference on Evolutionary Computation, pp. 246–253, 2006. doi: 10.1109/CEC.2006.1688315.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training, 2021. URL https://arxiv.org/abs/2105.03404.

Lilian Weng. What are Diffusion Models? — lilianweng.github.io. https://lilianweng.github.io/posts/2021-07-11-diffusion-models/. [Accessed 08-05-2025].

Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, and Daniela Rus. Safediffuser: Safe planning with diffusion probabilistic models, 2023. URL https://arxiv.org/abs/2306.00148.

Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. A survey on video diffusion models, 2024. URL https://arxiv.org/abs/2310.10647.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8941–8951, 2024a.

Zhenze Yang, Weike Ye, Xiangyun Lei, Daniel Schweigert, Ha-Kyung Kwon, and Arash Khajeh. De novo design of polymer electrolytes using gpt-based and diffusion-based generative models. npj Computational Materials, 10(1), December 2024b. ISSN 2057-3960. doi: 10.1038/s41524-024-01470-9. URL http://dx.doi.org/10.1038/s41524-024-01470-9.

Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, In So Kweon, and Junmo Kim. Text-to-image diffusion models in generative ai: A survey. 2024. URL https://arxiv.org/abs/2303.07909.

Hanyang Zhao, Haoxian Chen, Ji Zhang, David D. Yao, and Wenpin Tang. Score as action: Fine-tuning diffusion generative models by continuous-time reinforcement learning, 2025. URL https://arxiv.org/abs/2502.01819.

Table 1: Hyperparameters for MNIST experiment.

Hyperparameter	Value	Description
I	50	Maximum number of penalty iterations
c	0.05	Initial penalty parameter
r	1.2	Penalty growth factor
n	10	Number of iterations per penalty growth
η	3e-4	Learning rate
τ	0.01	Early stopping threshold
Batch size	128	Batch size per finetuning step
Epoch	100	Number of finetuning epochs

Table 2: Parameters of U-Net model for MNIST experiment.

# Res-Net layers per U-Net block	2
# Res-Net down/upsampling blocks	6
# Output channels for U-Net blocks	(128, 128, 256, 256, 512, 512)

A Experiments

A.1 MNIST

This experiment is adapted from the MNIST experiment from Constrained Diffusion Models via Dual Training (Khalafi et al., 2024).

Hyperparameters. For the penalty score function in the MNIST experiment Eq. (16), we pretrain an MNIST classifier and apply softmax to the predicted logits to approximate $p_\theta(y \mid \mathbf{x})$. We choose $p_\phi = 0.1$ based on the intuition that 0.1 is equivalent to random chance over 10 digits. We choose our hyperparameters for Algorithm 1 empirically, which are summarized in Table 1.

Model architecture. We follow the original paper (Khalafi et al., 2024) and use a time-conditioned U-Net model for the MNIST experiment. We add time conditioning as a positional embedding of the timestep to the input image. See Table 2 for the model parameters.

Compute resources. We run the MNIST experiments on Google Colab premium with the A100 GPU setting. For each run, the experiment takes roughly 10 minutes.

FID. We use FID to evaluate our generation results. We sample 990 samples from the diffusion model and 110 from every class in our MNIST dataset except for the discriminative class, which is 4 in our experiment. We show the FID trend in Figure 5. The FID trend increases due to the tradeoff of image quality for the decrease in occurrences of the discriminative class.

A.2 Diffusion Planning

This experiment is adapted from the Maze2D Planning experiment from Diffusion Forcing (Chen et al., 2024a).

Hyperparameters. For the penalty score function in the Diffusion Planning experiment Eq. (17), we use euclidean distance as our distance function $d(P, Q)$ and choose $R = 0.3$ for the radius of our forbidden zones in our experiments in order to leave space for the trajectory to move around the zones while continuing on its planned trajectory. We choose our hyperparameters for Algorithm 1 empirically, which are summarized in Table 3.

Model architecture. We follow the architecture from Diffusion Forcing (Chen et al., 2024a) and use residue MLPs (Touvron et al., 2021) instead of U-Net for the diffusion model that is behind the dynamics model.

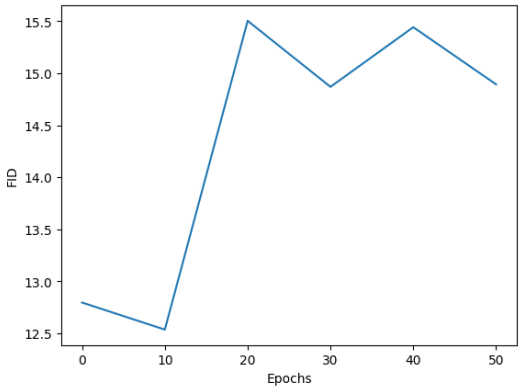

Figure 5: This graph shows the change in FID over epochs trained during the MNIST experiment. The FID increases overtime as image quality is traded off with the decrease in samples of the discriminative class.

Table 3: Hyperparameters for Diffusion Planning experiment.

Hyperparameter	**Value**	**Description**
I	100	Maximum number of penalty iterations
c	0.04	Initial penalty parameter
r	1.2	Penalty growth factor
n	10	Number of iterations per penalty growth
η	5e-4	Learning rate
τ	0.008	Early stopping threshold
Batch size	128	Batch size per finetuning step
Epoch	1	Number of finetuning epochs

Compute resources. We run the Diffusion Planning experiment on an M2 Macbook Pro. For a full run over one epoch, the experiment takes around 4-5 hours. We find that we can achieve similar results at a lower quality in around 30 minutes of runtime with the hyperparameters from Table 4.

B Division of Labor

Naicheng He is the initiator and organizer of the group. He was mainly responsible for the central algorithm design and the manuscript writeup. He also helps on the debugging side.

Nuo Wen Lei is responsible for implementing the algorithm with practical adjustments for MNIST and diffusion planning, running comprehensive experiments for those respective tasks, and documenting the details in the experiments section of the appendix. He also helps generate the figures used in the report.

Wanjia Fu works on literature review and related theory and work surrounding constrained diffusion and penalty-based methods. She was mainly responsible for the manuscript writeup. She also helps run the baseline for the maze planning task for diffusion planning.

Yixiang Sun researches applications with great implications and values that also fit the framework of fine-tuning diffusion model with constrained optimization. He runs the baselines for the diffusion planning and the polymer generation experiments. He is mainly responsible for the writing of the diffusion planning and polymer generation problems.

Table 4: Faster Hyperparameters for Diffusion Planning experiment.

Hyperparameter	**Value**	**Description**
I	100	Maximum number of penalty iterations
c	0.05	Initial penalty parameter
r	1.2	Penalty growth factor
n	10	Number of iterations per penalty growth
η	5e-4	Learning rate
τ	0.01	Early stopping threshold
Batch size	128	Batch size per finetuning step
Epoch	0.2	Number of finetuning epochs